11 The Video and Sound Macrocell

This chapter introduces the ARM7500FE video and sound system.

The Video and Sound Macrocell

11.1 Introduction

The ARM7500FE single chip computer contains a high performance video and sound controller, capable of meeting the requirements of a wide range of configurations.

The video and sound macrocell handles all the video processing aspects of the ARM7500FE functionality, making the ARM7500FE suitable for incorporation into a wide range of end products ranging from portable hand-held LCD systems through to higher performance SuperVGA desktop products.

The flexible bus interface provides hardware support for interfacing to DRAM memory systems in conjunction with the ARM7500FE memory controller. The video and sound macrocell obtains data from external DRAM under DMA control. The macrocell also incorporates a stereo digital sound system, with a serial sound output port suitable for connection to an external CD DAC.

Features include:

- VGA, SuperVGA, XGA resolution
- three 8-bit DACs giving 16M colors
- direct driving of LCD or CRT screens
- 1, 2, 4, 8, 16, 32 bits per pixel modes
- up to 120MHz pixel rate
- very low power consumption

11.2 Features

11.2.1 Flexible video system

The video and sound macrocell contains 288 write-only registers which offer a high degree of flexibility to the system programmer. 256 of these are used as the 28-bit video palette entries. These are programmed via an auto-incrementing address pointer. The remaining registers are specific control registers and allow the user to program the display parameters.

11.2.2 Hardware cursor

The video and sound macrocell has a hardware cursor for all its display modes:

- Normal
- Hi-Res
- LCD

By offering cursor support on chip the designer benefits in terms of speed and lower software overhead. The cursor is 32 pixels wide and any number of pixels high and can be displayed in 4 colors including transparent from its own 28-bit wide palette. In this way a cursor of any shape and size can be defined within the 32-pixel wide limit.

11.2.3 Palette

The video subsystem has a 28-bit wide 256-entry palette where each entry uses 8 bits for Red, 8 for Green and 8 for Blue, and 4 bits for external data. These external bits may be used outside the chip for a variety of purposes such as supremacy, fading, Hi-Res and LCD driving.

Look Up Tables (LUT) allow for logical to physical translation and gamma correction.The Red Green and Blue LUTs each drive their respective DACs, and the Ext LUT is normally configured to drive the 4-bit output port.

There are three 8-bit linear monotonic DACs (Red, Green and Blue) which give a total of 16M possible colors. The DACs are designed to operate up to 120 MHz and drive doubly-terminated 75Ω lines directly.

11.2.4 Pixel clock

The ARM7500FE is capable of generating a display at any pixel rate up to 120MHz. The pixel clock may be selected from one of 3 sources, and then the selected frequency of this clock may be further divided down by a factor of between 1 and 8.

The video and sound macrocell contains an on-chip phase comparator which, when used in conjunction with an external Voltage Controlled Oscillator (VCO), forms a Phase Locked Loop. This configuration allows a single reference clock to generate all the required frequencies for any display mode thus obviating the need for multiple external crystals.

11.2.5 Display modes

Irrespective of the memory configuration used, the video subsystem is capable of many different display formats. In addition to the normal linear CRT display, the video subsystem can generate a display suitable for either very high resolution displays, single or dual-panel LCDs.

For CRT displays, the video and sound macrocell is capable of operating in a variety of pixel modes - 1,2,4,8,16,32 bits/pixel, and can also directly drive LCD displays in 1,2 or 4 bits per pixel via an internal 16-level grey scaler. The grey scaler algorithm adopted is patented.

11.2.6 Power management

The macrocell is designed for power sensitive applications and incorporates design features to minimize power consumption. A *power down mode* allows power savings to be made when the device is not in use, for example, in conjunction with a battery powered LCD system. Additional power sensitive features include the powering down of functions of the device currently not in use, such as the video DACs and the LCD grey scaler. In addition the palette design has been segmented such that only one eighth of the palette is enabled and clocked at any one time. The power-down mode can be used in conjunction with the ARM7500FE's STOP mode to ensure minimum power consumption when clocks are stopped.

11.2.7 On-chip sound system

The ARM7500FE supports a 32-bit serial sound output suitable for driving external CD DACs. Enhanced 32-bit stereo sound is offered by the serial sound output, which consists of a three-pin serial interface. Each 32-bit sample consists of 16 bits for the left channel and 16 bits for the right channel.

11.3 Block Diagram

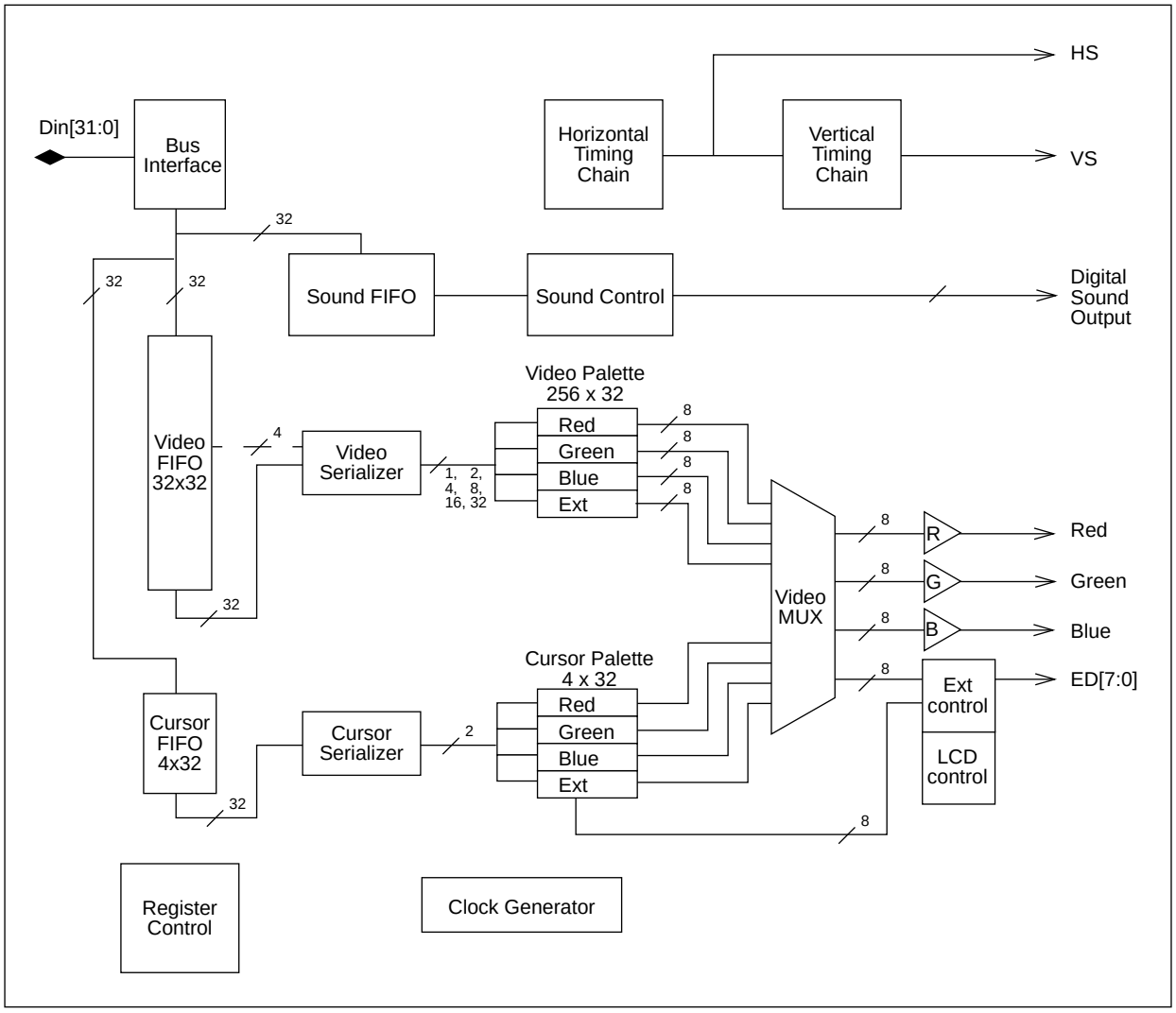


Figure 11-1: Video and sound macrocell block diagram

12 The Video and Sound Programmer's Model

This chapter details the video and sound macrocell programmable registers.

12.1 The Video and Sound Macrocell Registers

The video and sound macrocell contains 288 write-only registers. These are split into 2 categories; the 256 28-bit video palette entries, and the remaining control registers. The video palette entries are written via an auto-incrementing address pointer. All the other registers (including the 28-bit cursor palette) are written directly with the address encoded in the top 4 or 8 bits of the data word. To program the registers, the ARM7500FE address bus should be set to between 0x03400000 and 0x034FFFFF, and the data word written should include the individual register address in the upper 4 or 8 bits, as appropriate.

In order to define the display format correctly, eleven registers need to be programmed as shown in the diagram below:

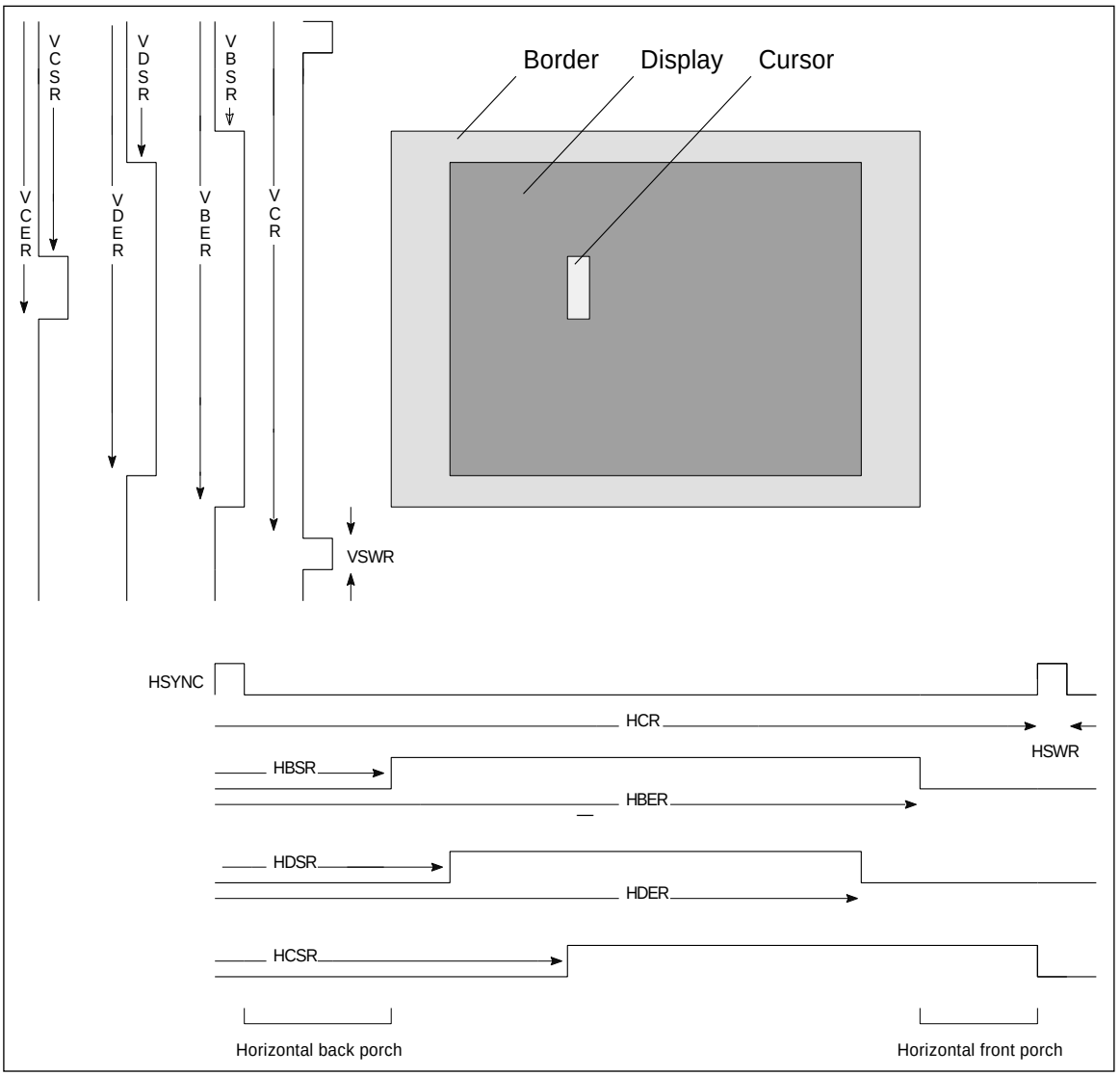


Figure 12-1: The video and sound macrocell display format definitions

The register allocation is shown in *Table 12-1: The video and sound macrocell register allocation*. An x denotes the actual data field, and any unused bit should be programmed with a logic zero.

Do not access any register at any location other than that shown as the actual register map is multiple-mapped.

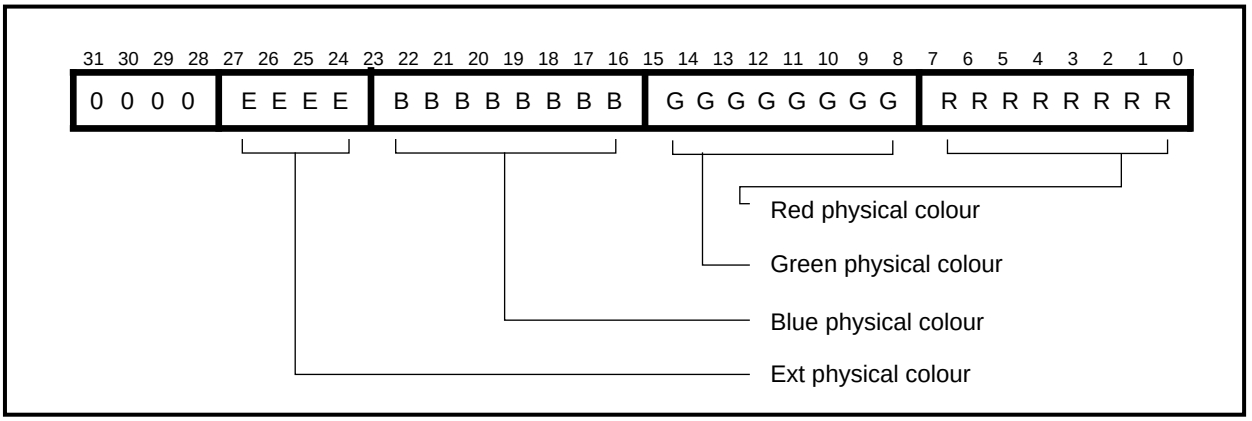
Address (hex)	Register
0xxxxxxx	Video Palette
100000xx	Video Palette Address Register
20000000	RESERVED
300000xx	LCD Offset register 0
310000xx	LCD offset register 1
4xxxxxxx	Border Color Register
5xxxxxxx	Cursor Palette logical color 1
6xxxxxxx	Cursor Palette logical color 2
7xxxxxxx	Cursor Palette logical color 3
8000xxxx	Horizontal Cycle Register
8100xxxx	Horizontal Sync Width Register
8200xxxx	Horizontal Border Start Register
8300xxxx	Horizontal Display Start Register
8400xxxx	Horizontal Display End Register
8500xxxx	Horizontal Border End Register
8600xxxx	Horizontal Cursor Start Register
8700xxxx	Reserved
8800xxxx	Test Register

Address (hex)	Register
8C00xxxx	Test Register
9000xxxx	Vertical Cycle Register
9100xxxx	Vertical Sync Width Register
9200xxxx	Vertical Border Start Register
9300xxxx	Vertical Display Start Register
9400xxxx	Vertical Display End Register
9500xxxx	Vertical Border End Register
9600xxxx	Vertical Cursor Start Register
9700xxxx	Vertical Cursor End Register
9800xxxx	Test Register
9A00xxxx	Test Register
9C00xxxx	Test Register
B00000x	Sound Frequency Generator
B10000x	Sound Control Register
C00xxxxx	External Register
D000xxxx	Frequency Synthesis Register
E00xxxxx	Control Register
F000xxxx	Data Control Register

Table 12-1: The video and sound macrocell register allocation

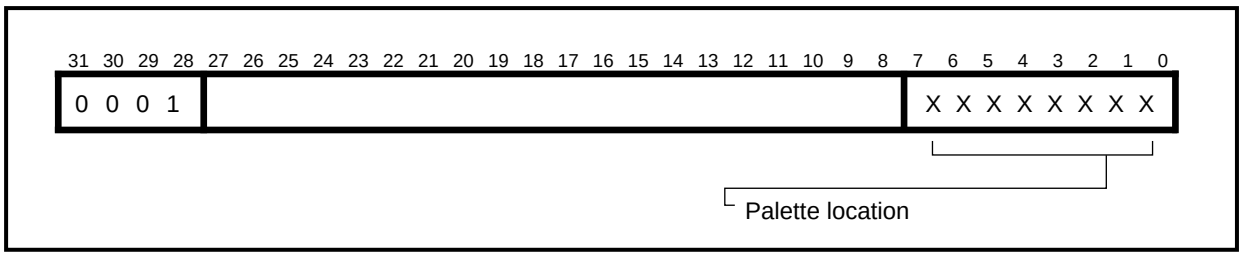
The External Register, Control Register, Sound Control Register and Data Control Register all contain bits that are not initialized at power up, and so must be programmed before the video and sound macrocell will operate correctly.

12.2 Video Palette: Address 0x0



All entries of the video palette are written at address 0. In order to write any or all of the palette locations, the address pointer must first be written, as described below. The palette is programmed with a 28-bit word representing the physical data field

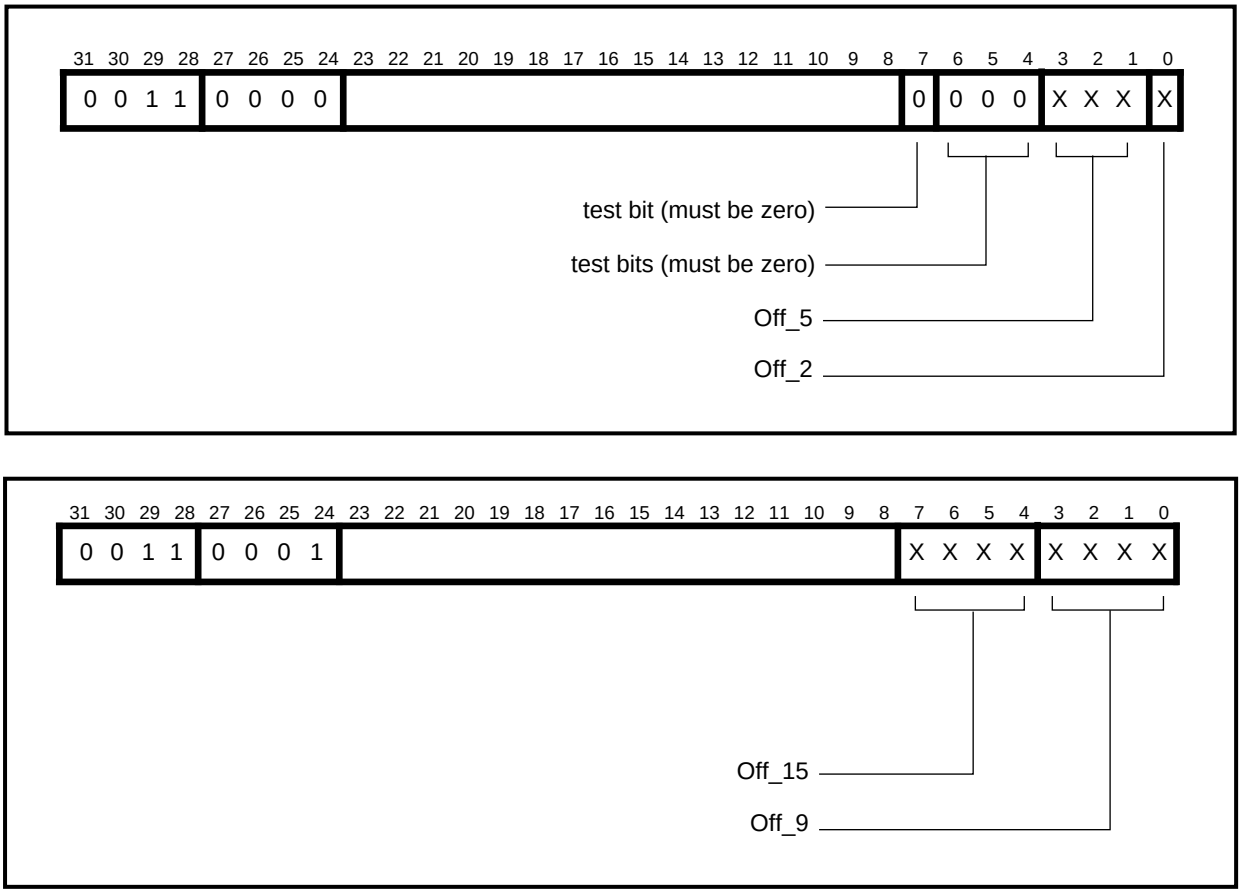
12.3 Video Palette Address Pointer: Address 0x1



The address pointer is programmed at address 1, and it may be programmed to any value from 0 to 255. The first write to the palette will then occur at this location, and the address pointer will post-increment so that the next palette write will occur to the following location. The counter will wrap around from 255 to 0.

Once the address pointer has been written, any number of palette locations can be programmed, and the pointer can be reprogrammed at any time if only part of the whole palette is to be updated.

12.4 LCD Offset Registers: Addresses 0x30 and 0x31



These two, 8-bit registers define the offsets required for driving a dual panel LCD screen. Register 0 defines the offsets for the five and two frame duty cycle grey scales, as well as reset and test mode bits. Register 1 defines the offsets for the nine and fifteen frame duty cycle grey scales.

The registers values are dependent upon the size of the LCD screen to be driven, and are calculated in the following way:

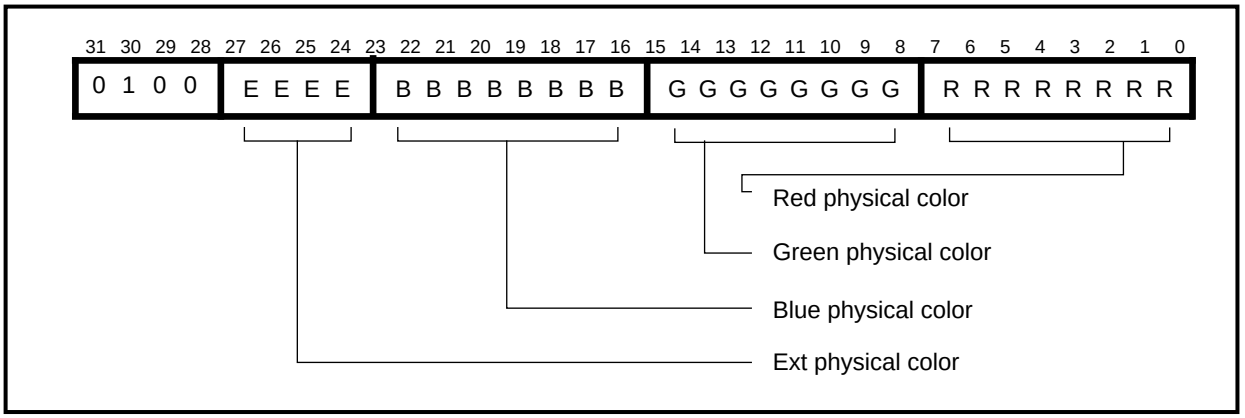
Off_15 = (3xL + 8) mod 15
Off_9 = (7xL + 4) mod 9
Off_5 = (1xL + 3) mod 5
Off_2 = 0

Where L is the number of lines in the upper panel of the dual panel LCD screen.

Bits 7-4 of register 0 are only used in test mode, and must all be set to zero in normal operation.

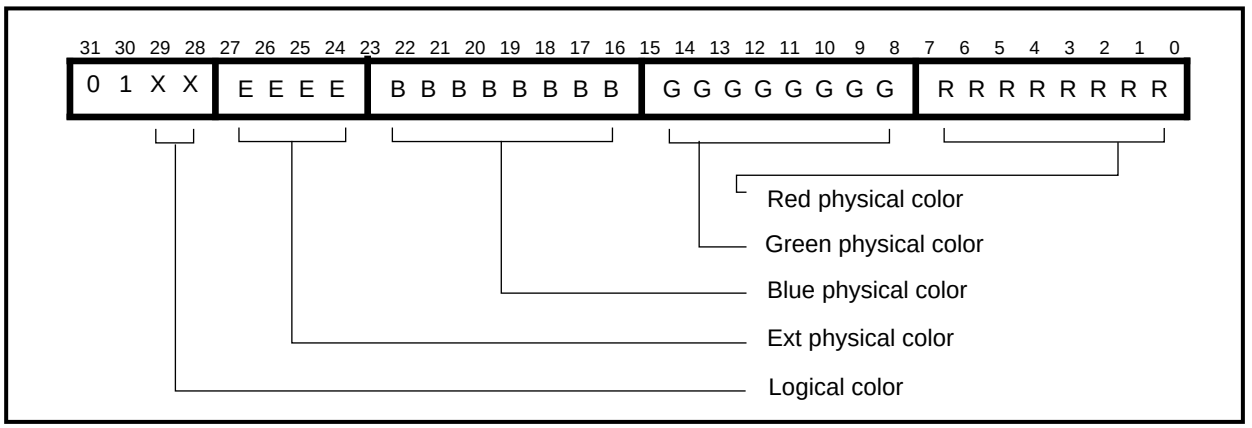
msel[2:0] are test bits and should be programmed LOW.

12.5 Border Color Register: Address 0x4



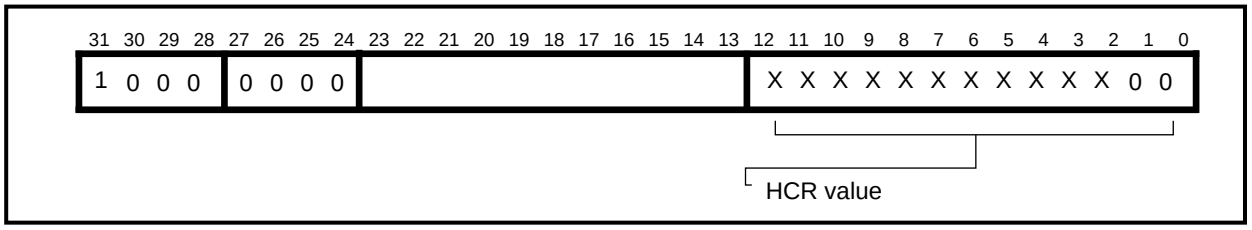
This register defines the physical border color, and is programmed with a 28-bit word. Note that this register is programmed directly, independent of the value of the video palette address pointer.

12.6 Cursor Palette: Addresses 0x5-0x7



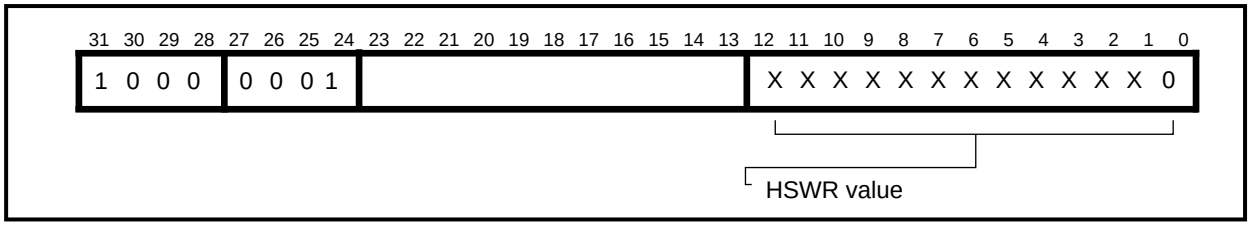
These three registers are programmed with the physical color of the three logical cursor colors. Note that cursor logical color 00 is defined as being transparent (i.e. no cursor display), and its location is used for the Border Color Register above.

12.7 Horizontal Cycle Register (HCR): Address 0x80



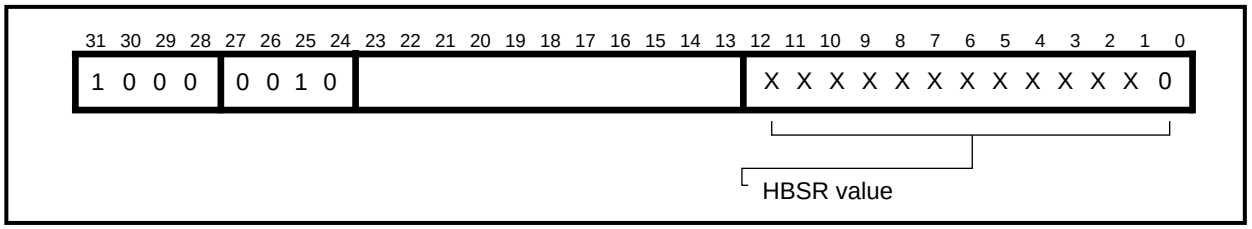
This register defines the period, in pixels, of the horizontal scan, i.e. display time + retrace time.
This is a 14-bit register of which the bottom 2 bits must be programmed to 0. If N pixels are required in the horizontal scan period, then value (N-8) should be programmed into the HCR. (N must be a multiple of 4).

12.8 Horizontal Sync Width Register (HSWR): Address 0x81



This register defines the period, in pixels, of the **HSYNC** pulse.
This is a 14-bit register of which the bottom bit must be programmed to 0. If N pixels are required in the **HSYNC** pulse, then value (N-8) should be programmed into the HSWR. (N must be a multiple of 2).

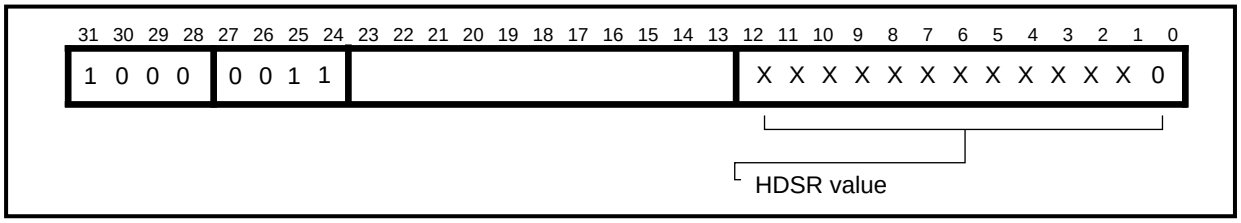
12.9 Horizontal Border Start Register (HBSR): Address 0x82



This register defines the time, in pixels, from the start of the **HSYNC** pulse to the start of the border display.
This is a 14-bit register of which the bottom bit must be programmed to 0. If N pixels are required in this time, then value (N-12) should be programmed into the HBSR. (N must be a multiple of 2).

Note: *This register must always be programmed, even when a border is not required. If a border is not required, then the value in the HBSR must be such as to start the border in the same place as the display start. i.e.* $N_{HBSR} = N_{HDSR}$.

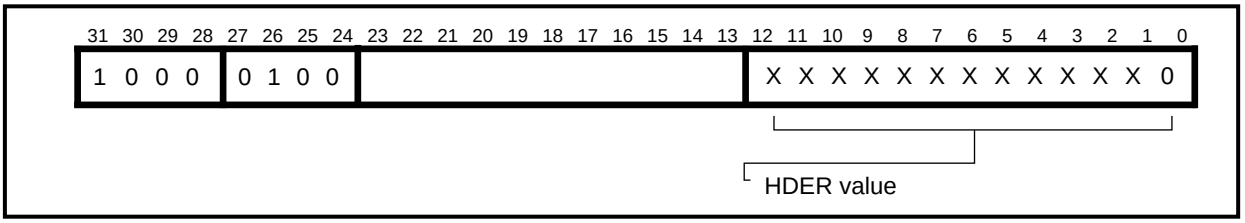
12.10 Horizontal Display Start Register (HDSR): Address 0x83



This register defines the time, in pixels, from the start of the **HSYNC** pulse to the start of the video display.

This is a 14-bit register of which the bottom bit must be programmed to 0. If N pixels are required in this time, then value (N-18) should be programmed into the HBSR. (N must be a multiple of 2).

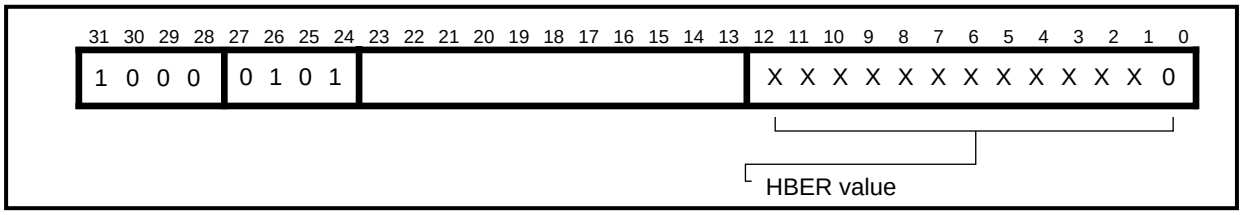
12.11 Horizontal Display End Register (HDER): Address 0x84



This register defines the time, in pixels, from the start of the **HSYNC** pulse to the end of the video display. (i.e. the first pixel which is not display).

This is a 14-bit register of which the bottom bit must be programmed to 0. If N pixels are required in this time, then value (N-18) should be programmed into the HBER. (N must be a multiple of 2)

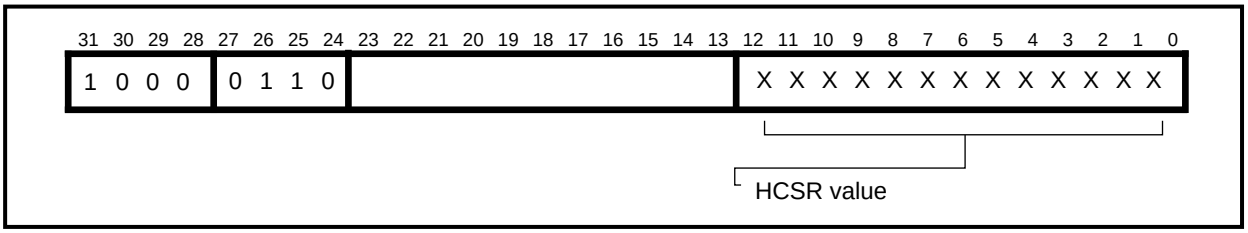
12.12 Horizontal Border End Register (HBER): Address 0x85



This register defines the time, in pixels, from the start of the **HSYNC** pulse to the end of the border display. (i.e. the first pixel which is not border).

This is a 14-bit register of which the bottom bit must be programmed to 0. If N pixels are required in this time, then value (N-12) should be programmed into the HBER. (N must be a multiple of 2). Again, if no border is required, this register must still be programmed such that $N_{HBER} = N_{HDER}$.

12.13 Horizontal Cursor Start Register (HCSR): Address 0x86



This register defines the time, in pixels, from the start of the **HSYNC** pulse to the start of the cursor display.

This is a 14-bit register of which all bits may be programmed. If N pixels are required in this time, then value (N-17) should be programmed into the HCSR. The cursor can thus be programmed to start on any pixel. In HiRes mode, the cursor can still only be programmed to start on a normal pixel boundary. However, because the resolution of the cursor can be defined to a micro-pixel, by using different cursor images it is possible to position the cursor to any micro-pixel.

Note that only the cursor start position needs to be defined, as the cursor is automatically disabled after 32 pixels in normal mode, or 16 pixels in HiRes mode. If a cursor smaller than this is required, then the remaining bits in the cursor pattern should be programmed to logical color 00 (transparent).

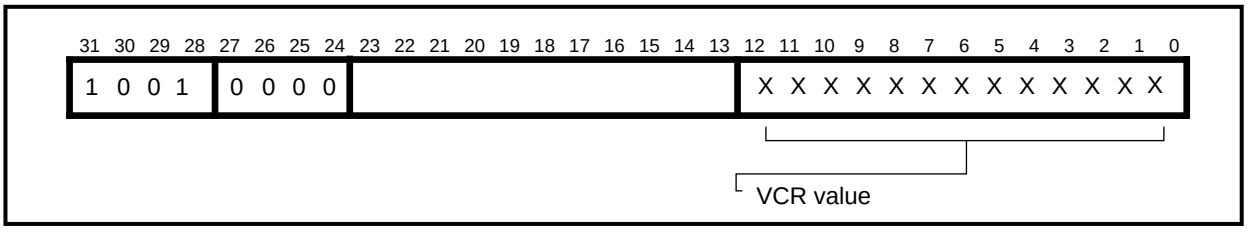
12.14 Horizontal Interlace Register (HIR): Address 0x87

Address 87H is reserved. Do not attempt to program this register.

12.15 Horizontal Test Registers: Addresses 0x88 & 0x8H

Two registers are provided for testing the chip in production. Neither of these registers are intended to be used during normal operation of the device.

12.16 Vertical Cycle Register (VCR): Address 0x90

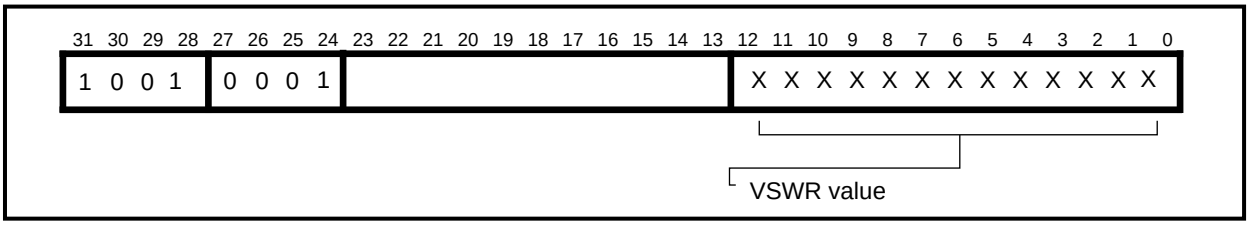


This 13-bit register defines the period, in units of a raster, of the vertical scan; i.e. display time + flyback time.

If N rasters are required in a complete frame, then value (N-2) should be programmed into the VCR.

If an interlaced display is selected, (N-3)/2 must be programmed into the VCR. [N must be odd]. Here N is still the number of rasters in a complete frame, *not* a field.

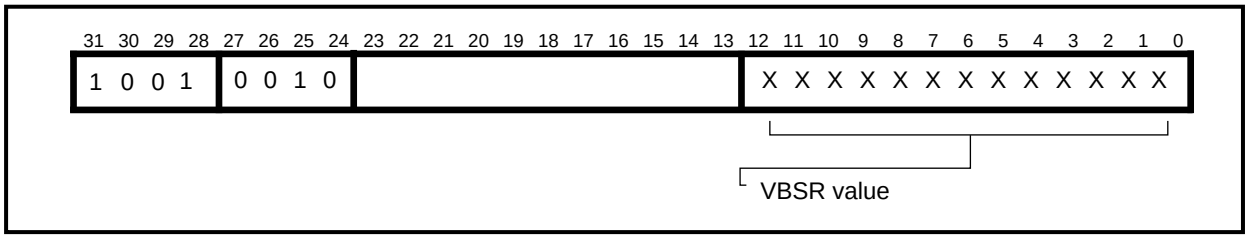
12.17 Vertical Sync Width Register (VSWR): Address 0x91



This 13-bit register defines the width, in units of a raster, of the **VSYNC** pulse.

If N rasters are required in the **VSYNC** pulse, then value (N - 2) should be programmed into the VSWR. The minimum value allowed for N is 2.

12.18 Vertical Border Start Register (VBSR): Address 0x92

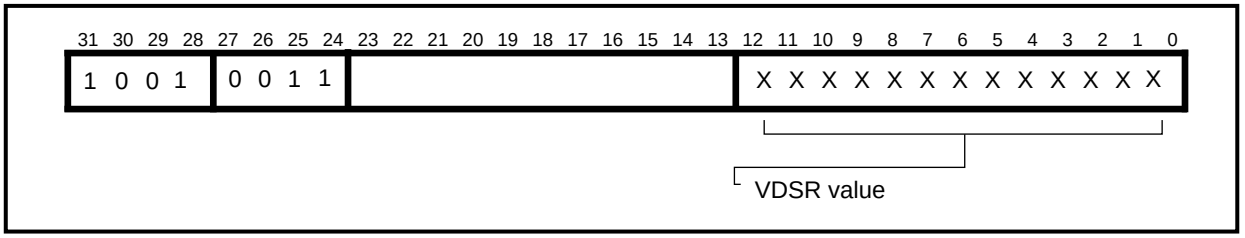


This 13-bit register defines the time, in units of a raster, from the start of the **VSYNC** pulse to the start of the border display.

If N rasters are required in this time, then value (N-1) should be programmed into the VBSR.

If no border is required, this register must still be programmed, in this case to the same value as the VDSR.

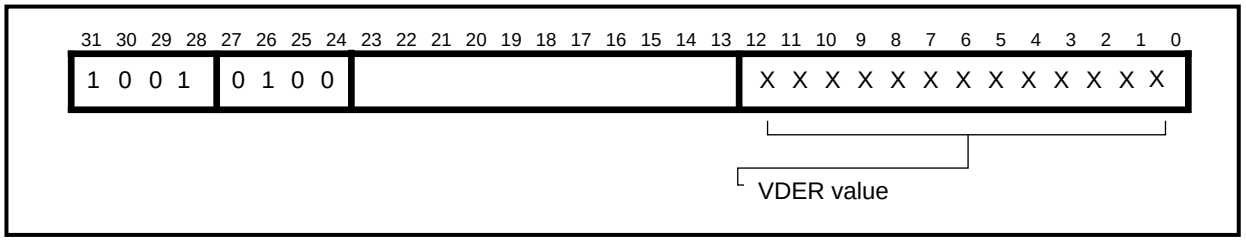
12.19 Vertical Display Start Register (VDSR): Address 0x93



This 13-bit register defines the time, in units of a raster, from the start of the **VSYNC** pulse to the start of the video display.

If N rasters are required in this time, then value (N-1) should be programmed into the VDSR.

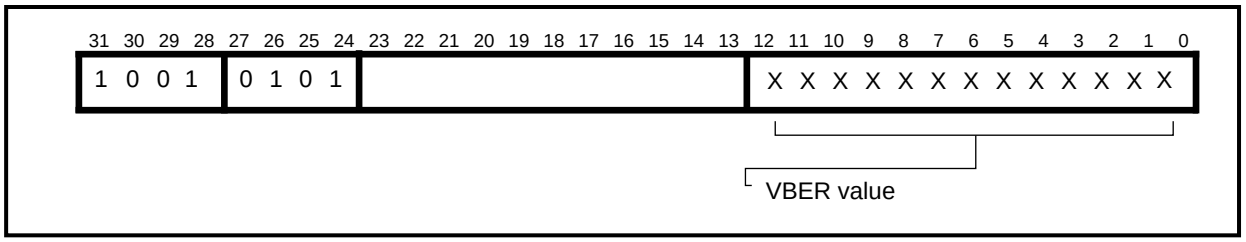
12.20 Vertical Display End Register (VDER): Address 0x94



This 13-bit register defines the time, in units of a raster, from the start of the **VSYNC** pulse to the end of the video display. (i.e. the first raster on which the display is *not* present).

If N rasters are required in this time, then value (N-1) should be programmed into the VDER.

12.21 Vertical Border End Register (VBER): Address 0x95

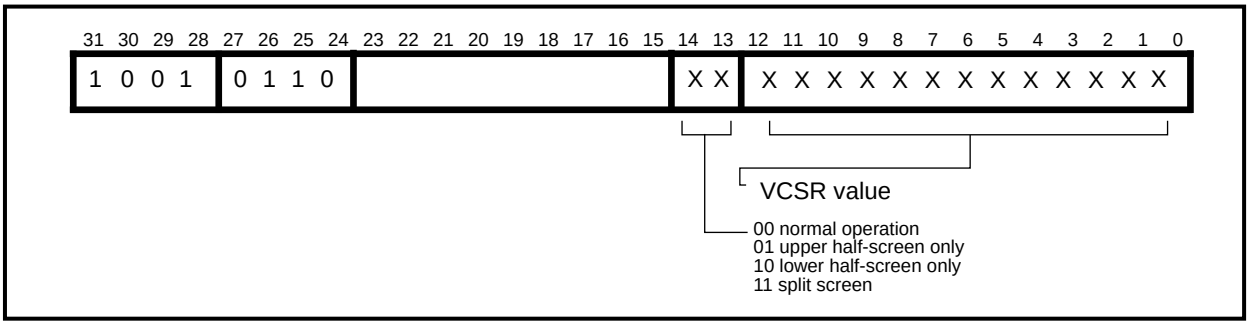


This 13-bit register defines the time, in units of a raster, from the start of the **VSYNC** pulse to the end of the border display. (i.e. the first raster on which the border is not present).

If N rasters are required in this time, then value (N-1) should be programmed into the VBER.

If no border is required, then this register must be programmed to the same value as the VDER.

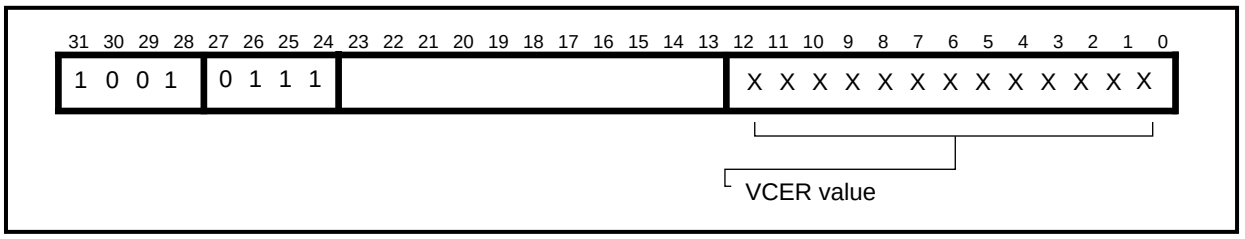
12.22 Vertical Cursor Start Register (VCSR): Address 0x96



This is a 15-bit register. The lower 13 bits define the time, in units of a raster, from the start of the **VSYNC** pulse to the start of the cursor display. If N rasters are required in this time, then value (N-1) should be programmed into the VCSR. The upper 2 bits are used to control the display of the cursor in duplex LCD mode. They should be programmed to zero in all other modes.

When the upper 2 bits are programmed to be 11 (split screen) the meaning of VCSR and VCER are altered as follows. The cursor is displayed in the lower half-screen only from the value of VDSR to the value of VCSR, and again in the upper half screen only from the value of VCER to the value of VDER. This allows a cursor to be positioned across the boundary of the upper and lower half screens of an LCD.

12.23 Vertical Cursor End Register (VCER): Address 0x97



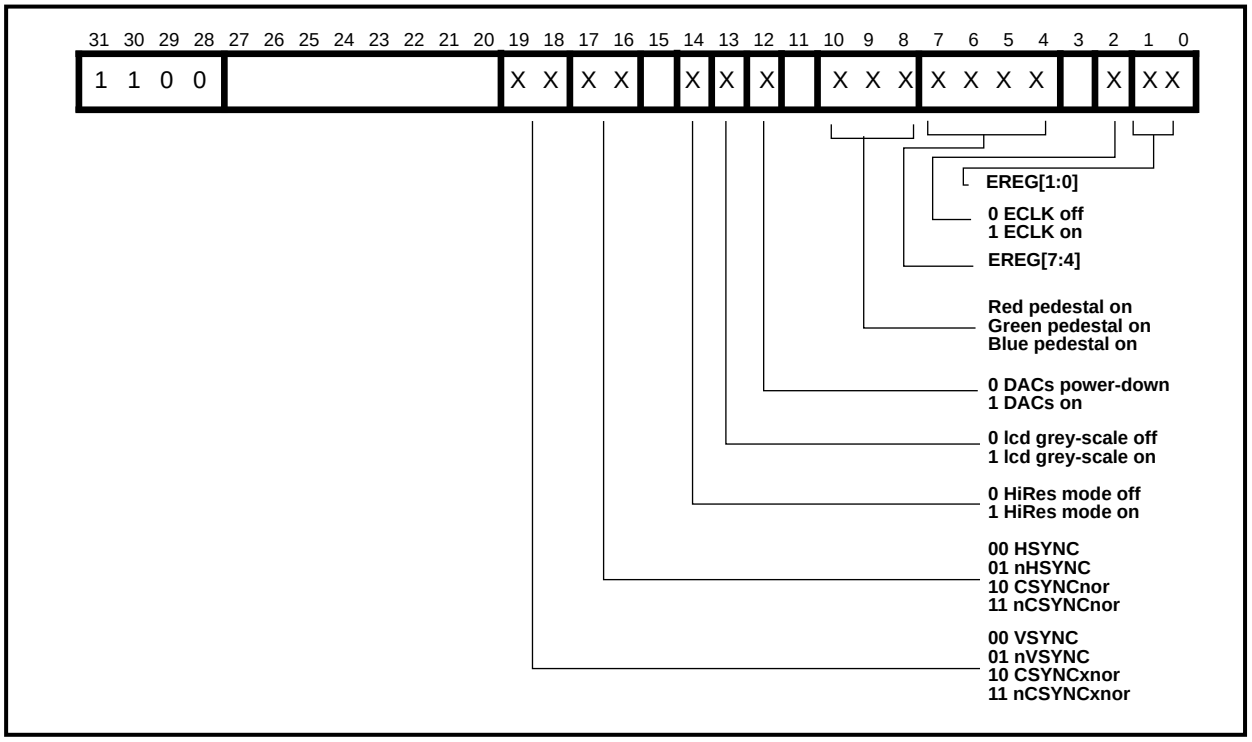
This 13-bit register defines the time, in units of a raster, from the start of the **VSYNC** pulse to the end of the cursor display. (i.e. the first raster on which the cursor is not present).

If N rasters are required in this time, then value (N-1) should be programmed into the VCER.

12.24 Vertical Test Registers: Addresses 0x98, 0x9A & 0x9C

Three registers are provided for testing the chip in production. None of these registers are intended to be used during normal operation of the device.

12.25 External register (ereg): Address 0xC



This register contains the control bits for the external functions of video and sound macrocell. In particular it controls the DACs, the configuration of the External Port **ED[7:0]**, and the configuration of the sync lines.

EREG[1:0] are internally mapped to drive esel[1:0] by ARM7500FE.

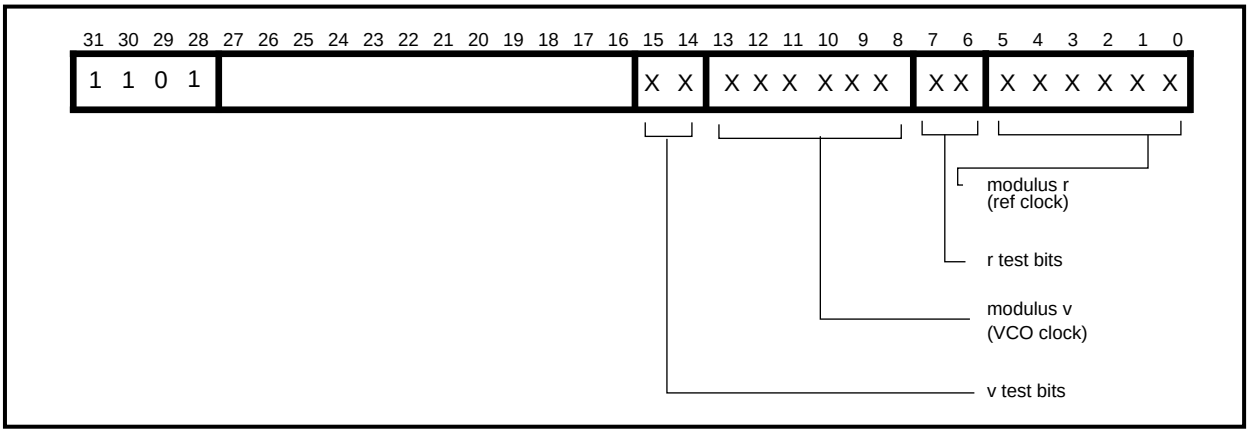
EREG[7:4] are exported from the chip on **ED[7:4]** if EREG[1:0]**=**3. Refer to *14.6 External Support* on page 14-9.

The use of pedon[2:0] and DAC is defined in *14.7 Analog Outputs* on page 14-12.

The uses of lcd and hrm are defined in *14.6 External Support* on page 14-9.

ARM7500FE can export a variety of sync configurations on the pins **HSYNC** and **VSYNC**, as specified by the bits 16-17 and 18-19 respectively. For further explanation see *14.6.3 Vertical and horizontal synchronization* on page 14-11.

12.26 Frequency Synthesizer Register (fsynreg): Address 0xD



The ARM7500FE is able to drive a VCO to provide a suitable input frequency for the pixel clock derived from a reference clock. This is achieved by dividing the reference clock by modulus r, and the VCO clock by modulus v, and comparing the resulting frequencies. Refer to *14.1 Pixel Clock* on page 14-2 for a more detailed explanation. The two moduli, r and v are each 6-bit values, and are programmed in this register.

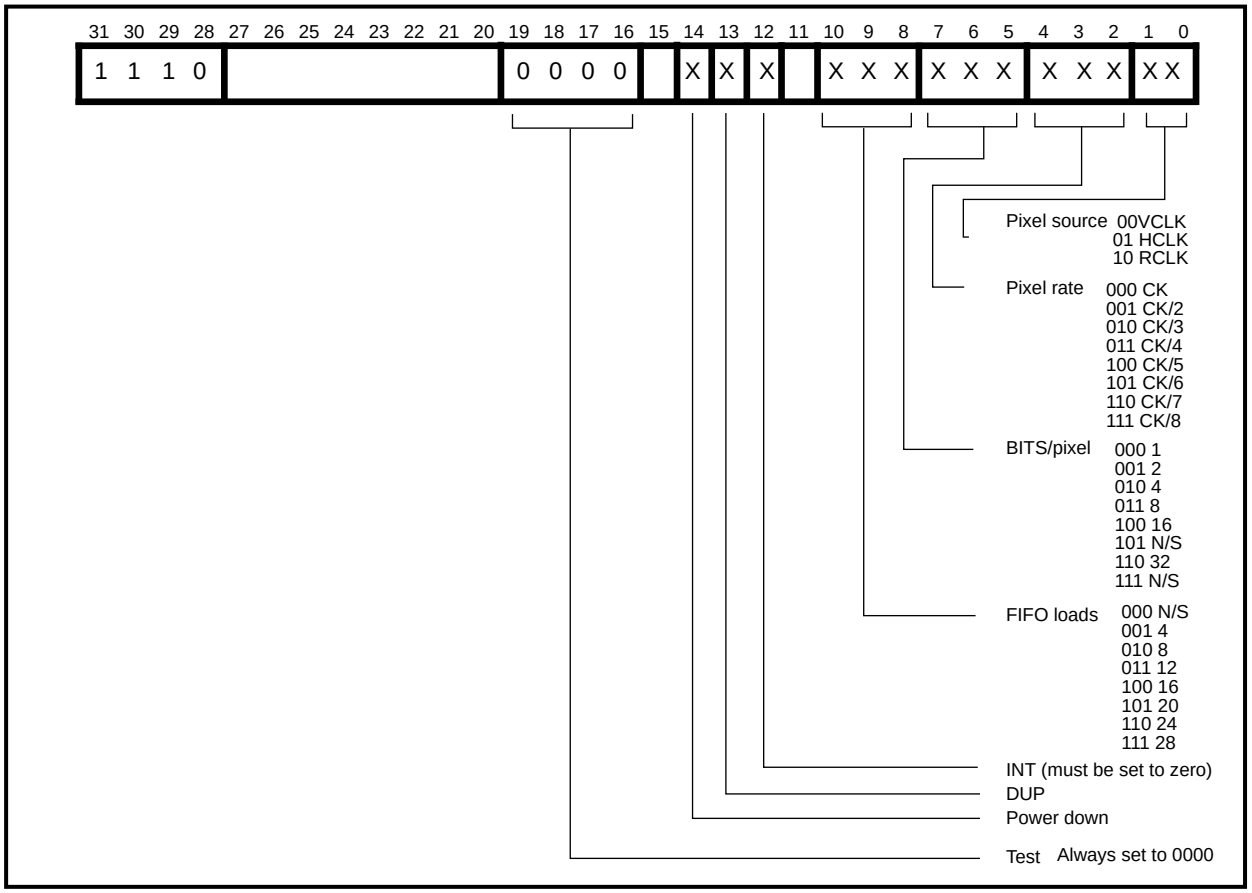
Each counter has 2 associated test bits which should normally be programmed to 0.

Setting bit[6]	forces the phase comparator HIGH, which drives **PCOMP** HIGH.
Setting bit[7]	clears the r-modulus counter.
Setting bit[14]	forces the phase comparator LOW, which drives **PCOMP** LOW.
Setting bit[15]	clears the v-modulus counter.

To reduce power consumption, program this register with large values when the frequency synthesizer is not in use. In particular, bits [6] and [14] should not be set at the same time.

To get a modulus of r, value (r-1) should be programmed into the fsynreg. Likewise for the v-modulus.

12.27 Control Register (conreg): Address 0xE



The main control register determines the basic operation of the chip. In particular the pixel clock source, the pixel rate, the number of bits/pixel, the control of the video FIFO, and the data format are programmed here. In addition there is a 4-bit test register which must be programmed to zero for normal operation.

Note *The INT bit should always be set to zero.*

The pixel clock (pixclk) is selected from one of 3 sources, corresponding to the respective input pins, and the selected clock is then fed through a prescaler as defined by the 3 bits conreg[4:2]. The output of this prescaler is the actual pixel clock. See *Chapter 14: Video Features* for more detail.

The Video FIFO can be programmed to have any number of quad words loaded into it at the start of display. The value chosen should take into account the bandwidth of the display as well as the latency of the DMA subsystem. Refer to *Chapter 13: Video Macrocell Interface* before programming these values.

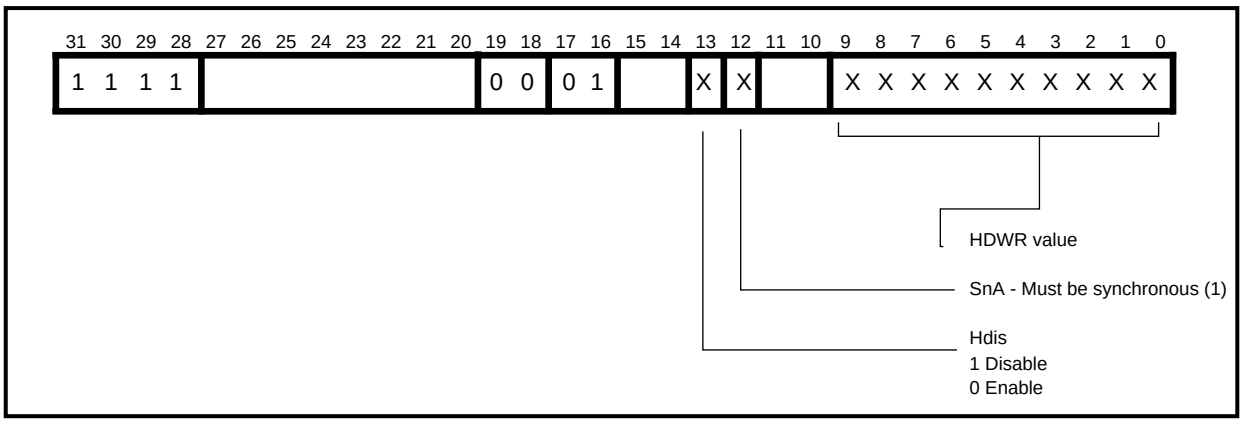
Setting the dup bit configures the display for dual-panel LCDs. This is described further in *Chapter 14: Video Features* .

Note: *After a reset the Control Register should be the first register programmed. The Powerdown bit (14) must immediately be programmed LOW. The test registers bits (16 to 19) also should be programmed LOW, as any other setting will inhibit normal operation.*

The video macrocell uses dynamic logic structures for maximum performance. When the powerdown bit is set HIGH, the main video data path will be set into a state where it will not consume static current. This must be done before the ARM7500FE is set into STOP mode.

12.28 Data Control Register (DCTL): Address 0xF

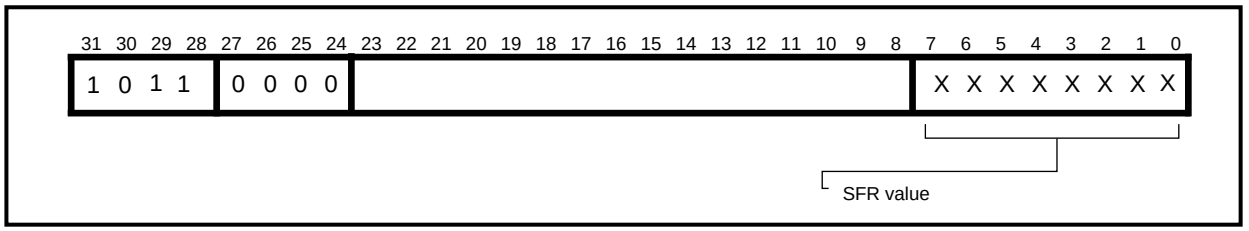
The horizontal display width is also defined in this register, and should be programmed to be the number of words of data in a displayed raster. It must be programmed in most configurations of the device, as it inhibits a DMA request near the end of a raster, when there are enough words in the video FIFO for that raster. The request is uninhibited after the **HSYNC** at the start of the next raster. When driving a dual panel LCD screen, this register must be programmed with twice the number of words in a displayed raster. Hdis should normally be programmed to zero. If Hdis is programmed to one, the inhibition of DMA requests is disabled.



Note *Bits 19:16 MUST be set to 0001 (binary).*

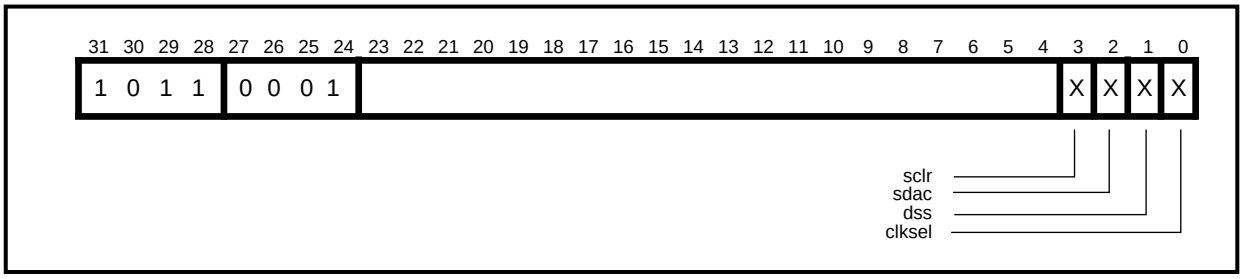
12.29 Sound Frequency Register: Address 0xB0

This 8-bit register specifies the byte sample rate of the sound data. It is defined in units of 1µS. See *Chapter 15: Sound Features* for more detail.



If a sample rate of N µs is required, then N-2 should be programmed into the SFR. N may take any value between 3 and 256.

12.30 Sound Control Register: Address 0xB1



This is a 4-bit register which defines various control bits for the sound system.

Bit 3: SCLR	This bit should always be programmed LOW.
Bit 2:	This bit should be written as zero.
Bit 1: serial sound	This bit is used to select serial sound mode.
Bit 0: CLKSEL	This bit is used to select which clock is used in the sound system. When HIGH, the ARM7500FE's internal 32MHz I/O reference clock is used, when LOW the optional sound clock is used.

13 Video Macrocell Interface

This chapter describes the video macrocell interface within the ARM7500FE.

Video Macrocell Interface

13.1 Bus Interface

The video macrocell does not use the ARM address bus. The address for programming video and sound registers (0x03400000 to 0x034FFFFF) is decoded elsewhere in ARM7500FE and the internal nPROG signal is generated as a general register write strobe. The specific register to be programmed is selected according to the state of the upper bits of the 32-bit input data bus.

All video and sound data is then obtained by DMA under the control of the nVIDRQ internal request signal. This signals to the main ARM7500FE bus arbitration logic that a DMA request is pending, and the request will be serviced at the first available opportunity. All DMA is quad word, so four complete data words will be read from memory and stored in the appropriate video, cursor or sound FIFO for each DMA burst. Note that video DMA may be read from memory in bursts of more than 4 words allowing almost continuous DRAM page mode access to occur.

The system software should create a video frame buffer in DRAM memory, and program the DMA address pointers to the start, end and desired initial location within the buffer. All DMA pointer addresses should be quad word aligned. Once the display has been enabled, video registers should only be programmed during the flyback period to ensure flicker free updating of the screen. See *Chapter 16: Memory and I/O Programmers' Model* for details of how to program the DMA controller.

13.2 Setting the FIFO Preload Value

The Video FIFO is a 32-entry, 32-bit wide FIFO. Words of video data are clocked into the top of the FIFO under control of the internal ARM7500FE signals, BUSCLK and nVIDAK. Words are clocked out of the bottom of the FIFO as the video system displays the data, which is controlled by the pixel clock.

The FIFO is flushed during vertical flyback time, so before the start of the frame the FIFO is empty. At the start of the frame a video request is made to the memory subsystem by asserting the internal ARM7500FE signal, nVIDRQ. When a predetermined number of words have been loaded into the FIFO the request is removed. As the data in the FIFO is displayed, further video requests are made to refill the FIFO to the desired level.

The Control Register includes a 3-bit field (bits 10:8) to set the preload value of the Video FIFO. In this way the FIFO can be programmed to load 4,8,12,16,20,24 or 28 words of data into the FIFO at the start of frame. After the start of frame, the FIFO will request more data when the number of words in it falls below the preloaded value.

The point at which the FIFO should request more data to be loaded is dependent upon system considerations: if the FIFO is reloaded too late, there is a danger that it will run out of data (underflow); if it is reloaded too early, then there is a danger that the data will not fit into the FIFO (overflow). In general, the higher the bandwidth of the screen, then the more words need to be preloaded into the FIFO. In a low bandwidth screen mode, it is not always desirable to have a large preload value, as the bus traffic will have long bursts of data transfer at the start of the frame.

The optimum value to be preloaded depends upon the screen mode in use (i.e. the rate at which data is read from the FIFO), and both the latency of the memory controller and the rate at which data is provided to ARM7500FE. It is generally prudent to program the minimum value possible to keep the bus traffic even.

Let:

n be the value programmed into the control register.

v (words/µs) be the rate at which video data is displayed

Lmax (µs) be the maximum latency in the memory system. (This is the maximum time between ARM7500FE requesting more video data and the memory system delivering the first word of that data.)

If the FIFO is almost empty then it takes 0.025µs for a word of data to reach the bottom of the FIFO before it can be used.

The minimum value for *n* is deduced from the following condition to avoid the FIFO underflowing:

There are 4*n* words in the FIFO when the FIFO requests more data, and if not refilled, then the FIFO would be empty in 4n/v µs.

So *n* must be chosen such that 4*n*/v > (*Lmax*+ 0.025).

The maximum value for *n* is deduced from the following condition to avoid the FIFO overflowing:

n may take the maximum value of 7, and the FIFO can never overflow, as there will always be 4 words available in the top of the FIFO, even if the video request is serviced immediately.

13.2.1 Example

For ARM7500FE, the value of *v* (words/µs) will change depending on the video mode selected and the pixel clock rate chosen, and the worst case DMA latency *Lmax* will alter depending on whether ROM accesses, DRAM accesses or internal programming bursts are slowest, and the MEMCLK frequency used.

The memory subsystems chapter demonstrates how to calculate the worst case DMA latency for a particular system using the ARM7500FE, and the value calculated there should be imported as lmax into the formula in the previous section.

Assume that an 8 bit per pixel mode is being used with a pixel clock rate of 60MHz (period = 16.7ns). In each pixel clock tick, 1/4 of a word will be used, so in a whole µs, 0.25 x 1/0.0167 = 14.9 words will be required.

Hence the value of *n* must be such that:

$4n/v > (Lmax + 0.025)$

So, assuming an *Lmax* value of 1.0µs

$n > 3.74(1.0 + 0.025) \Rightarrow n > 3.83$

So in this case the minimum value for n to prevent FIFO underflow is 4.

14 Video Features

This chapter details the video capabilities available with the ARM7500FE.

14.1 Pixel Clock

The video and sound macrocell is capable of generating a display at any pixel rate up to 120MHz. The pixel clock may be selected from one of three sources, and the frequency of this clock may be further divided down by a factor of between 1 and 8. These attributes are programmed by the lower 5 bits of the control register, CONREG.

If a maximum of three master frequencies are sufficient, then the clock inputs can be used directly. However, it is often a requirement to have many different master clock frequencies. In order to obviate the need for many crystals on the PCB, the video and sound macrocell is designed to drive a Voltage Controlled Oscillator (VCO) to provide the master frequency. The VCO and filter are external to ARM7500FE, but everything else is built into the chip. Operation is described below:

An internal reference frequency of 32 MHz is supplied via the **I_OCLK** input of ARM7500FE. The signal from the VCO is input into ARM7500FE on the pin **VCLKI**. **VCLKO** is simply the inverse of **VCLKI**, and this may be used to bias the input signal about the threshold if the VCO output is not a full amplitude signal. The mark-space ratio of the VCO output should be as close as possible to 50-50 if operation at 120MHz is to be achieved.

The reference clock is divided by a programmable number set by the r-modulus in the fsynreg. The VCO clock is divided by a programmable number set by the v-modulus in the fsynreg. Each of the moduli may be a 6 bit number. The output of each of these dividers is fed into a phase comparator, and the result is output from ARM7500FE as **PCOMP**. This pin should then be filtered and used to control the VCO output frequency. In this way, the VCO can be set to have a frequency of v/r * F_{ref}.

The phase comparator is of the phase-frequency type. The output **PCOMP** is normally tri-state, but when the VCO frequency needs to be decreased the output is LOW, and when the VCO frequency needs to be increased the output is HIGH. When the 2 frequencies are in lock, **PCOMP** will normally be tri-state, but will be driven to the midpoint for a very short time (a few ns) every r/F_{ref}+ period. The output impedance of this pin when it is driven is about 50Ω. *Figure 14-1: ARM7500FE internal subsystems for pixel clock generation* on page 14-3.

The choice of filter and VCO is left to the user. It is important to avoid any low-frequency modulation of the VCO frequency. It has been found that a suitable VCO is a 74AC04 inverter element with feedback, with the supply voltage controlled by the **PCOMP** output. (See *Appendix E: ARM7500FE Video Clock Sources*.)

With this approach, an enormous number of frequencies are possible. The 32MHz reference frequency generated within ARM7500FE can be used to yield the following common VCO frequencies in the table on the next page. For some frequencies, there are many possible values of r and v. In this case it is sensible to choose a set of values which favors the filter response. (Remember large moduli yield a lower comparison frequency).

It may be best to limit the VCO range, and use the prescaler within video and sound macrocell to get a lower pixel rate than the VCO frequency. It is expected that the VCO range may have to be constrained so that it cannot provide the highest frequencies at which the video and sound macrocell can operate. In this case, a single high-frequency clock can be fed into ARM7500FE on the **HCLK** pin, and this can be selected for the pixel clock.

r-modulus	v-modulus	VCO frequency/MHz
8	2	8.0
16	6	12.0
4	2	16.0
8	6	24.0
2	2	32.0
8	9	36.0
16	35	70.0
4	15	120.0

Table 14-1: Synthesized VCO frequency settings

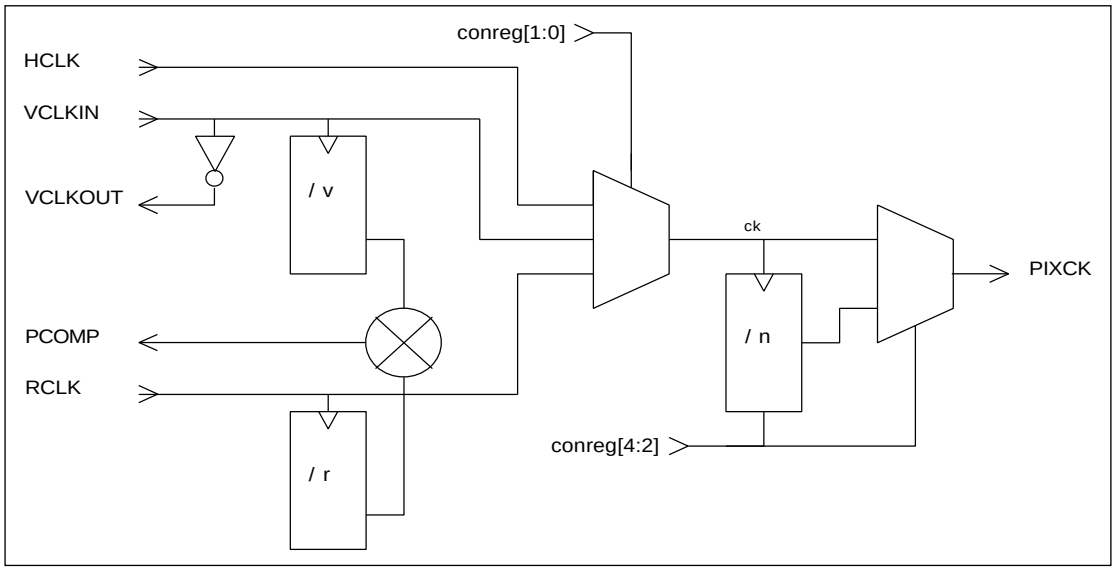


Figure 14-1: ARM7500FE internal subsystems for pixel clock generation

14.2 The Palette

ARM7500FE has a 28-bit wide 256-entry palette which is constructed out of three 8-bit wide look-up-tables (LUTs), each with 256 entries, named Red, Green, and Blue, and one 4-bit wide LUT with 16 entries, named Ext. The Red, Green and Blue LUTs each drive their respective DACs, and the Ext LUT is normally configured to drive the **ED[3:0]** output port, except when Hires mode or LCD mode is selected. These bits may be used outside the chip for a variety of purposes such as supremacy, fading, HiRes and LCD driving. The **ED[7:4]** output port is normally driven from the Ext register, ereg[7:4], which may be written at any time, so these bits can be used as a DC control port.

The mapping of the logical colors through the LUTs is dependent on the mode in use, as follows:

- In 1,2,4 bits/pixel modes, the logical data is fed simultaneously to all 4 LUTs. This gives a fully flexible palette with any logical color being mapped to any physical color, and any **ED[3:0]** value. The palette will give 16 colors from a selection of 2^{24}.
- In 8-bits/pixel modes, the logical data is fed simultaneously to all 4 LUTs. This gives a fully flexible palette with any logical color being mapped to any physical color. Logical colors 0-15 access the Next LUT, and logical colors 16-255 access location 0 of the Ext LUT. The Ext LUT again drives **ED[3:0]**. The palette will give 256 colors from a selection of 2^{24}.
- In the 16-bits/pixel mode, a patented technique has been developed. This approach is highly flexible and allows many different addressing modes e.g. 5-5-5, 5-6-5 etc. In this mode 2^{16} colors are available from a selection of 2^{24}.
- In the 32-bits/pixel mode, 24 bits from the logical field will drive the 256 entries in each of the color LUTs (8 bits to each LUT) and 4 bits will drive the Ext LUT. The upper 4 bits are discarded.The palette will give the full range of 2^{24} colors.

Note that where a logical field does not drive all the palette entries (such as in 4 bits/pixel mode) only the lower part of the palette is used. Unused sections need not be programmed.

When HiRes mode or LCD mode is selected, the palette must be set up in a predetermined configuration. This is explained in the chapters on hi-res support and LCDs.

14.2.1 Palette updating

A signal FLYBK exists within ARM7500FE as an output from the video and sound macrocell. FLYBK goes HIGH at the start of the first raster which is not displayed, and goes LOW at the start of the first raster which is displayed. The rising edge of this signal can cause an interrupt via the ARM7500FE IRQA interrupt registers, and the palette should be updated at this time for flicker-free updating.

14.3 Cursor

ARM7500FE has a hardware cursor 32 pixels wide and any number of pixels high. Its 2 bits per pixel allow 4 colors, which include “transparent” plus three other colors from a selection of 2^{24}. It is possible to display the cursor in the horizontal border, but not in the vertical border.

The cursor has a 3 entry palette which is 28 bits wide, allowing each cursor logical color to be any physical color. In addition, there is a 28 bit wide border color register.

At the start of every frame, 16 bytes of cursor data are transferred to the video subsystem during the horizontal retrace period. This is enough data for two raster's worth of cursor. After they have been displayed, a request is made for another 16 bytes. Thus, in normal mode, requests are made on every other raster on which there is cursor, and enough data is transferred for two rasters each. In Hi-Res mode, a request is made every raster. Note that the cursor data is always transferred in bursts of four words.

14.3.1 Cursor in hi-res mode

In order to allow micro-pixel resolution of the cursor in Hi-Res mode when operating at 4 micro-pixels per normal pixel, it is necessary to define 2 bits per micro-pixel, or 8 bits per normal pixel. The 16 bytes of cursor data available for each raster can thus generate 64µ-pixels of cursor. In Hi-Res mode the cursor palette is not used (though the border may be programmed). Refer to the chapter on Hi-Res support.

The cursor is always positioned to align with a normal pixel. In order to position the cursor to a µ-pixel horizontally, four different copies of the cursor are required: each copy defines the cursor offset by a single µ-pixel. It is possible to define transparency to a resolution of a µ-pixel, so by selecting the correct cursor image, the required position can be achieved.

14.3.2 Cursor in LCD mode

The video subsystem is capable of displaying the hardware cursor in LCD mode. However, because of the split-screen nature of duplex LCDs, the cursor needs special attention. If the cursor is entirely in the upper or lower half-screen, then the cursor should be programmed as normal, but VCSR[14:13] should be programmed accordingly (0x10 = upper half-screen; 0x01 = lower half-screen). If the cursor “straddles” the split screen, then the cursor image in memory must start at the top of the lower half-screen, and end with the bottom of the upper half screen. Hence two contiguous images of the cursor image are required, and the start pointer moved accordingly. In practice, four images of the cursor are required, to ensure that a resolution of one raster is maintained across the boundary. As the cursor moves from one panel to the other, the pointer to the cursor image in memory must be moved. For more details, refer to *Appendix B: Dual Panel Liquid Crystal Displays*.

In the case where the cursor straddles the split screen, the meaning of the VCSR and VCER registers are changed. The VCER register now defines the start of cursor in the upper half-screen, and the VCSR defines the end of the cursor in the lower half-screen. Thus the cursor is actually displayed in the lower half-screen from the start of display until VCSR, and then again in the upper half-screen from VCER

until the end of display. This mode is selected by programming VCSR[14:13] = 0x11. Further details of how to use ARM7500FE with dual panel LCD screens are given in *Appendix B: Dual Panel Liquid Crystal Displays*.

14.4 Hi-Res Support

ARM7500FE is able to support color screens with resolutions above 1024 by 768 pixels. For higher resolutions, externally serializing the data is required to produce monochrome (or grey-level) pictures. In this scheme one 16ns-pixel could theoretically be serialized to make eight 2ns-pixels, ie. about 500MHz. However, this is dependent on the availability of external hardware capable of generating a serial bitstream at this frequency.

14.4.1 ARM7500FE support for hi-res mode

When the hrm bit in the Ext register is set, and EREG[1:0] is set to value 0x10, ARM7500FE outputs 8 bits of data for every normal pixel on the **ED[7:0]** port. These bits can then be serialized to form a high frequency monochrome pixel stream; alternatively they can be serialized to 2 or 4 bits, which could then drive a high-speed monochrome DAC for grey level displays. With the pixel clock running at a fundamental frequency of about 100MHz, the external serial clock could be running at up to several hundred MHz. In order for the external circuit to be able to synchronize to the ARM7500FE output data, ARM7500FE also outputs a pixel clock synchronous to the data stream when the hrm bit is set.

In this mode, with EREG[1:0] set to value 0x10, the video data is driven from the Blue LUT, which outputs data BPD[7:0]. Depending on how the external serializer circuit is arranged, the LUT must be set up to give a one-one correlation between the logical address and the physical data value. So, for example, if 4 bits are externally serialized into a single bit stream, then 4 bits/pixel mode should be selected, and **ED[6,4,2,0]** should be used. The lower 16 words of the Blue LUT should be programmed to give all 16 combinations of BPD[6,4,2,0]. If 8 bits are externally serialized to give a single bit-stream, then 8 bits/ pixel mode should be selected, and all 256 values of the Blue LUT should be programmed as a one-one mapping.

Hardware cursor support is provided as follows. The cursor palette is not used, though the Blue border may be programmed. Eight bits of cursor data (CD[7:0]) are defined for each normal pixel. The 8 bits are divided into 4 pairs, with the lsb (least significant bit) of each pair defining whether the video data (BPD) or the msb (most significant bit) of the cursor pair is displayed. Each cursor bit-pair operates on 2 bits of the video data (BPD) according to the following tables.

So if the external circuit serializes **ED[6,4,2,0]** into a single bit stream, or **ED[7:0]** into a 2-bit data stream then the cursor can be positioned and defined to any micro-pixel: in each case the cursor can be transparent, black or white. If all 8 bits are serialized into a single very high frequency bit stream, then the cursor can only be positioned and defined to units of 2 micro-pixels.

CD[7]	CD[6]	ED[7]	ED[6]
0	0	BPD[7]	BPD[6]
0	1	0	0
1	0	BPD[7]	BPD[6]
1	1	1	1

Table 14-2: Deriving high-speed 2-bit cursor data from the normal 8-bit output—CD[6&7]

CD[5]	CD[4]	ED[5]	ED[4]
0	0	BPD[5]	BPD[4]
0	1	0	0
1	0	BPD[5]	BPD[4]
1	1	1	1

Table 14-3: Deriving high speed 2-bit cursor data from the normal 8-bit output - CD[4&5]

CD[3]	CD[2]	ED[3]	ED[2]
0	0	BPD[3]	BPD[2]
0	1	0	0
1	0	BPD[3]	BPD[2]
1	1	1	1

Table 14-4: Deriving high-speed 2-bit cursor data from the normal 8-bit output—CD[2&3]

CD[1]	CD[0]	ED[1]	ED[0]
0	0	BPD[1]	BPD[0]
0	1	0	0
1	0	BPD[1]	BPD[0]
1	1	1	1

Table 14-5: Deriving high speed 2 bit cursor data from the normal 8 bit output - CD[0&1]

14.5 Liquid Crystal Displays

ARM7500FE is capable of driving single panel Liquid Crystal Displays at 1, 2, 4, 8, 16 or 32 bits per pixel, and dual panel LCDs at 1, 2 or 4 bits per pixel. Grey-scaling is provided at up to 16 shades. ARM7500FE is also capable of driving single panel color LCDs with no grey scaling in its normal (video) mode. Two control bits are provided for LCD operation:

lcd (bit 13 in the Ext register) configures the external data port **ED[7:0]** for LCD operation, and enables the grey-scaling logic (EREG[1:0] must be set to 0x01);

dup (bit 13 in the control register) enables duplex mode, and should be set for dual-panel LCDs.

14.5.1 LCD grey-scaling

To obtain a grey-scaled output from ARM7500FE, the lcd bit (bit 13 in the Ext register) must be set. This configures the External port for LCD operation. The DACS should be disabled to save power since ARM7500FE cannot drive both CRT and LCD displays simultaneously. In order to get this data out of the **ED[7:0]** port, EREG[1:0] must be set to value 0x01.

ARM7500FE provides a grey-scaling algorithm which modulates the data output. Grey-scaling is possible at 1, 2 or 4 bits per pixel. The data is output from the chip as one or two 4-bit quantities, depending on whether single or dual panel LCDs are used, at one quarter of the pixel rate. The lower 4 bits of the Green LUT control the upper panel (**ED[7:4]**), and the 4 bits of the Ext LUT control the lower panel (**ED[3:0]**). Thus, the palette can still be used to provide a mapping of logical to physical color. The cursor palette is used similarly, though the programming of the cursor position needs special treatment - refer to Appendix B. If a single panel LCD is used, **ED[7:4]** should be used, and the Green LUT programmed accordingly (**ED[3:0]** are held low in this mode). The grey-scaling logic lies between the output of the video multiplexer and the external port and works as described below.

There are effectively 16 physical grey levels available, and in 1,2, or 4 bits per pixel mode the palettes are programmed to give a mapping of the logical color to physical shade. The resultant 4 bit pixel value out of the video multiplexer is modulated according to its value and the raster number and the point on the raster at which it is generated. The result is a single bit which on average is HIGH for a time equal to the actual 4-bit value. For a single panel screen, 4 of these bits are then collected together and output as a nibble at one quarter of the pixel rate on **ED[7:4]**. **ED[4]** represents the 4th pixel, and **ED[7]** represents the 1st pixel.

If duplex mode is selected, then the pixel stream for the upper half screen is obtained from the Green LUT and that for the lower half screen is obtained from the Ext LUT. Both these pixel streams are passed through the grey-scale logic simultaneously and output as two nibbles on **ED[7:4]** (upper half screen) and **ED[3:0]** (lower half screen).

14.5.2 Dual panel LCDs (duplex mode)

Duplex mode is configured by setting the dup control bit as well as the lcd control bit. The screen parameters are set up according to the requirements of the LCD panel.

Note: *Since the upper and lower panels are driven simultaneously, ARM7500FE only produces data for half the total number of lines on the dual panel. Thus the vertical registers must be programmed as if there were only one panel.*

ARM7500FE requests data in units of two quad-words. The first quad word the memory controller delivers is for the upper half-screen, and the second quad-word is for the lower half-screen. ARM processor then serializes the data into two simultaneous bit-streams as described above. 1, 2 or 4 bits/pixel may be selected.

For details of the ARM7500FE register programming requirements for duplex DMA, see *Chapter 16: Memory and I/O Programmers' Model* .

14.5.3 Single panel color LCDs

If neither dup nor lcd control bits are set, then the **ED[7:0]** port may be used to gain access to all of the physical bits out of the video multiplexer. This would allow many other types of display to be driven.

14.6 External Support

ARM7500FE has an 8-bit output port, **ED[7:0]** and a synchronous clock, **ECLK**, which have different functions in different modes. The port is controlled by the 2 bits, EREG[1:0], in the control register that essentially select which of the bytes from the video multiplexer are chosen. Additionally, an ARM7500FE register bit (bit 1 of the VIDMUX register) can be used to cause the data selection for the **ED** port to be modified according to the state of the **ECLK** output. This feature is intended to be used to increase the bandwidth for driving color LCD screens. When this control bit is set LOW, the behavior of the **ED** port is as shown below. The bit is intended to be used with 'LCD' set LOW. When the VIDMUX bit is HIGH, and EREG[1:0] is set LOW, if **ECLK** is LOW, the Red LUT is output on **ED[7:0]**. If **ECLK** is high, the Green LUT is output on **ED[7:0]**.

When EREG[1:0] = 0:

the Red LUT is output on **ED[7:0]**.

When EREG[1:0] = 1:

if lcd = 0, the Green LUT is output on **ED[7:0]**.

If lcd = 1, the grey-scaled LCD signals are output. **ED[7:4]** carries the data for the upper half screen from the Green LUT, and **ED[3:0]** carries the data for the lower half screen from the Ext LUT.
Note that if lcd = 1, data is output at one-quarter of the ARM processor pixel rate, since the data output actually represents 4 pixels for each half-screen.

When EREG[1:0] = 2:

if hrm = 0, the Blue LUT is output on **ED[7:0]**.

If hrm = 1, the multiplexed Blue LUT and HiRes cursor data is output on **ED[7:0]**. See *14.4 Hi-Res Support* on page 14-6.

Also, **ED[7:0]** is re-timed, and delayed by one extra pixel.

When EREG[1:0] = 3:

if dac = 0, **ED[3:0]** are driven by the Ext LUT, and **ED[7:4]** are driven by the value of the Ext Register,EREG[7:4], which is intended as a DC control port in this mode.

If dac= 1, **ED[3:0]** are delayed by one pixel, so that they are exported from the chip in the same pixel as the analog data to which they correspond. In this configuration **ED[3:0]** bits may be used for supremacy, for overlaying pictures on a pixel-by-pixel basis. Because several bits are output, analog fading and mixing on a pixel basis is possible.

14.6.1 ECLK

ECLK is output along with the data **ED[7:0]**, so that the data can be externally latched and multiplexed. **ECLK** is controlled by lcd and EREG[2]. If EREG[2] = 0, then **ECLK** is output as logic 0. This should be configured whenever **ECLK** is not required, in order to save power. If EREG[2] = 1, then if lcd = 0, **ECLK** is the pixclk, output synchronously with the data stream. If lcd = 1, then **ECLK** is the LCD clock, which runs at a quarter of the pixel rate. The lcd clock is only enabled whilst horizontal display data is being output and is synchronous to the data stream. The timing diagrams below show the relationship between **ED** and **ECLK.**

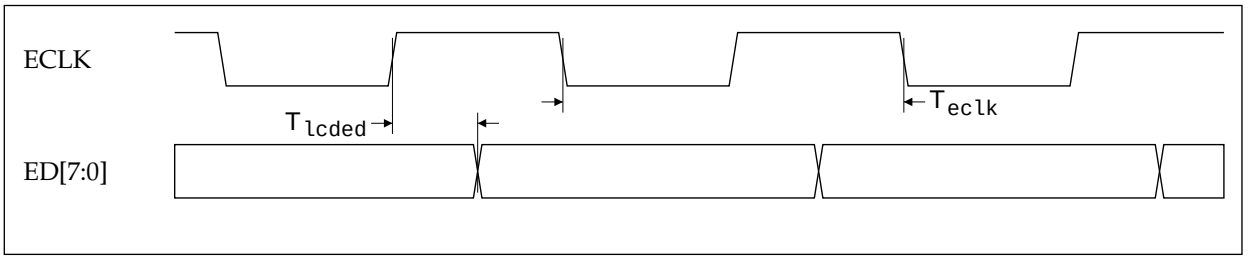


Figure 14-2: Timing relationship between ECLK and ED in LCD grayscale mode

Figure 14-3: Timing relationship between ECLK and ED in all other modes

Symbol	Parameters	Min	Max	Units	Notes
Ted	**ECLK** to **ED** delay	5	7	ns	1
Tlcded	**ECLK** to **ED** delay—LCD mode	Teclk/4 + 5	Teclk/4 + 7	ns	

Table 14-6: ARM7500FE ECLK and ED timing

Note 1: **ECLK** mark space ratio is not always 1:1, depends on pixel clock divide.

14.6.2 Power-saving considerations

The External Port can consume a lot of power, but steps may be taken to minimize power usage. In particular, it is very important not to load the signals heavily, especially **ECLK** which can clock at the pixel rate. When it is not in use, it should not be putting out the raw pixel data, but should be outputting static signals. This is done by selecting EREG[1:0] = 3, and setting all entries of the Ext LUT to be all one value. **ECLK** should be turned off by setting EREG[2] = 0.

If an LCD is fitted, but not operated, it may be necessary to power down the input signals to it. This can be achieved by setting bit 13 low, which disables the grey scaler, and by disabling the external port as described above.

14.6.3 Vertical and horizontal synchronization

Software control over the polarities of the synchronization pulses is provided. Two types of Composite Sync may be output, each of either polarity. The logical OR of Hsync and Vsync may be output on the Horizontal Sync (**HSYNC**) pin, and the XOR of Hsync and Vsync may be output on the Vertical Sync (**VSYNC**) pin. Equalization pulses in the composite synchronization signal are supported for interlace mode. When LCD mode has been selected, the external **HSYNC** and **VSYNC** pulses are modified in accordance to the requirements of an LCD screen.

The **HSYNC** and **VSYNC** pins are programmed with the Ext Register, EREG[19:16].

14.6.4 Genlocking

Genlocking is supported by ARM7500FE. A pin is provided to reset the vertical counter to the first raster (**SYNC**).

14.7 Analog Outputs

ARM7500FE outputs analog R, G, and B signals. It is designed to drive doubly-terminated 75Ω lines directly.

14.7.1 DAC control

There are 4 control bits in the Ext Register associated with the DACs. These are dac and ped[2:0].

Power-save mode

When dac is HIGH, the DACs are all enabled and will generate a current proportional to the digital values from the video multiplexer. When dac is LOW, the reference current into all three DACs is turned off, so the DACs generate no output current, and hence consume much less power. This is useful when operating in LCD mode, or at any time when the screen should be blanked.

Pedestal current

The DACs may be programmed to generate a pedestal offset of 20 lsb equivalent currents. These are controlled individually by pedon[2:0], though they will typically all be programmed on or off together, depending on the monitor characteristics. pedon[0] controls the red pedestal, pedon[1] the green pedestal, and pedon[2] the blue pedestal. If pedon[n] is HIGH, the pedestal current is switched on as the border starts, and is turned off as the border ends.

14.7.2 Video DAC currents

The DACs are each 8 bit resolution, so they source 256 units of current according to the digital value from the video multiplexer. The current step is set by a common reference current, **VIREF**. The recommended reference current is 0.56mA which gives a DAC step of 69µA. Hence digital value 0 gives 0 current and digital value 0xFF gives an output current of (255 * 69)=17.6mA. If pedon is set, then during display time, digital value 0 will generate (20 * 69)=1.38mA, and digital value 0xFF will generate (275* 69)=18.98mA. A 3.4kΩ resistor connected between **VIREF** and **VDD** will provide the desired 0.56mA at about 3.0V; the actual value of resistor may need to be adjusted to obtain the required video output levels.

DAC accuracy

At 120MHz the DACs are accurate to 8 bits absolute resolution. They will always be monotonic.

14.7.3 Monochrome output

ARM7500FE does not generate a separate composite monochrome signal. This can be generated by resistively mixing the R,G and B externally, if required.

15 Sound Features

This chapter details the sound capabilities available with the ARM7500FE.

Sound Features

15.1 Sound

The video and sound macrocell has a digital sound system. This is a 32-bit serial sound interface suitable for driving external CD DACs.

15.2 The Sound FIFO

At the core of the sound system is a 4-word FIFO and a byte-wide latch. When empty, the FIFO fills completely by a DMA request. Data is then clocked out of the FIFO, one byte at a time through the latch.

15.3 The Digital Serial Sound Interface

The serial sound interface offers a high quality 32-bit stereo sound, needing only a small amount of external circuitry. The serial sound system consists of a three-pin serial interface:

SDCLK is the Serial Data Clock output

SDO is the Serial Data output

WS is the Word Select output

When no sound is required, (sctl[2:1]=0), these outputs are stable (**SDCLK**=0, **SDO**=0, **WS**=1).

When in this mode, bytes from the sound FIFO are output in most-significant first order. This is because the serial sound output must go msb first to be compatible with other serial sound devices. Each byte of data is loaded into a parallel-in, serial-out register, and clocked out under control of the bit clock.

15.3.1 Timing formats

There are two timing formats available for the interface:

- normal
- Japanese

The selection of these is controlled by bit 0 of the VIDMUX register in the main part of ARM7500FE.

Normal format

When configured for normal mode (VIDMUX bit 0=LOW), each 32-bit sample consists of 16 bits for the left hand channel, and 16 bits for the right hand channel.
To distinguish between them, a 'word select' (**WS**) signal is produced. This signal changes when the lsb of the previous word is output. When **WS** is HIGH, the right-hand channel is being output.

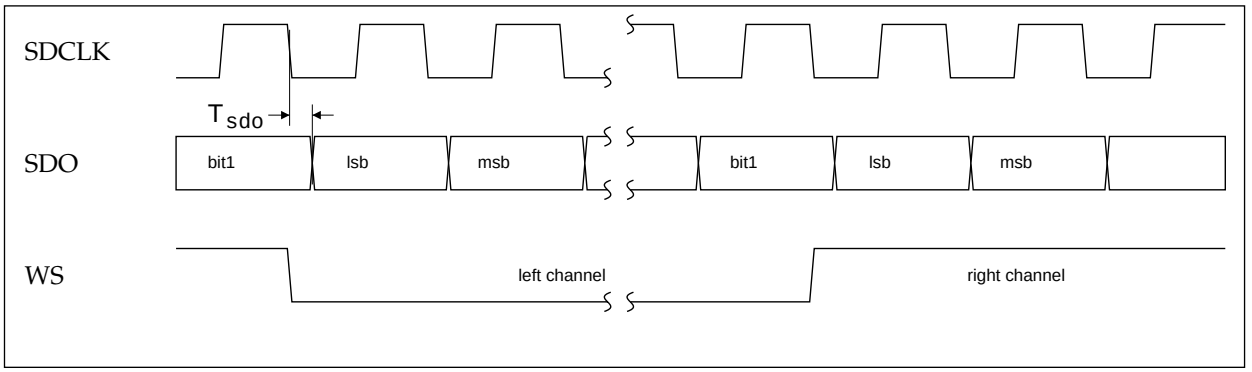


Figure 15-1: Serial sound output — normal format

Japanese format

In Japanese format, the **WS** signal changes when the msb of the new word is output. In addition, the polarity of **WS** is reversed. This is shown in the diagram below.

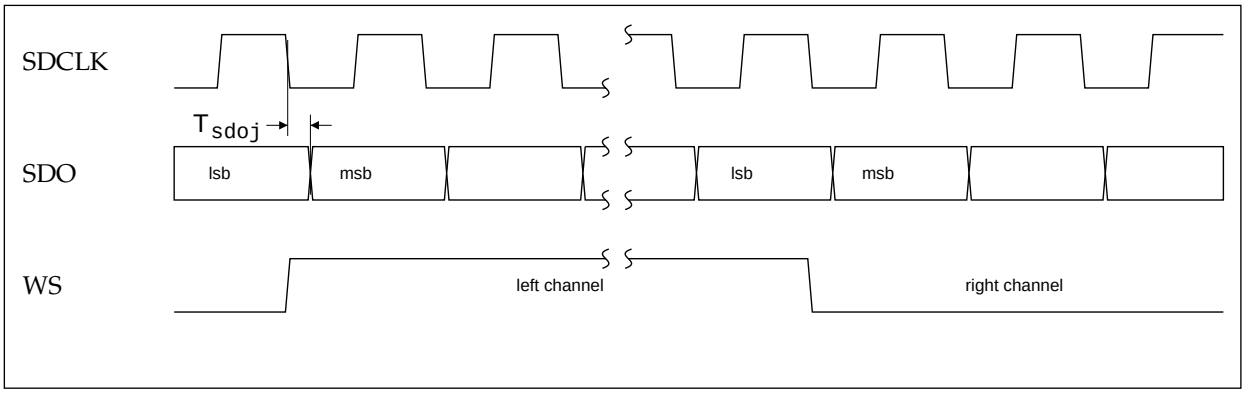


Figure 15-2: Serial sound output — Japanese format

Symbol	Parameter	Min	Max	Units
Tsdo	SDCLK falling to SDO valid (normal format)	0	5	ns
Tsdoj	SDCLK falling to SDO valid (Japanese format)	0	5	ns

Table 15-1: Sound output timing

15.3.2 Using external SCLK input

The serial sound output can be used with any DAC with a serial sound input. Many DACs require a 11.2896MHz input clock, and to reduce the number of on board crystals required, the video and sound macrocell can cope with this frequency on the **SCLK** input. When using this, the following parameters need to be programmed in the registers.

serial sound (SCTL Register bit 1)	1
clksel (SCTL Register bit 0)	0
Sound Frequency Register	2

The sound system is not limited to operating with this frequency alone; however, the Sound Frequency Register must be set to produce the necessary bit rate accordingly.