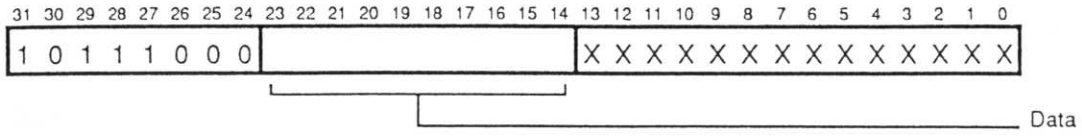
5.20 Vertical Cursor Start Register (VCSR) : Address B8H

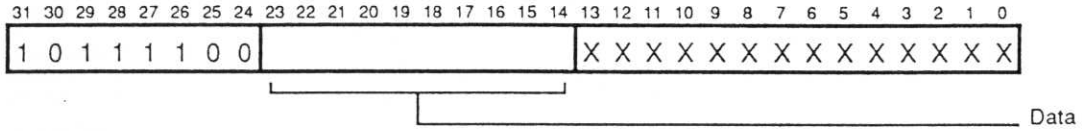


This register defines the time, in units of a raster, from the start of the VSYNC pulse to the start of the cursor display.

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

This is a 10 bit register, with bit 14 the least significant.

5.21 Vertical Cursor End Register (VCER) : Address BCH

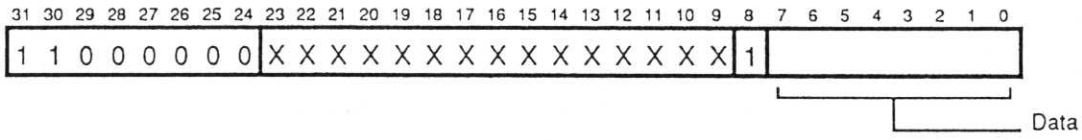


This register defines the time, in units of a raster, from the start of the VSYNC pulse to the end of the cursor display. (ie. the first raster on which the cursor is *not* present).

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

This is a 10 bit register, with bit 14 the least significant.

5.22 Sound Frequency Register (SFR) : Address C0H

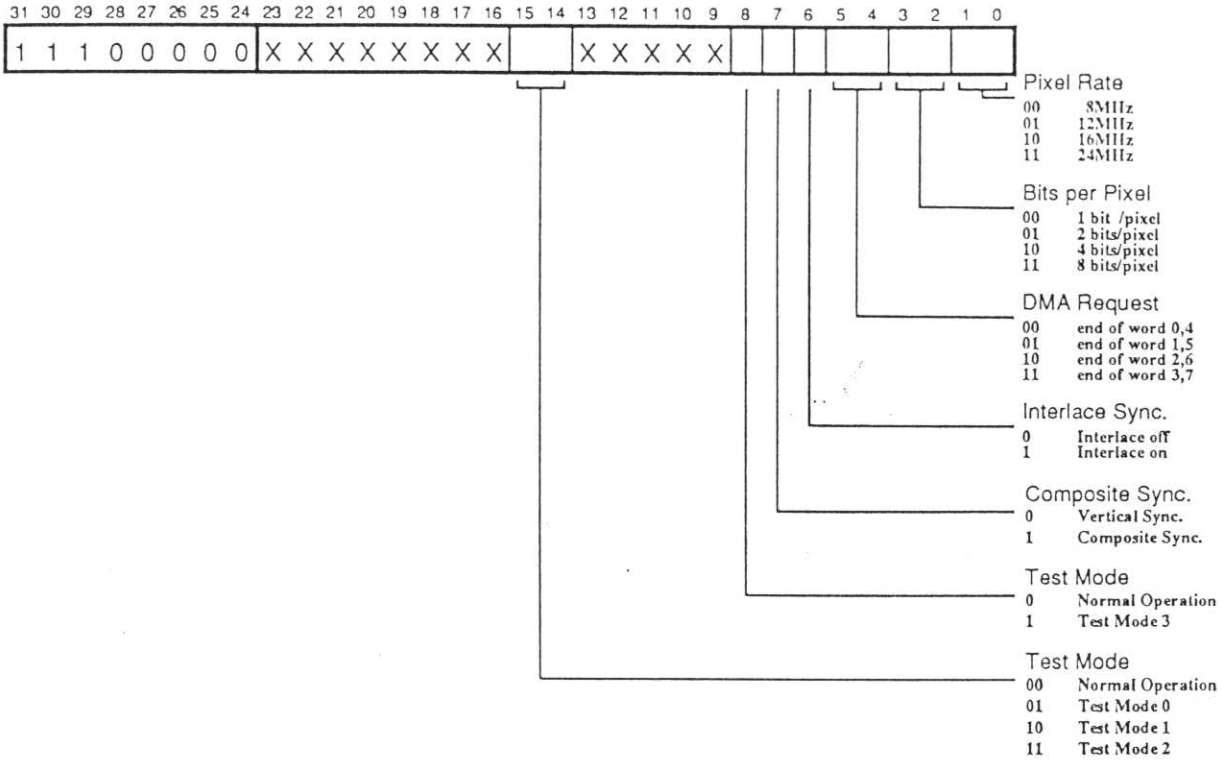


This register defines the byte sample rate of the sound data. It is defined in units of 1µs.

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

This is a 9 bit register with bit 0 the least significant. Bit 8 in the SFR is used as a test bit, and should always be set to 1. When this bit is set LOW, all the internal timing signals are cleared.

5.23 Control Register (CR) : Address E0H



This register defines the basic operating mode controls. A total of 11 bits are defined, and three of these are for test purposes only. Note that bit 8 in the **SFR** must also be set before the device can operate correctly.

The two bit-pairs for the pixel rate select and the bits per pixel select are defined in the diagram above. The bit-pair to define the point at which the DMA request flag is set is further explained in section 6.2.

To select interlaced sync. displays, **D[6]** in this register must be set as well as setting the correct values in the vertical and horizontal timing registers.

The $\overline{\text{V/CSYNC}}$ pin on the device can be programmed to output either the VSYNC pulse or the CSYNC pulse which is the exclusive-NOR of VSYNC and HSYNC. Selection is made by **D[7]**.

The remaining three bits are for testing the device out of circuit. Their action is as follows:

In test mode 0 (**D[14]** HIGH, **D[15]** LOW), the upper 5 bits of the horizontal counter are clocked by a derivative of the pixel clock.

In test mode 1 (**D[14]** LOW, **D[15]** HIGH), the lower 5 bits of the vertical counter are clocked by a derivative of the pixel clock.

In test mode 2 (**D[14]** HIGH, **D[15]** HIGH), the upper 5 bits of the vertical counter are clocked by a derivative of the pixel clock.

In test mode 3 (**D[8]** HIGH), the pin $\overline{\text{L/R}}$ outputs a signal which is 8 times the frequency of the sound byte sampling clock, and the pins $\overline{\text{SD[0:3]}}$ output the inverse of the data which is fed to the green DAC [**NIBSEL** LOW] or the blue DAC [**NIBSEL** HIGH].

Note that the device cannot function properly in test modes 0, 1 and 2, but test mode 3 has no effect on the normal operation .

6. Device Operation

6.1 The DMA Interface

The VIDC has 3 FIFOs into which DMA data is written. The Sound FIFO is four 32-bit words deep, and works independently from the other 2 FIFOs. The Video FIFO is eight 32-bit words deep, and the Cursor FIFO is again four 32-bit words deep.

6.1.1 Sound FIFO

Each word of data is strobed into the FIFO on the rising edge of $\overline{\text{SNDAK}}$. Data is read out of the FIFO into a bytewide latch which then drives the DAC. When the last byte in the FIFO is read into the latch, the signal $\overline{\text{SNDRQ}}$ is driven LOW, requesting another 16 bytes of data. The signal $\overline{\text{SNDRQ}}$ is driven HIGH when the first $\overline{\text{SNDAK}}$ is received.

The time available to service this data request is dependent on the sound data rate. The minimum value allowed in the *SFR* is 2, which defines a byte-rate of 3µs. Hence in this case the first word must be loaded into the FIFO less than 3µs after the $\overline{\text{SNDRQ}}$ signal is generated.

6.1.2 Cursor FIFO

The Cursor FIFO contains 16 bytes of data, which is enough for 2 rasters of cursor display. When the VIDC is programmed to display a cursor, $\overline{\text{VIDRQ}}$ is driven LOW at the same time as HSYNC goes LOW on the first raster on which the cursor is to appear. Data is loaded into the FIFO on the rising edge of $\overline{\text{VIDAK}}$. The FIFO must be filled completely (ie. 4 words) when the request is generated. The load cycle must be complete before the HSYNC pulse has ended.

$\overline{\text{VIDRQ}}$ is driven HIGH again when the first $\overline{\text{VIDAK}}$ is received. The cursor may be any number of rasters high, and the Cursor FIFO requests data during the HSYNC of every alternate raster on which it is displayed.

6.1.3 Video FIFO

The Video FIFO is eight 32-bit words deep, and it is arranged as a circular buffer. Data must always be loaded into it in blocks of 4 words, and this FIFO shares the same $\overline{\text{VIDRQ}}$ and $\overline{\text{VIDAK}}$ signals as the Cursor FIFO.

To accomodate the vastly different rates at which video data is required in the different modes, and to accomodate different DRAM speeds, the point at which more data is requested can be varied. This is controlled by bits 4 and 5 in the *Control Register* .

During the VSYNC pulse, the FIFO is cleared, and the signal $\overline{\text{VIDRQ}}$ is HIGH. After the HSYNC pulse of the first displayed raster, $\overline{\text{VIDRQ}}$ is driven LOW. Eight words must now be written into the FIFO by driving $\overline{\text{VIDAK}}$ LOW 8 times. This fills the FIFO. $\overline{\text{VIDRQ}}$ is set HIGH again when the fifth $\overline{\text{VIDAK}}$ is received.

Thereafter, the $\overline{\text{VIDRQ}}$ signal is set LOW whenever the FIFO empties to the point determined by bits 4 and 5 in the *Control Register* . The $\overline{\text{VIDRQ}}$ signal is normally set HIGH when the first $\overline{\text{VIDAK}}$ signal is received. However, if the data request is not serviced quickly, and the FIFO has emptied to the point where another 4 words have been read out when the first new data word arrives, then the $\overline{\text{VIDRQ}}$ signal will stay LOW, requesting another 4 words of data.

6.1.4 The Video DMA Interface

As noted above, the Cursor and Video FIFOs share the same DMA interface signals. Normally, a $\overline{\text{VIDRQ}}$ LOW during the HSYNC pulse is a request for Cursor data, and a $\overline{\text{VIDRQ}}$ LOW at any other time is a request for Video data. See Figure 1.

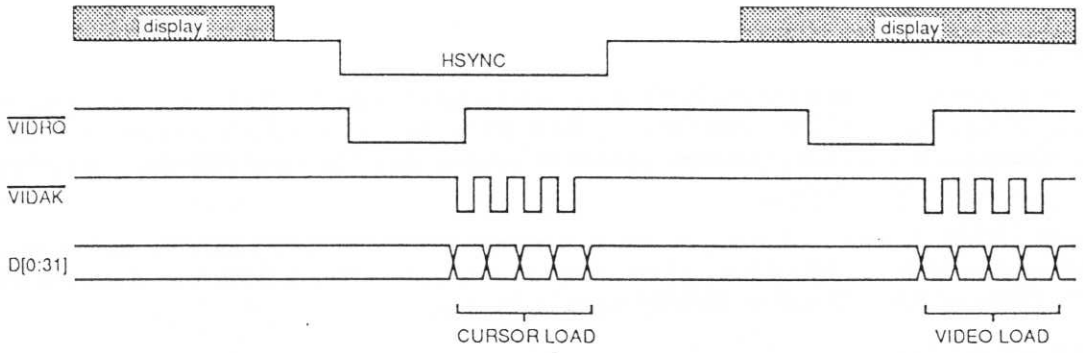


Figure 1: Video & Cursor DMA

However, often a video request happens just before the end of a raster display. (This is the data for the next raster). The load cycle for this video request is allowed to overlap the HSYNC pulse, even if a cursor request happens during the HSYNC pulse. Note that in this case the $\overline{\text{VIDRQ}}$ signal *may* not be driven HIGH between these two cycles. The first cycle must be video data and the second cycle must be cursor data. The cursor load cycle must still be complete before the end of the HSYNC pulse. This is shown in Figure 2.

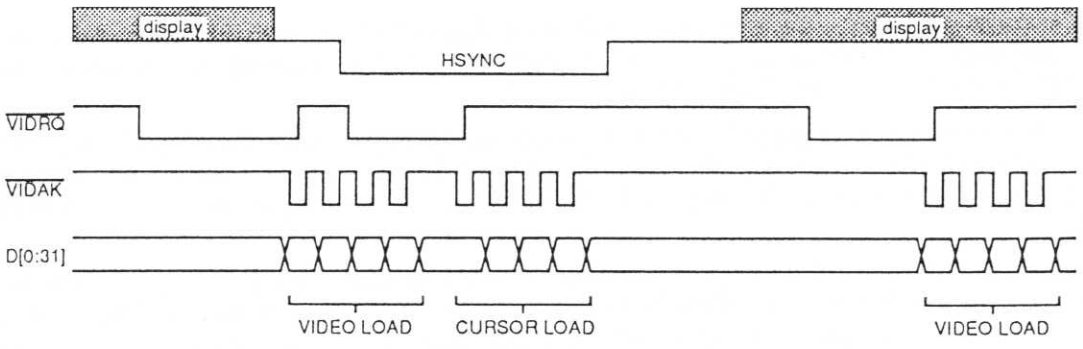


Figure 2: Video DMA overlapping HSYNC

Figure 3 shows the situation where a cursor is displayed on the first raster of the frame. Note the double video load cycle. The cursor load cycle must not overlap the end of the HSYNC pulse (otherwise data will be loaded into the wrong FIFO), and the first word of video data must be loaded into the FIFO before the display starts.

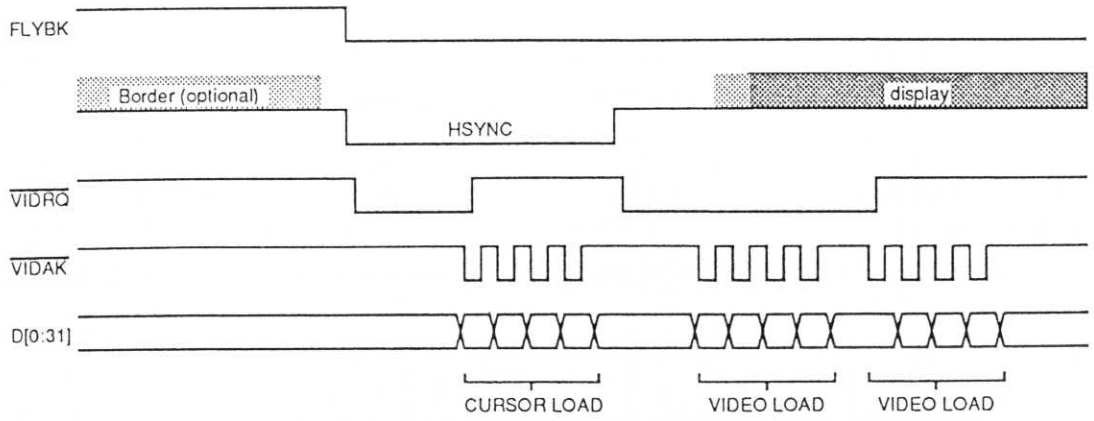


Figure 3: Cursor DMA at the Start of a Frame

6.2 Restrictions on Parameters

Certain restrictions must be applied to the screen parameters, most of which are system dependent. The following paragraphs assume the VIDC is being used in a system with the ARM and MEMC and 4 / 8MHz page mode DRAM. In this system DRAM cycles consist of an N-cycle (250ns) followed by up to 3 sequential S-cycles (125ns). Hence a VIDC FIFO 4-word load cycle consists of N + 3S which takes 625ns.

6.2.1 FIFO Request Pointer Values (Control Register D[4:5])

The Video FIFO is arranged as a circular buffer, though the core is asynchronous with a ripple-through time of 150ns from the top to the bottom. Data is loaded in blocks of 4 words, and is read out in bytes, starting with byte 0 of word 0. When the four bytes of word 0 are used, the pointer moves on to byte 0 of word 1 and so on. $\overline{\text{VIDRQ}}$ can be set LOW half way through reading the last byte of any of words 0-3 (and correspondingly 4-7) according to **D[4:5]** in the *Control Register*. Hence, in the high resolution video modes where the bytes are being consumed quickly, the request signal must be set at an earlier point than in the low resolution modes. The settings are defined in Table 4.

The request signal $\overline{\text{VIDRQ}}$ should be set LOW as soon as there *will be* enough room in the FIFO to accept the 4 words of data when they arrive. The minimum time between setting the request and receiving the last word of data is 187ns + 625ns = 812ns. [The 187ns figure is the minimum time in which MEMC can start a DMA cycle]. Now if the FIFO is full at the start, then it will have 4 words spare 150ns after the start of word 4. [150ns is the ripple time of a word through the FIFO]. Hence the request should be made at the first opportunity after (812ns -150ns = 652ns) before the start of word 4. The request can be made halfway through the last byte of any of words 0 through 3 by programming the *Control Register*.

Control Register D[5]	D[4]	$\overline{\text{VIDRQ}}$ Set at End of Words
0	0	0 , 4
0	1	1 , 5
1	0	2 , 6
1	1	3 , 7

Table 4: FIFO Request Pointer settings.

Depending on the video mode in use, data can be read from the FIFO at 1.5, 2, 3, 4, 6, 8, 12, or 16MBytes s^{-1}.

Figure 4 shows the case for the 16MBytes s^{-1} mode. The request must be set at the end of words 1 and 5.

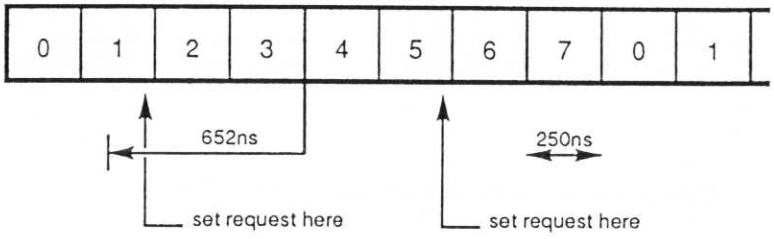


Figure 4: FIFO operating at 16MBytes s^{-1}.

Figure 5 shows the case for the 12MBytes s^{-1} mode. The request must be set at the end of words 2 and 6.

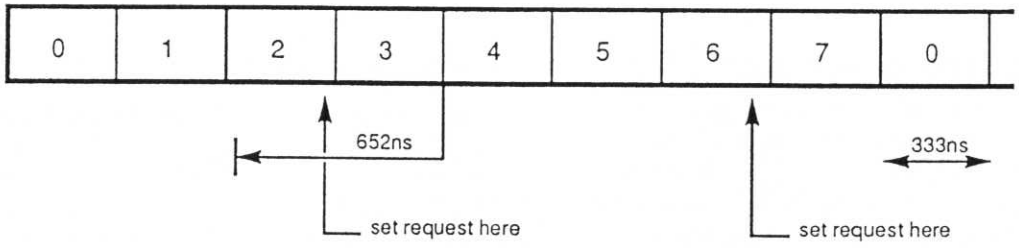


Figure 5: FIFO operating at 12MBytes s^{-1}.

Figure 6 shows the case for the 8MBytes s^{-1} mode. Again the request must be set at the end of words 2 and 6.

In all the other modes, the request should be set at the end of words 3 and 7.

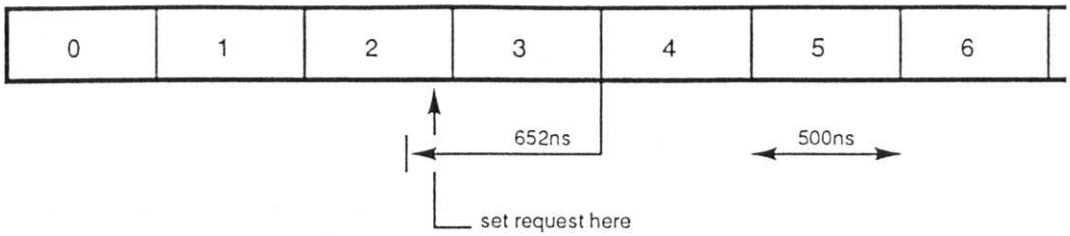


Figure 6: FIFO operating at 8MBytes s^{-1}.

6.2.2 Horizontal Sync Pulse Width

The HSYNC pulse width must be long enough to allow a complete load of the cursor FIFO. This is made up as follows:

2*[N+3S] (current + cursor cycles) + *syncmax* + 2**Tprop*.

ie. 2*625 + 312 + 100 = 1662ns.

syncmax is the maximum time MEMC can take to recognise the DMA request. *Tprop* is the time taken for the $\overline{\text{VIDRQ}}$ signal to reach MEMC, or the time taken for $\overline{\text{VIDAK}}$ to reach VIDC.

The pulse must also be long enough to allow the DMA Address Generator (DAG) in MEMC to reset the screen pointer. This may be as follows:

3*[N+3S] (current + cursor + sound cycles) + DAG reset.

ie. 3*625 + 250 = **2125ns**. This larger value must therefore be used.

6.2.3 Horizontal Front Porch Width

The front porch may be of zero length.

The total time from the end of display to the end of the HSYNC pulse must be more than 1912ns. As the HSYNC pulse width has to be at least 2125ns, this does not impose a restriction on the value of the front porch.

6.2.4 Horizontal Back Porch Width

The back porch must be long enough to allow the load of at least one word into the Video FIFO before the data is read out again. This is important at the start of the frame. Then the data is required in the bottom of the FIFO at least 4 pixel-times before the start of display, due to the pipelining in the VIDC. Hence the back porch must be greater than :

N+3S+N (current cycle + video N cycle) + syncmax + 2*Tprop + FIFO-ripple + 4 pixels.

ie. 250 + 375 + 250 + 312 + 100 + 150 + 4*83 = 1770ns for 12MHz displays.

or 250 + 375 + 250 + 312 + 100 + 150 + 4*125 = 1937ns for 8MHz displays.

6.2.5 Vertical Sync. Pulse and Porch Width

There are no restrictions on the values of the vertical front porch, back porch or sync. width. The Vertical Sync. Width Register (VSWR) may be programmed to value 0 which gives a VSYNC width of one raster.

6.2.6 Horizontal Display Width

If vertical scrolling is required, then the number of bits in the pixels of each raster must be a multiple of 128. If vertical scrolling is *not* required, then the number of bits in the pixels of each raster must be a multiple of 32.

6.2.7 Border

The border cannot be disabled. If no border is required, then it should be programmed to start and finish in exactly the same place as the display (both vertically and horizontally).

6.2.8 Cursor Position

The cursor should not be programmed to be outside the display area vertically, but it may be programmed to start or end outside the display area horizontally. The cursor must not be programmed to "run off" the right hand side of the screen, though it may be programmed to start before the left hand side. If a cursor of, say, only eight pixels wide is required, then the image should be programmed to be at the right hand end of the 32-pixel block, and the first 24 pixels should be programmed to be transparent. In this way, the displayed cursor may be positioned anywhere on the screen. Note that the cursor will not be displayed outside the border area either vertically or horizontally.

6.3 Display Formats

6.3.1 Screen Modes

14 of the possible 16 basic display modes are supported.

24MHz	8 bits/pixel	*Not Supported*
	4 bits/pixel	12MBytes s^{-1}
	2 bits/pixel	6MBytes s^{-1}
	1 bit /pixel	3MBytes s^{-1}
16MHz	8 bits/pixel	16MBytes s^{-1}
	4 bits/pixel	8MBytes s^{-1}
	2 bits/pixel	4MBytes s^{-1}
	1 bit /pixel	2MBytes s^{-1}
12MHz	8 bits/pixel	12MBytes s^{-1}
	4 bits/pixel	6MBytes s^{-1}
	2 bits/pixel	3MBytes s^{-1}
	1 bit /pixel	1.5MBytes s^{-1}
8MHz	8 bits/pixel	8MBytes s^{-1}
	4 bits/pixel	4MBytes s^{-1}
	2 bits/pixel	2MBytes s^{-1}
	1 bit /pixel	*Not Supported*

6.3.2 Data Display

Pixels are displayed starting at the top left hand corner of the screen, with the least significant bits of the first word in the FIFO.

In 8 bits/pixel mode, bits 0-7 of word 0 are the first displayed pixel.
In 4 bits/pixel mode, bits 0-3 of word 0 are the first displayed pixel.
In 2 bits/pixel mode, bits 0-1 of word 0 are the first displayed pixel.
In 1 bit /pixel mode, bit 0 of word 0 is the first displayed pixel.

6.3.3 Logical Data Fields

In 1 bit/pixel mode, the data field addresses the palette at location 0 or 1. The other 14 locations need not be programmed.
In 2 bits/pixel mode, the data field addresses the palette at locations 0 through 3. The other 12 locations need not be programmed.
In 4 bits/pixel mode, the data field addresses the palette at all 16 locations.
In 8 bits/pixel mode, the least significant 4 bits drive the palette as in 4 bits/pixel mode, and the most significant 4 bits drive the most significant bits of the RGB DACs directly.

6.3.4 Physical Data Fields

In 1,2,4 bits/pixel mode, the physical data field is :

SUP.	BLUE				GREEN				RED			
D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

In 8 bits/pixel mode, the physical data field is :

SUP.	BLUE				GREEN				RED			
D12	L7	D10	D9	D8	L6	L5	D5	D4	L4	D2	D1	D0

D_n: These bits come from the palette field.
L_n: These bits come from the logical field.

6.3.5 Cursor Format

The cursor is the same format in all video modes, and is automatically defined to be 32 pixels wide, though it may be any number of rasters high. Any pixel may be defined as being transparent, enabling cursors of any shape to be constructed within the 32 pixel horizontal limit. The format is always 2 bits per pixel, with bits 0,1 in the first word in the Cursor FIFO representing the first pixel, etc. The logical cursor pixel bit-pairs are defined in Table 5.

MSB	LSB	
0	0	transparent
0	1	cursor logical colour 1
1	0	cursor logical colour 2
1	1	cursor logical colour 3

Table 5: Cursor logical colours

The cursor physical field is exactly as the video physical field in 1,2,4 bits/pixel modes.

SUP.	BLUE				GREEN				RED			
D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

6.3.6 Border Field

The border physical field is exactly as the video physical field in 1,2,4 bits/pixel modes.

SUP.	BLUE				GREEN				RED			
D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

6.4 Controlling the Screen

6.4.1 Screen On / Off

The simplest method of turning the screen off is to program the Vertical Display End Register (*VDER*) to be less than the *VDSR*. This will not generate any video requests, but will display the border colour over the whole screen. The border can be turned off either by programming it to physical colour black, or by programming the *VBSR* to be greater than the *VBER*. Doing the latter will also disable the cursor, though cursor data requests will still be generated. Turning the screen back on should only be done during vertical flyback.

6.4.2 Cursor On / Off

The cursor should be turned off by setting the *VCER* to be less than the *VCSR*. [Value 0 is suggested]. This will also disable cursor data requests. Turning the cursor on, and moving it around should only be done during vertical flyback to prevent flash.

6.4.3 Writing to the Palettes & Other Registers

The palettes may be programmed reliably at any time, but are best programmed during vertical flyback. Changing the values of other registers should only be done during vertical flyback.

The signal **FLYBK** is set HIGH from the start of the first raster *after* the end of display (though it may still be border), until the start of the first raster which contains display data.

6.5 Video DAC Outputs

The Video DAC outputs are current sinks. Each DAC has a resolution of 4 bits, giving a linear transfer characteristic with 16 values. The magnitude of the output is a function of the video reference input current, with the maximum current sink being 15 times the reference input current.

In device VIDC1, a digital input value of 4 zeros gives the maximum current sink, and a digital input value of 4 ones gives zero current sink.

In device VIDC2, a digital input value of 4 zeros gives zero current sink, and a digital input value of 4 ones gives the maximum current sink.

The difference between the 2 devices results in a different output buffer circuit. A suitable circuit for VIDC1 is an emitter follower with appropriate level shifting. A suitable circuit for VIDC2 is shown in Figure 7. In this circuit, the diode, D1, should have similar characteristics to the base-emitter junction of TR1.

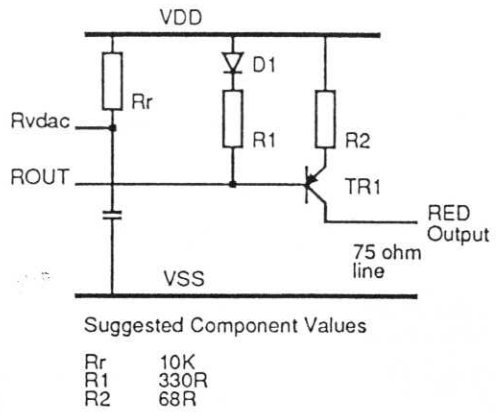


Figure 7: suggested video output circuit for VIDC2.

6.6 High Resolution Modes

The 4 bits of digital data which normally drive the red DAC are available to the user on pins **VED[0:3]** This pixel-rate bit-stream can be externally serialised to a single bit-stream of 4 times the VIDC pixel rate. With the VIDC operating at 24MHz, 4 bits/pixel mode, 96MHz bit-rates are generated giving very high resolution monochrome displays. Alternatively, with an external DAC, 48MHz grey-level displays are possible.

Refering to the Block Diagram, it will be noted that the data passes through the *High Res. Shifter* block before reaching the pins **VED[0:3].** This block is to enable the cursor to be positioned to any (96MHz) pixel. Note that this block also inverts the data from the red DAC.

When used in this mode, the VIDC must be programmed to a different set of values. But note that the "normal" analog modes of VIDC are still available simply by reprogramming; the addition of the shifter hardware will not affect the other modes, and the sound system is totally independent of this.

The following should be noted:

(1) 4 bits per pixel should always be selected.

(2) The programmed VIDC pixel rate is one quarter of the external pixel rate. The vertical timing parameters are unaffected by this as they are defined in units of a raster, but the horizontal timing parameters which are defined in units of 2 (24MHz) pixels can only be programmed in units of 8 (96MHz) pixels. There are now 4 times as many pixels on a line as are actually programmed. For example, if a display of 1024 * 1024 is required, the VIDC should be programmed to generate a display of 256 (horizontal) by 1024 (vertical).

(3) All 16 locations of the video palette should be programmed to be a 1:1 logical to physical mapping. Only **D[0:3]** (red DAC values) need to be programmed, as **D[4:11]** are ignored. The supremacy bit (**D[12]**) may be used if required.

(4) **D[4,5]** in the Border Colour Register must be set to zero. **D[0:3] & D[12]** may also be programmed if a border is required.

(5) The cursor palette should be programmed to the following values:
cursor colour 1 : 10H
cursor colour 2 : 20H
cursor colour 3 : 30H
Supremacy may also be used.

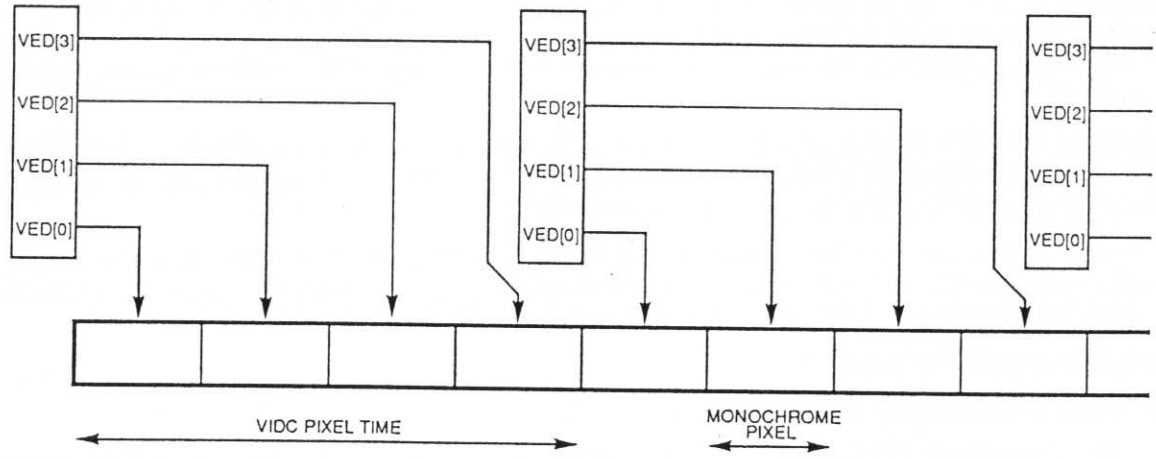
Then the 2 bits which define each cursor pixel are shown in Table 6.

MSB	LSB	
0	0	transparent
0	1	cursor black
1	0	*do not use*
1	1	cursor white

Table 6: Cursor logical colours, high res. mode

Note that the cursor can only be defined horizontally in units of 4 (96MHz) pixels, though it can be positioned anywhere on the screen to within one (96MHz) pixel. See section 5.12

The Hardware should be arranged so that $\overline{\text{VED[0]}}$ is the first bit to be serialised



6.7 External Synchronisation and Mixing

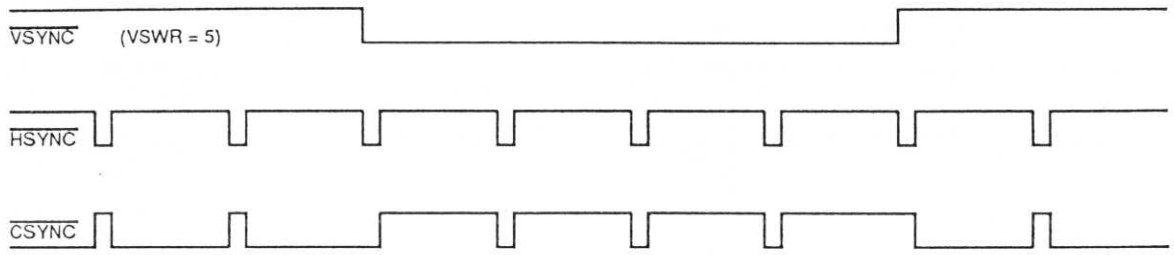
The VIDC has 2 signals associated with external synchronisation applications: $\overline{\text{SUP}}$ and SINK.

The signal $\overline{\text{SUP}}$ is an output which can be used to control an external multiplexer for mixing the VIDC output with that from an external source. All video and cursor logical colours from the palettes and the border colour can control $\overline{\text{SUP}}$. When D[12] in any of the above registers is set and that colour is being displayed, $\overline{\text{SUP}}$ is driven LOW. The output is pipelined and is synchronous with the DAC outputs and the $\overline{\text{VED[0:3]}}$ signals.

The signal **SINK** is an input which when driven HIGH will reset the vertical counters to the first raster. If an interlaced sync. display is being generated, then **SINK** will reset the counters to the first raster of the *odd* field. The pulse applied to this pin must be shorter than a raster time. The horizontal counters are not affected by this signal. The horizontal synchronisation must be done by phase-locking (or in simple applications, by interrupting) the input clock **CKIN**. Remember that the sound system is also driven from a derivative of **CKIN**.

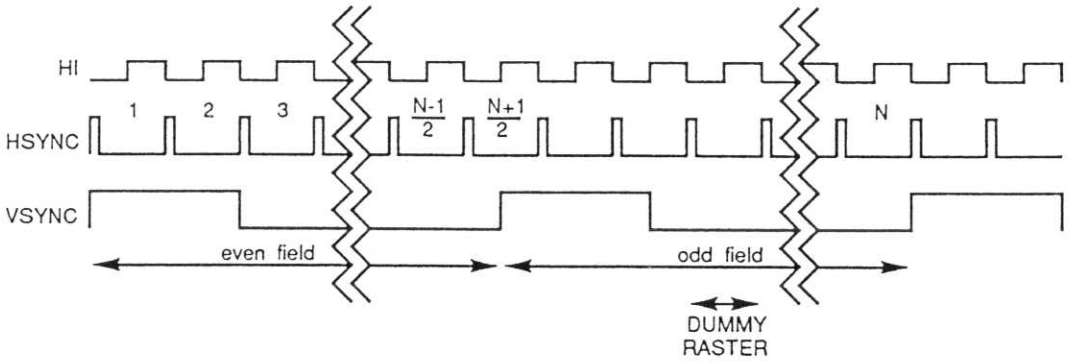
6.8 Composite Sync.

If D[7] in the *Control Register* is set HIGH, the $\overline{\text{V/CSYNC}}$ pin will output a composite sync. pulse. This is synthesised from the exclusive-NOR of VSYNC and HSYNC.



6.9 Interlaced displays.

The VIDC can be programmed to generate an interlaced sync. display. Normally the video data in each field is the same. The VIDC *Vertical Cycle Register* is set to a value (N-3)/2, where N is the total number of rasters in a frame. There are N/2 rasters in the *even* and *odd* fields. On raster (N+1)/2, the VSYNC pulse is output and the cycle repeats, but this is now the *odd* field, so the VSYNC pulse is delayed by half a raster time as defined by the value in the *IIIR*. On the first raster in the *odd* field after VSYNC, a dummy raster is inserted. This makes the *odd* field N/2 rasters long as well.



6.10 Sound System

The sound system consists of a four word FIFO and bytewide latch which drive a 7-bit exponential DAC. The eighth bit steers the DAC output to one of 2 pairs of output pins, one pair designated "+" and the other pair designated "-". The sound signal is generated externally by integrating and then subtracting these two pairs of signals. This is shown in Figure 8 below. Here the integration is performed by the capacitor C.

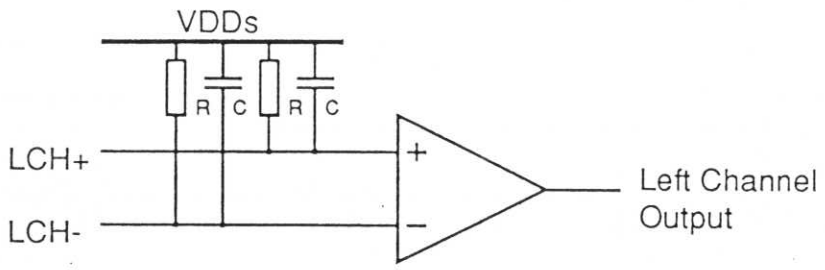
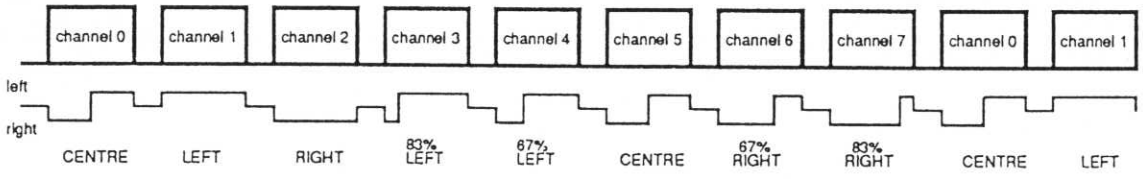


Figure 8: Combination of signals to produce left channel output.

Stereo image is synthesised by time-division multiplexing the output between the "left" and "right" pair of output signals. The first quarter of each sample is muted to allow for DAC settling and deglitching. The stereo image is specified for each channel by programming the corresponding Stereo Image Register.



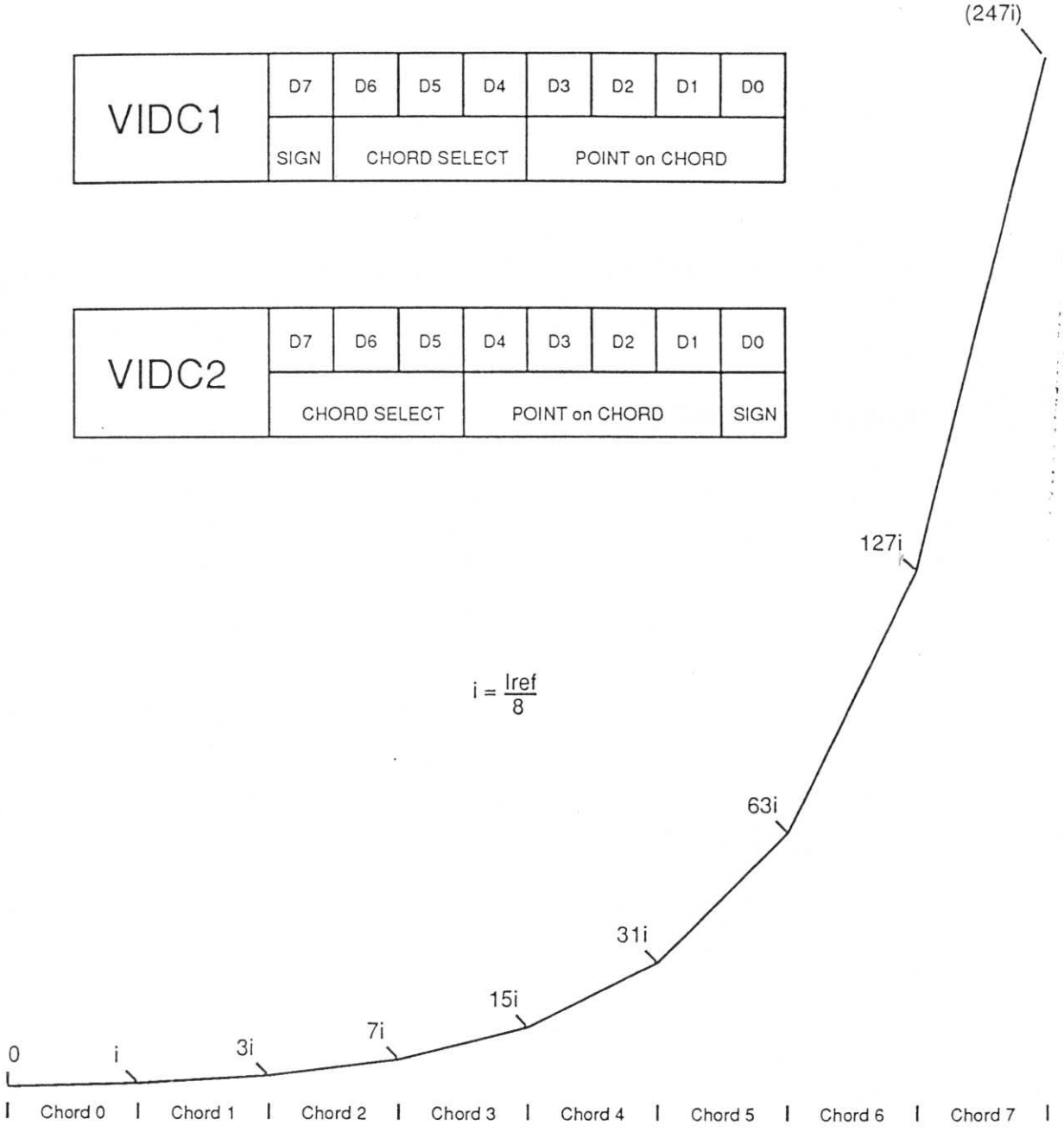
The system can operate in 1, 2, 4 or 8 channel modes. In 8 channel mode, the channels are sampled sequentially, starting with the first byte of data, which is channel 0; the second byte of data is channel 1 and so on. The external integrating time constant must be long enough to integrate over a complete sample cycle. In 4 channel mode, the fifth byte to be sampled is again channel 0, so Stereo Image Register 4 must be programmed to the same value as Stereo Image Register 0, and so on. In 2 channel mode, Stereo image registers 0,2,4 and 6 correspond to channel 0 and Stereo Image Registers 1,3,5 and 7 correspond to channel 1. In single channel mode, all eight Stereo Image Registers need to be programmed to the same value.

The sample rate is selectable by the *SFR* in units of 1µs, with a minimum value of 3µs. In eight channel mode the bytes for each channel are sampled with one-eighth of the frequency of single channel mode for a given value in the *SFR*.

The DAC transfer characteristic consists of 8 linear segments (*chords*). Each chord consists of 16 steps, and the step size in one chord is twice the step size in the preceding chord. This gives an approximation to the "μ255 law". Note that the order of the bits used to generate the sound values differs between VIDC1 and VIDC2. This is defined in the diagram below.

VIDC1	D7	D6	D5	D4	D3	D2	D1	D0
	SIGN	CHORD SELECT			POINT on CHORD			

VIDC2	D7	D6	D5	D4	D3	D2	D1	D0
	CHORD SELECT			POINT on CHORD				SIGN



The outputs are current sinks. The magnitude of the output is a function of the sound reference input current. The reference current is equal to the step size in the highest chord, which is 8i in the figure above.

7. DC Parameters

7.1 Absolute Maximum Ratings

Symbol	Parameter	Min	Typ	Max	Units
Vdd	supply voltage	-0.5		+7	V
Vip	voltage on any pin	-0.5		+7	V
Ts	storage temperature	-40		+125	deg C

7.2 DC Operating Conditions

Symbol	Parameter	Min	Typ	Max	Units	Note
Vdd	supply voltage	4.75	5	5.25	V	1
Vih	input logic "1"	2.4		Vdd	V	1
Vil	input logic "0"	0		0.8	V	1
Ivout	output current video DACs			-2	mA	
Isout	output current sound DAC			-2	mA	
Ta	ambient operating temperature	0		70	deg.C	

NOTES:

(1) Voltages measured with respect to VSS.

(2) IT - TTL compatible inputs.

KEY TO TABLES

Mes - Values measured in an ARM/MEMC/VIDC/IOC system running at 8MHz
Nom - Nominal values
Lim - Values required to meet ARM/MEMC/VIDC/IOC system specifications

7.3 DC Characteristics

measured at Vdd = +5.0V 25°C

PARAMETER	Mes	Nom	Lim	Units	Note
supply current	17			mA	1
output short circuit current	25			mA	2
input / output latch current	200			mA	3
output Hi voltage wrt Vdd		-150		mV	
ouptut Lo voltage wrt Vss		150		mV	
input Hi voltage threshold	2.0			V	
input Lo voltage threshold	1.7			V	
RVDAC, RSDAC voltage wrt Vdd		-1.3		V	4
voltage compliance, video DACs wrt Vdd at Iout = -2mA	1.8	1.7		V	
current compliance, video DACs at Vdd - 0.7V		4.5		mA	
voltage compliance, sound DAC wrt Vdd at Iout = -2mA		1.5		V	
current compliance, sound DAC at Vdd - 0.7V		3		mA	
input capacitance	5.0	5.0		pF	5

NOTES:

(1) Measured at a pixel rate of 24MHz. This value does not include any current output by the video DACs.

(2) Not more than one output should be shorted to either rail at any time, and for no longer than 1 second.

(3) This value represents the current that input/output pins can tolerate before the chip latches up. As sustained latch-up is catastrophic, this value must never be approached.

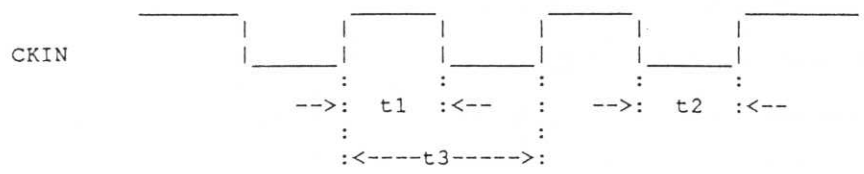
(4) This assumes a 10K resistor to VDD

(5) This value includes the capacitance of the chip carrier and socket.

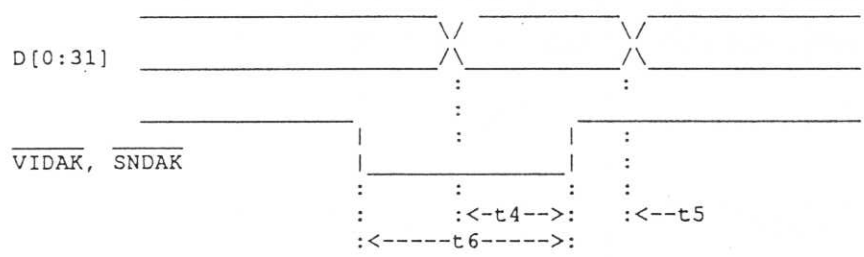
8. AC Parameters

8.1 AC Operating Conditions

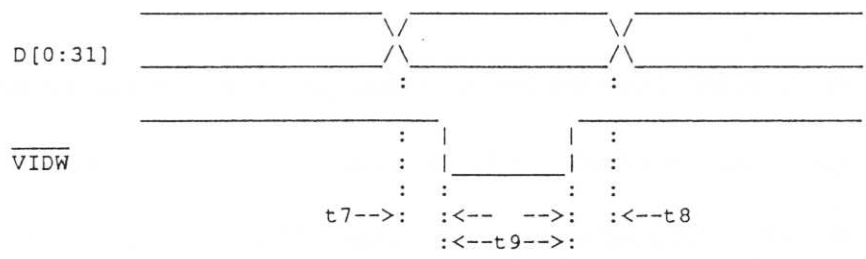
8.1.1 Input Clock



8.1.2 DMA Writes



8.1.3 Register Writes



Symbol	Parameter	Mes	Nom	Lim	Units	Note
t1	CKIN High	20	20	>6	ns	1
t2	CKIN Low	20	20	>8	ns	1
t3	CKIN frequency	24		24	MHz	1

Symbol	Parameter	Mes	Nom	Lim	Units	Note
t4	DATA - STROBE setup VIDAK, SNDAK	70	20	>5	ns	1
t5	DATA - STROBE hold VIDAK, SNDAK	30	15	>2	ns	1
t6	STROBE pulse width VIDAK, SNDAK	62	62	>15	ns	1

Symbol	Parameter	Mes	Nom	Lim	Units	Note
t7	DATA - STROBE setup VIDW	80	20	>5	ns	1,2
t8	DATA - STROBE hold VIDW	85	15	>5	ns	1
t9	STROBE pulse width VIDW	83	80	>15	ns	1

NOTE:

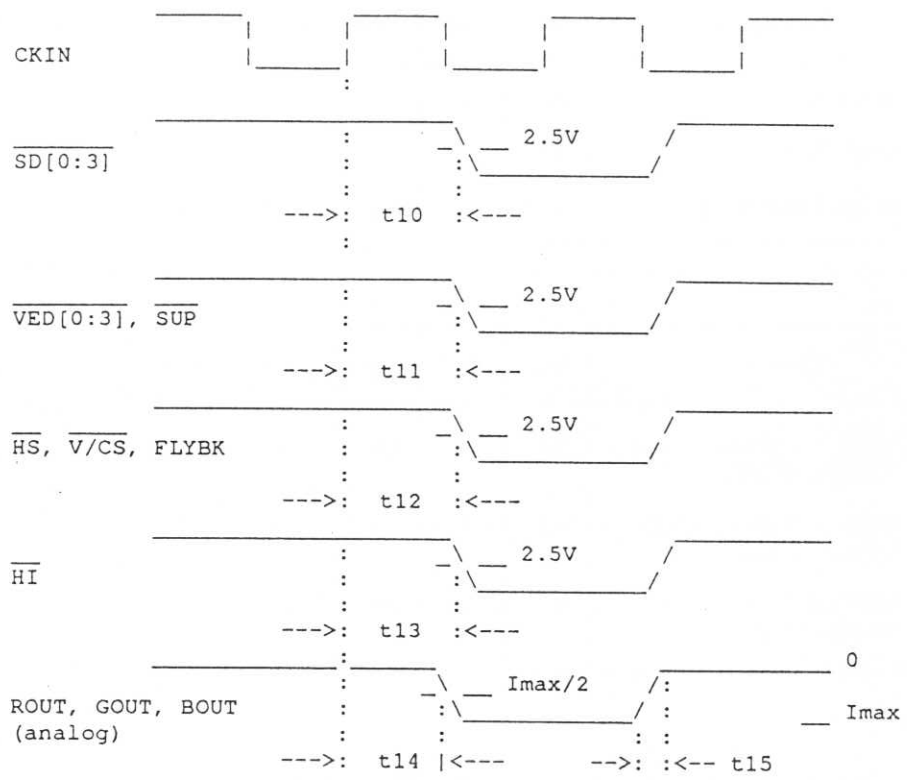
(1) The Limit values were measured for a sample VIDC. A factor of two should be allowed for process variations.

(2) As the data also carries the address, the data must be set up before $\overline{\text{VIDW}}$ goes LOW.

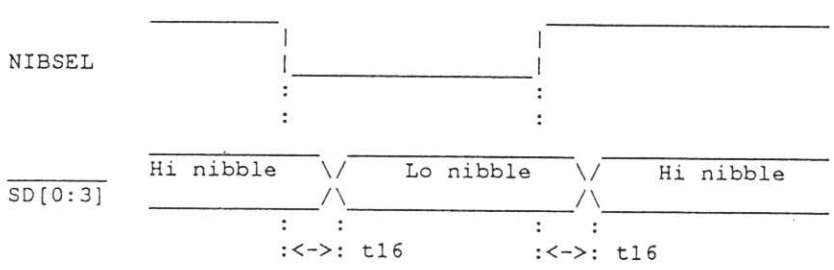
8.2 AC Characteristics

measured at Vdd = +5.0V 25°C

8.2.1 Clock - Outputs



8.2.2 NIBSEL - Output



Symbol	Parameter	Mes	Nom	Lim	Units	Note
t10	CKIN - $\overline{\text{SD[0:3]}}$	40			ns	1,2,3
t11	CKIN - $\overline{\text{VED[0:3]}}$, $\overline{\text{SUP}}$	33	42		ns	1,3
t12	CKIN - $\overline{\text{HS}}$,$\overline{\text{V/CS}}$, FLYBK	40			ns	3
t13	CKIN - $\overline{\text{HI}}$	26			ns	3
t14	CKIN - ROUT, GOUT, BOUT	30			ns	1
t15	Analog rise / fall	7	10		ns	4

NOTES:

(1) For pixel rates of 12MHz and 24MHz, the outputs are referenced to the rising edge of CKIN. For pixel rates of 8MHz and 16MHz, the outputs are alternately referenced to either edge of CKIN.

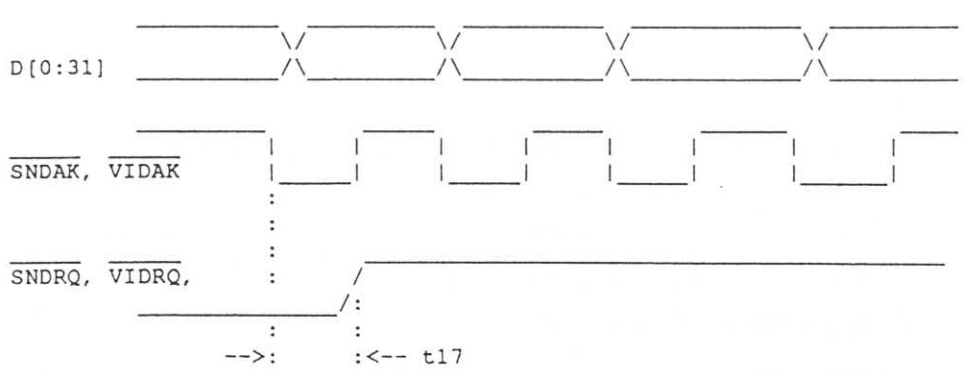
(2) The $\overline{\text{SD[0:3]}}$ signals are output one pixel time before the corresponding $\overline{\text{VED[0:3]}}$ due to pipelining differences.

(3) All digital outputs measured into 40pF load.

(4) Assumes a 5pF external load.

Symbol	Parameter	Mes	Nom	Lim	Units	Note
t16	NIBSEL - $\overline{\text{SD[0:3]}}$		10		ns	3

8.2.3 DMA Acknowledge - Request



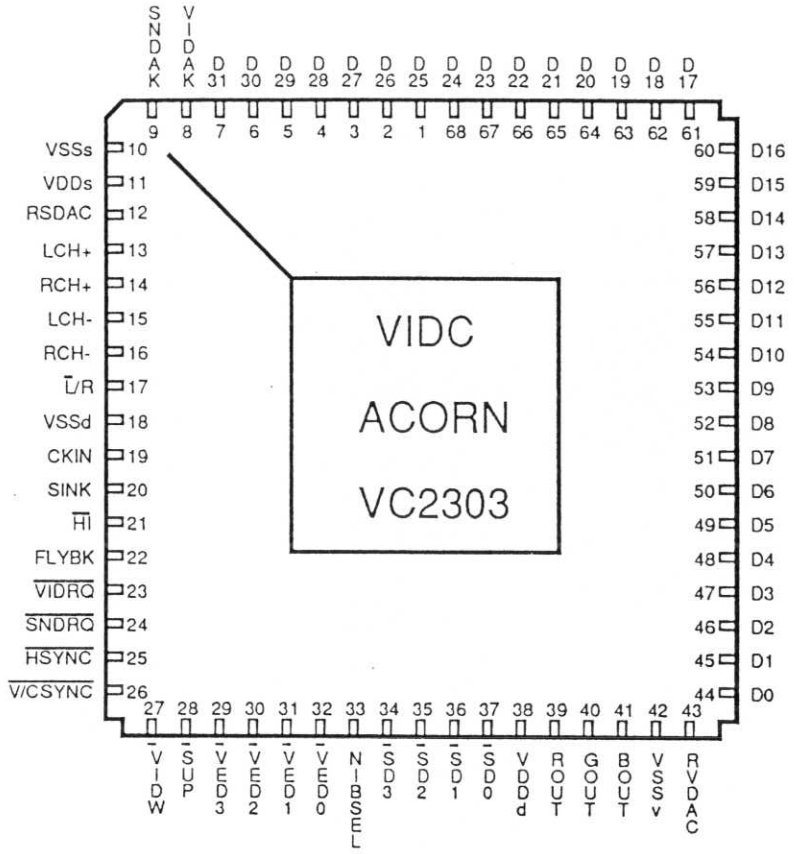
Symbol	Parameter	Mes	Nom	Lim	Units	Note
t17	$\overline{\text{SNDAK}}$ - $\overline{\text{SNDRQ}}$ $\overline{\text{VIDAK}}$ - $\overline{\text{VIDRQ}}$	25	40		ns	1

NOTE:

(1) $\overline{\text{VIDRQ}}$ or $\overline{\text{SNDRQ}}$ are cleared by the first $\overline{\text{VIDAK}}$ or $\overline{\text{SNDAK}}$ repectively, so long as no further request is pending.

9. Packaging

The device is packaged in a JEDEC B ceramic leadless chip carrier, or JEDEC C PLCC.



Suitable sockets are

(1) AMP 55159-1 for the ceramic device

(2) Burndy QILE68P-410T for the plastic device

These sockets both have a footprint as shown below.