VS23S010D-L Guide - 1 Megabit SPI SRAM
with Serial and Parallel Interfaces and Integrated Video Display Controller

Features

- Flexible 1.5V - 3.6V operating voltage
- 131,072 x 8-bit SRAM organization
- Serial Peripheral Interface (SPI) mode 0 compatible
  - Byte, Page and Sequential modes
  - Supports Single, Dual and Quad I/O read and write
  - Fast operation: the whole memory can be filled in 262158 or read in 262159 cycles (Quad-I/O SPI, Quad address mode)
  - XHOLD and XWP pins
- 8-bit Parallel Interface (Simplified 8080 and NAND FLASH Type Interface)
  - Sequential read and write in 4 byte blocks
  - Fast operation, the whole memory can be filled or read in 131077 cycles
- Integrated Video Display Controller with Video DAC
  - Supports NTSC and PAL video formats
  - Fully configurable by user
  - 9-bit Video DAC and 8x Video PLL
- High operating frequencies
  - Up to 38 MHz for SPI
  - Up to 40 MHz for 8-bit parallel interface
  - Over 35 MHz for Video Display Controller
  - (TBD) MHz for SRAM writes when Video Display Controller enabled
- Active Low-power
  - Read current 340 μA at 1 MHz (Single I/O, SO=0, T_A=+85°C, VDD=3.3V)
- Industrial temperature range
  - -40°C to + 85°C
- Pb-Free and RoHS compliant

Description

The VLSI Solution VS23S010D-L is an easy-to-use and versatile serial SRAM device. The memory is accessed via an SPI compatible serial bus. The device also contains Video Display Controller, which can be configured to continuously output analog composite video from the memory array data to implement a video frame buffer.

Alternatively, a 8-bit parallel interface can be used to access the SRAM instead of the SPI.

To sum up, there are four separate operating modes in VS23S010D-L:

- SPI Single, Dual, or Quad operation and 4 General Purpose I/O pins
- SPI Single, Dual, or Quad operation and simultaneous Video Display Controller
- 8-bit Parallel Interface operation
- 8-bit Parallel Interface operation and simultaneous Video Display Controller

Applications

- Micro-controller RAM extension
- VoIP and internet data stream buffer
- Audio data buffer
- Video frame buffer

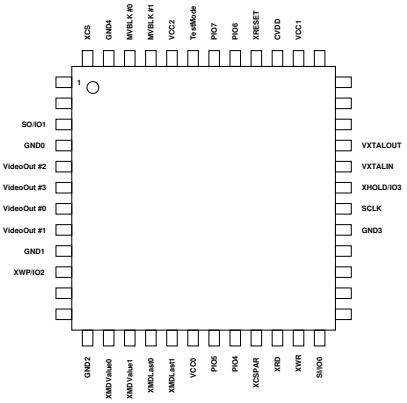


Figure 1: LQFP48 pin out (not to scale)

Operating Modes

VS23S010D-L operates in one of four modes: SPI, SPI and Video Display Controller, 8-bit parallel mode or 8-bit parallel mode and Video Display Controller.

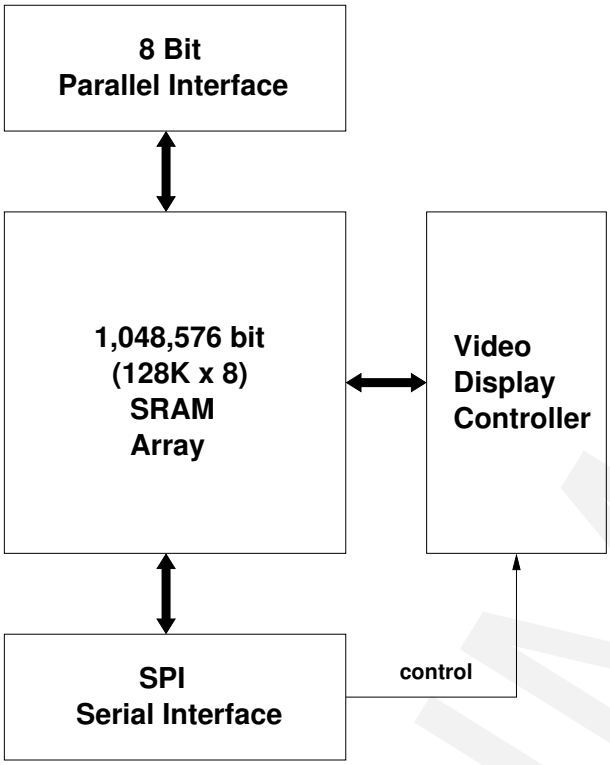


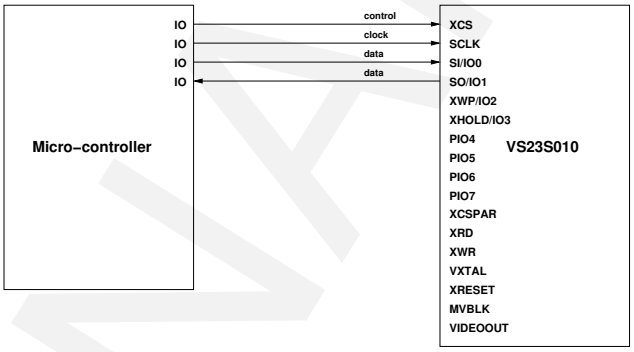
Figure 2: SPI or 8-bit parallel interface and Video Display Controller can be enabled at the same time.

In SPI mode SRAM and control registers can be accessed. Dual-I/O and Quad-I/O modes are used only for SRAM read and write.

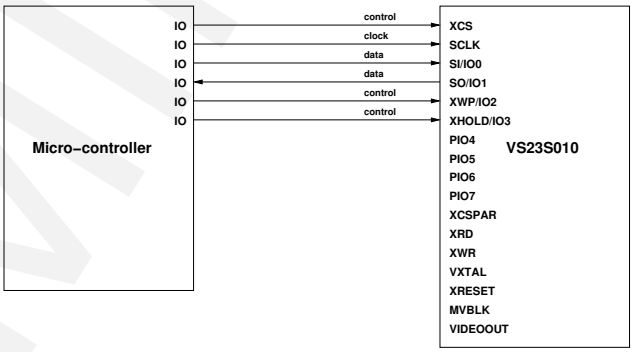
When Video Display Controller is enabled SPI can be used simultaneously. There is an additional limit to maximum SPI access rate in this mode.

When 8-bit parallel interface is used to access SRAM, SPI must be inactive. Video Display Controller can be operational simultaneously. However, Video Display Controller can be controlled only by SPI. There is a limit to maximum access rate for 8-bit parallel interface when Video Display Controller is enabled.

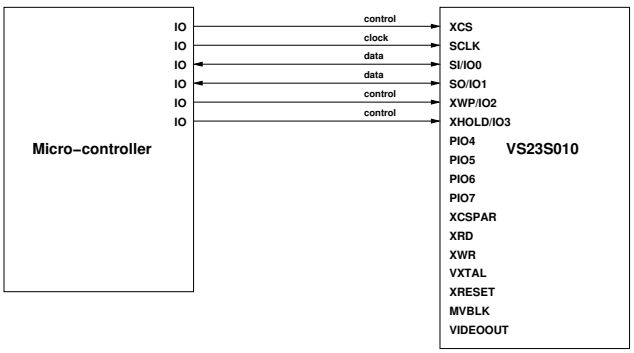
Following are connection examples for different operating modes. Some I/Os of VS23S010D-L are unconnected, because they have internal pull-up or pull-down resistors. Note also, that power and ground connections are not shown in the following examples.



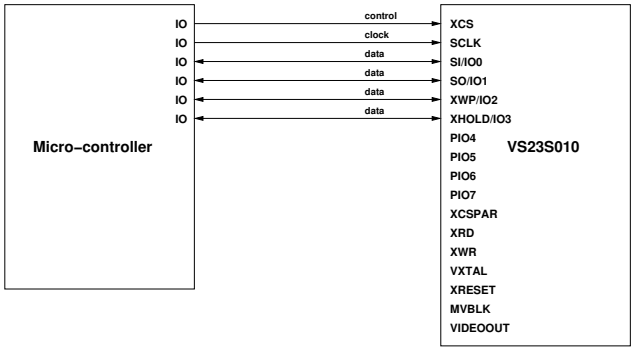
SPI connection, minimum configuration



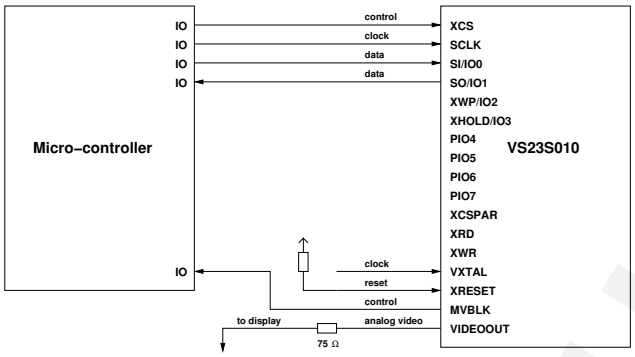
SPI connection, basic configuration



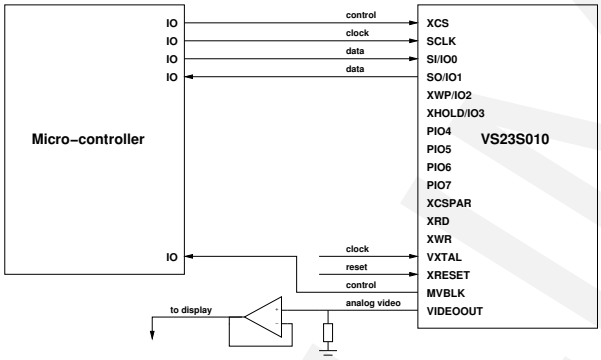
SPI Dual−I/O connection



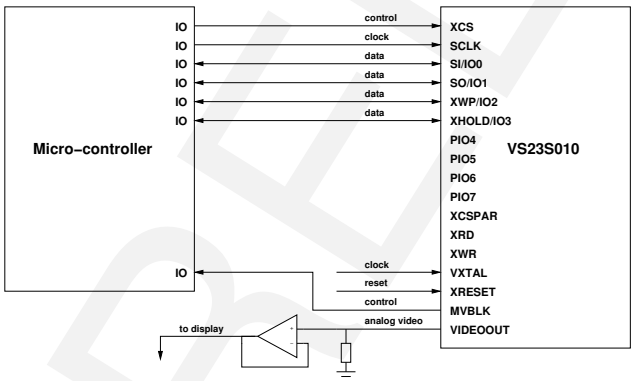
SPI Quad−I/O connection



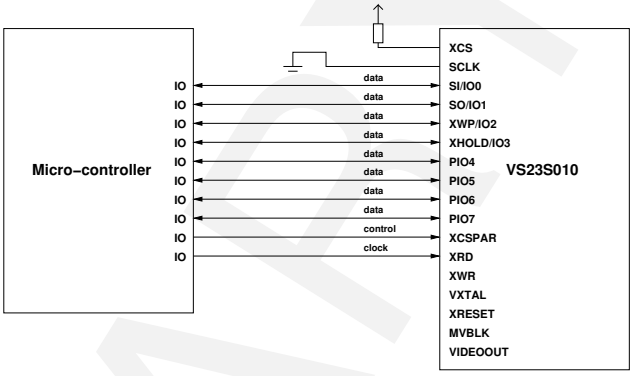
SPI connection (minimum configuration), video generator enabled



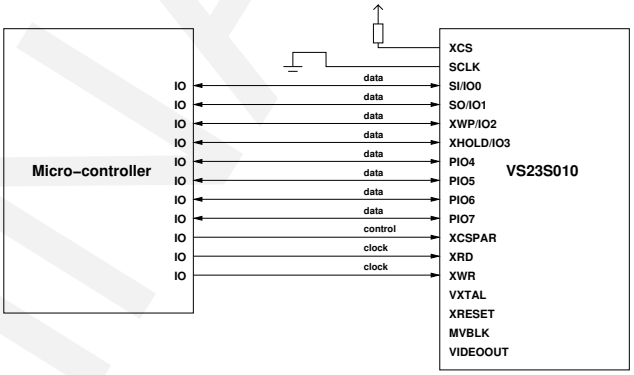
SPI connection (minimum configuration), video generator enabled, video buffer



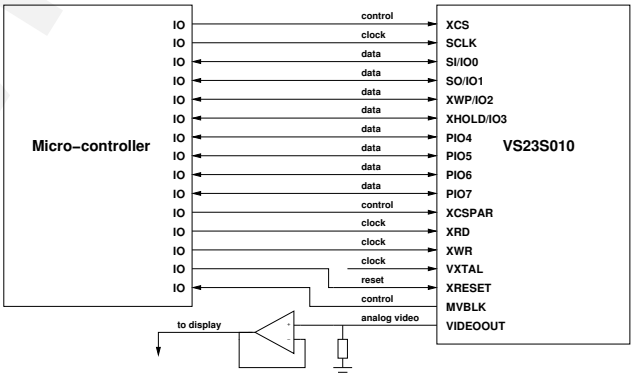
SPI Quad−I/O connection, video generator enabled, video buffer



8−bit parallel interface (minimum configuration, one clock is enough)



8−bit parallel interface



8−bit parallel interface, video generator enabled, video buffer
(SPI for video generator control when 8−bit interface is not active)

Contents

List of Figures

1 Disclaimer

This is a *preliminary* guide. All properties and figures are subject to change.

2 Definitions

B Byte, 8 bits

b Bit

CSClk Clock, which frequency is Color Subcarrier Frequency of a video format.

GPIO General Purpose I/O

LSB Least Significant Bit

MSB Most Significant Bit

NTSC National Television System Committee video format, color subcarrier frequency is 3.579545 MHz.

PAL Phase Alternating Line video format, color subcarrier frequency is 4.43361875 MHz.

POR Power On Reset

SPI Serial Peripheral Interface

SRAM Static Random Access Memory

TBD To Be Defined

U, V Chrominance components (color information)of video signal

VClk Video Display Controller clock of the VS23S010D-L. It can come directly from VXTAL oscillator or can be generated on-chip by 8x PLL from VXTAL pins. **VClk frequency has to be 8 times the color subcarrier frequency of the selected analog video format.**

$F_{VClk} = 8 \times F_{CSClk}$

If on-chip PLL is used, the VXTAL clock frequency is 3.579545 MHz for NTSC and 4.43361875 MHz for PAL video. If Video Display Controller clock is directly from the VXTAL pins without using the on-chip PLL, then VXTAL clock frequency is 28.63636 MHz for PAL and 35.46895 for NTSC video.

Y Luna component (the brightness) of video signal

3 Connection Guidelines

To minimize power supply noise connect suitable by-pass capacitors between VCC supply pins and GND. Place by-pass capacitors as near as possible to VS23S010D-L for best effect.

VXTALIN and VXTALOUT are crystal oscillator pins for Video Display Controller.

Make sure that there is the lowest possible capacitive coupling between different clocks and chip selects (SCLK, XRD, XWR, PGCLKIN, PGCLKOUT, XCS and XCSPAR) and particularly to data signals on the circuit board. This is for minimizing interference between these signals.

VideoOut can be connected to a display via 75 Ω series resistor or by using an op-amp buffer.

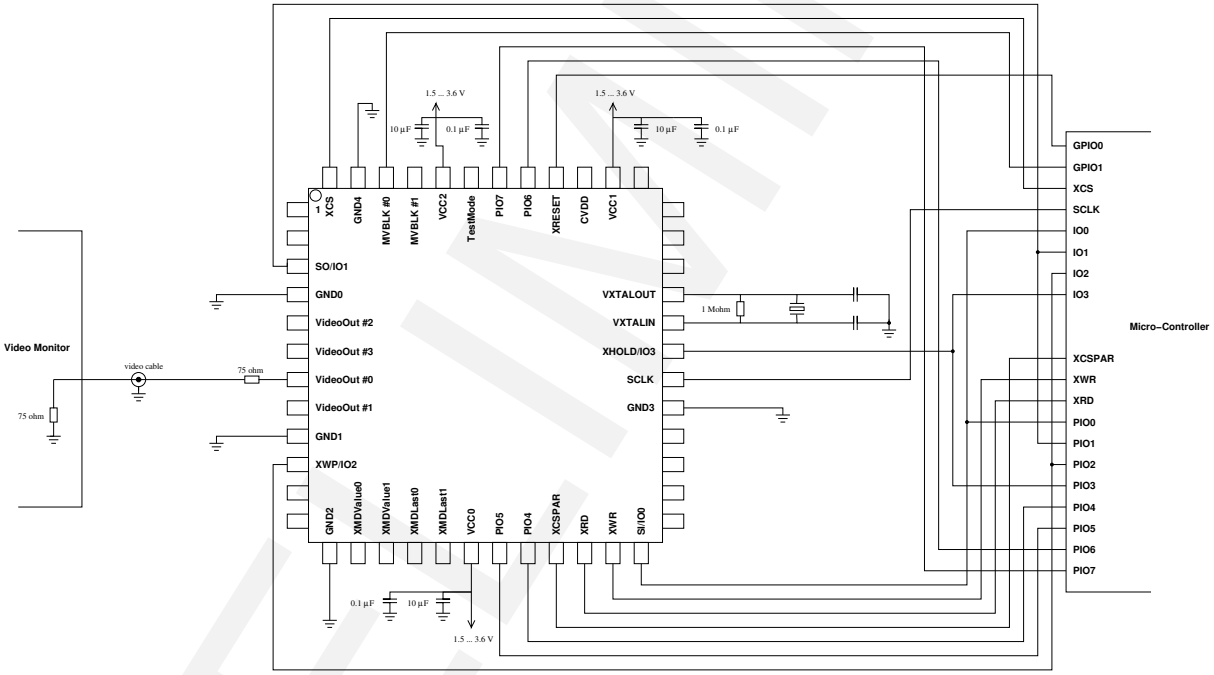


Figure 3: Connection example

4 Device Operation

The device consists of following main blocks: SPI, Video Display Controller, 8-bit Parallel Interface and SRAM. SPI and Video Display Controller can be enabled simultaneously and also 8-bit Parallel Interface and Video Display Controller can be enabled at the same time. However, SPI and 8-bit Parallel Interface have to be used separately because they share I/O. The SRAM can be written and read by all other blocks of VS23S010D-L.

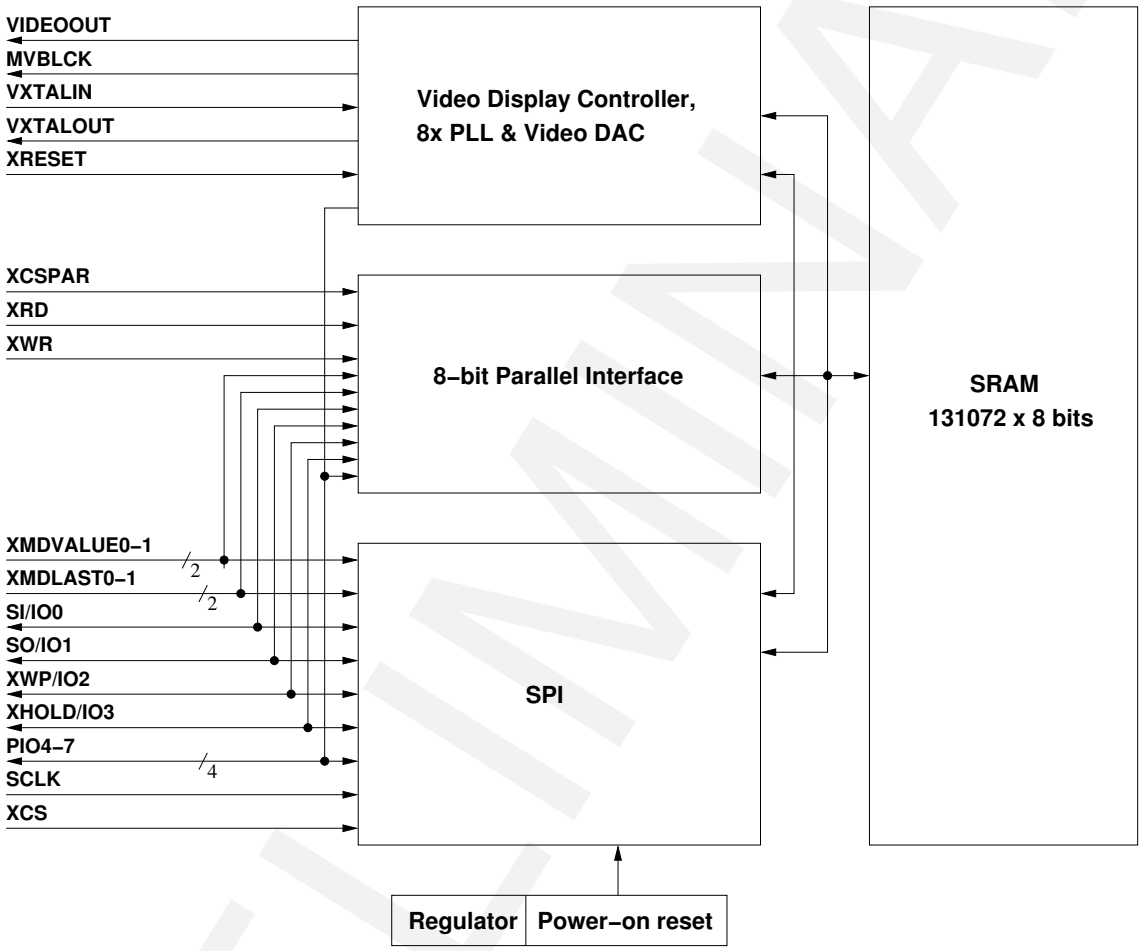


Figure 4: Device Organization

In this guide is explained Video Display Controller usage considerations in detail. The operation of data interfaces is described in the datasheet of VS23S010D-L.

5 Video Display Controller Operation

The data in SRAM can be converted to analog composite video by Video Display Controller block. The Video Display Controller is fully configurable by user. Refer to datasheet of VS23S010D-L for details of required SPI commands.

5.1 Block General Description

Video Display Controller is very versatile analog video generation device. The contents of the SRAM can be converted to analog composite video output using several SPI commands. Video clock crystal oscillator (pins VXTALIN and VXTALOUT) is used for generating the clock (VClk) for Video Display Controller. Video clock crystal oscillator output can be used as VClk or its frequency can be multiplied by 8 in the 8x PLL. The frequency of the VClk is eight times the color subcarrier frequency (CSClk frequency) of desired video format, for PAL 4.433618 MHz and for NTSC 3.579545 MHz. Following table summarizes general properties of the Video Display Controller block.

Summary of Video Display Controller	
Versatile organization of SRAM	
Configurable SRAM block move	
9-bit video DAC	
8×PLL for VXTAL	user selectable (VXTAL or PLL)
Microcode programmable	4 byte program, 2 to 16 VClk cycles for a pixel
Supported formats	Composite video (PAL, NTSC etc.) and direct DAC mode
Video SRAM capacity	1048576 bits
Color subcarrier frequency for PAL	4.433618 MHz
Color subcarrier frequency for NTSC	3.579545 MHz
Pixels per line	up to 2048 (theoretical)
Lines per picture	up to 1023 (theoretical)
Line types	Proto (fixed code) and Normal (programmable)
Y width	1 to 8 bits, unsigned
U width	0 to 6 bits, signed
V width	0 to 6 bits, signed
U presets	four 4 bit values
V presets	four 4 bit values
Digital output	4-bit, programmable for video synchronization

SPI or 8-bit parallel interface can be operated when Video Display Controller is enabled. The initialization and enabling of the Video Display Controller are made by SPI so during that period 8-bit parallel interface can't be used. Theoretical maximum operating frequencies of SPI or 8-bit parallel interface when Video Display Controller is enabled are shown on the following table. The Status register StFastWV bit can be used to accelerate SPI write operations when Video Display Controller is on. When the bit is enabled, the module-4 of the first write address has to be zero. Also the modulo-4 of the amount of bytes has to equal zero in this mode. The given data rates are theoretical maximum values and in reality they are more of guidelines.

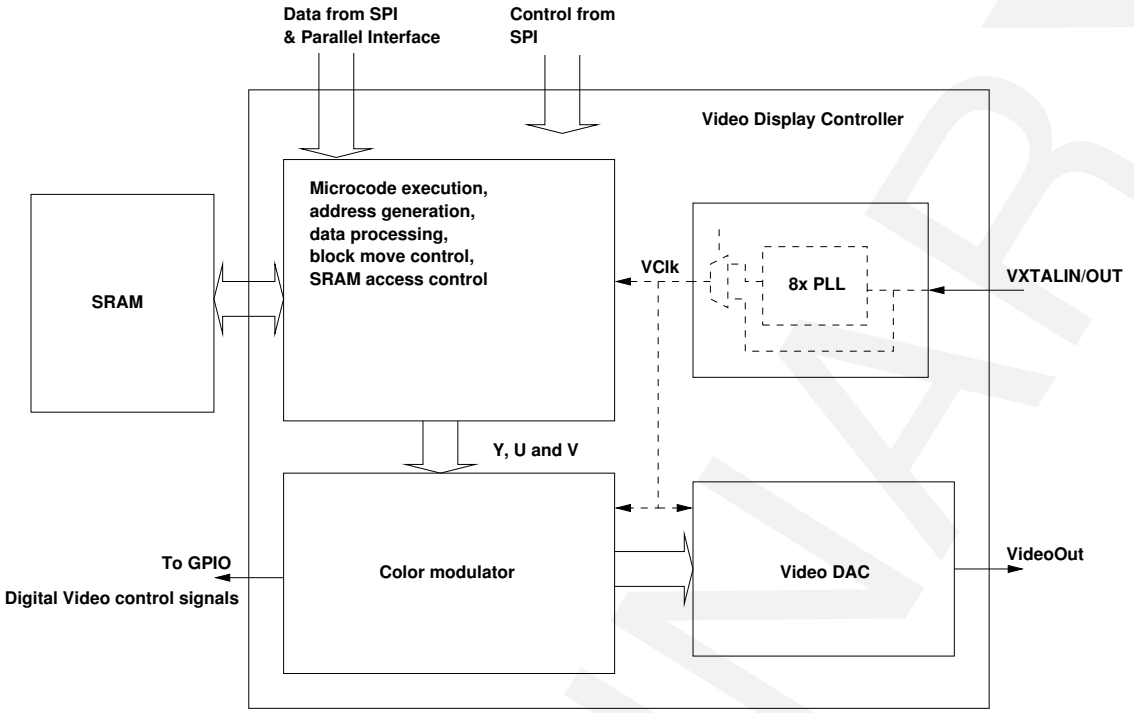


Figure 5: Video Display Controller block diagram

Max. interface speeds when Video Display Controller is enabled (theoretical)				
Mode	Read/Write	StFastWV bit	Max. interface clk freq. ($\times F_{VClk}$)	VClk cycles/Byte
SPI Single	R	don't care	1	8
SPI Single	W	"0"	1	8
SPI Single	W	"1"	15	$8/15$
SPI Dual	R	don't care	1	4
SPI Dual	W	"0"	1	4
SPI Dual	W	"1"	7	$4/7$
SPI Quad	R	don't care	1	2
SPI Quad	W	"0"	1	2
SPI Quad	W	"1"	3	$2/3$
SPI register op.	don't care	don't care	SPI max. speed	does not affect
8-b Parallel	R	don't care	2	$1/2$
8-b Parallel	W	don't care	2	$1/2$

The example of possible picture resolutions are shown in the following table. There is shown maximum amount of colors for each resolution.

Resolution	H	V	Pixels	Colors [1]	Bits per pixel	Memory bytes
NTSC YUV422 [2]	352	240	84480	65536	8+4	126720
MCGA	320	200	64000	65536	16	128000
CDG	300	216	64800	65536	16	129600
QVGA	320	240	76800	8192	13	124800
NTSC VCD	352	240	84480	4096	12	126720
PAL VCD	352	288	101376	1024	10	126720
NTSC non-interlaced	440	243	106920	512	9	120285
PAL non-interlaced	520	288	149760	128	7	131040
HVGA	480	320	153600	64	6	115200
EGA	640	350	224000	16	4	112000
VGA letterbox	640	400	256000	16	4	128000
NTSC Analog	440	486	213840	16	4	106920
NTSC SVCD	480	480	230400	16	4	115200
NTSC DVD	720	480	345600	8	3	129600
VGA	640	480	307200	8	3	115200
PAL Analog	520	576	299520	8	3	112320
PAL SVCD	480	576	276480	8	3	103680
PAL DVD	720	576	414720	4	2	103680

[1] Theoretical number of colors based on aligned memory consumption (integer bits per pixel). Actual performance can vary due to implementation details.
[2] YUV422, 8 bits luminance per each pixel plus 8 bits chrominance for each pixel pair.

5.2 Parameters of Video Display Controller

There are several adjustable parameters in the video picture. Figure 6 shows the main parameters of a video frame:

1. Line length is defined in VClk cycles. This means that increasing the length by 8 increases the duration of line by one CSClk (color subcarrier) cycle. Line length is a 12 bit value ranging from 1 to 4096. Each line begins with a fixed black level (i.e. zero) signal lasting 10 VClk cycles which is the same as 1.25 CSClk (color subcarrier) cycles. So the line total length can vary from 11 to 4106 VClk cycles. The line length in PAL video system is 283.75 color clock cycles. The register value for this is $283.75 \times 8 - 10 = 2260$, which is 8D4h . The line length in NTSC video system is 227.5 color clock cycles. The register value for this is $227.5 \times 8 - 10 = 1810$, which is 712h.

   Line length is set by Write Line Length command.

2. Line count is the amount of lines per video frame. It is a 10-bit value ranging from 1 to 1023. When the last line is output the system starts again from the first line.

   Line count is set by Write Video Display Controller Control1 command.

3. Picture start is given in CSClk (color subcarrier) cycles (i.e. 8 times VClk cycles). It defines the CSClk cycle where and after video data is fetched from the defined normal

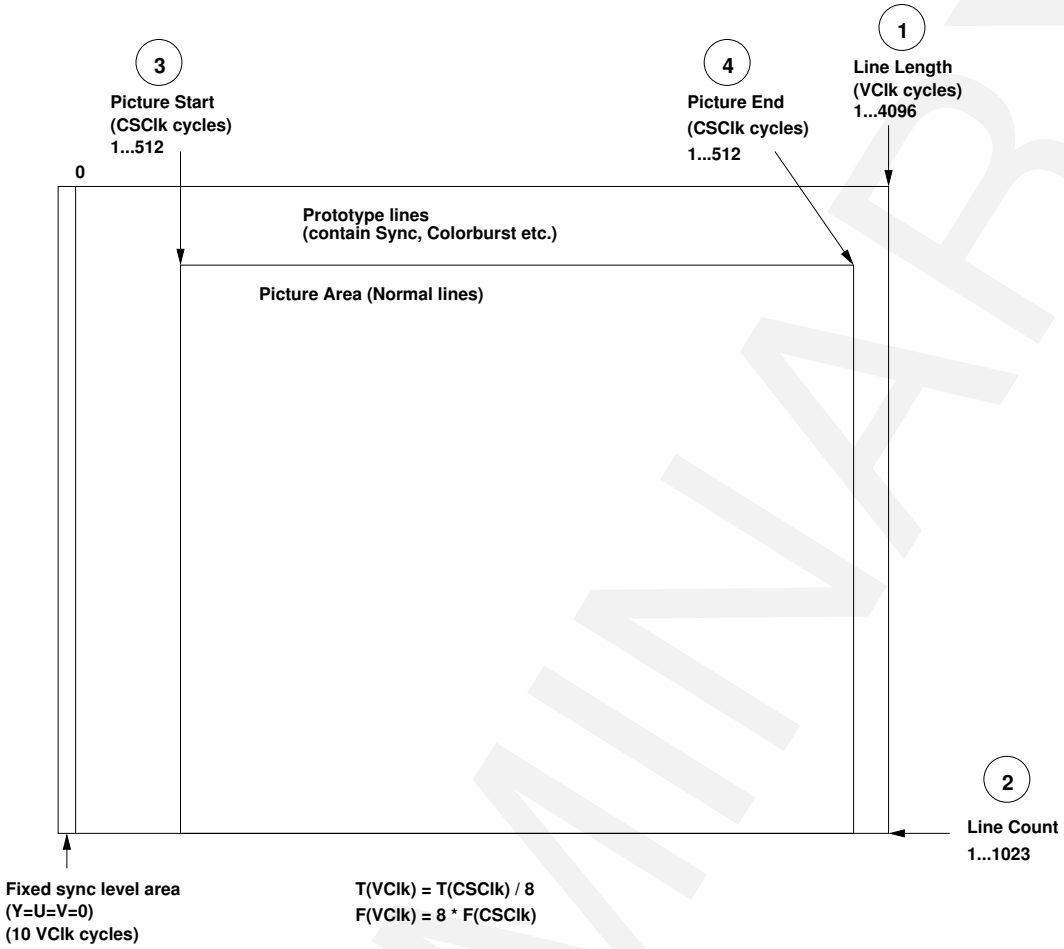


Figure 6: Video picture parameters

line SRAM area. Video data before Picture start cycle is fetched from a defined prototype line area. Prototype and normal lines can have different video formats. Picture start has a 10 bit value and it ranges from 1 to 512.

Picture start is set by Write Picture Start value command.

4. Picture end is given in CSClk cycles. It defines the CSClk cycle where and after video data is fetched again from the defined prototype line SRAM area. Video data starting from Picture start cycle to Picture end cycle minus one is fetched from a defined normal line area. Prototype and normal lines can have different video formats. Picture end has a 10 bit value and it ranges from 1 to 512.

   Picture end is set by Write Picture End value command.

Microcode program is used for controlling the video generation. The program consists of four bytes. Each program run can last from 2 to 15 VClk cycles. One code line is executed on each VClk cycle. If the run is less than 4 cycles, then only the N first lines of code are executed. If the run is more than 4 cycles, then the rest of the cycles are idle. The program syntax is as follows:

```
                pick            bits            shift
cycle           a|b|y|-         1...8           0...6

0               a               4               4       // take V(4), shift 4
1               b               4               4       // take U(4), shift 4
2               y               8               4       // take Y(8), shift 4
3               -               x               4       // idle, shift 4
```

Each code line can have one of the four functions:

- Pick a, which takes the amount of bits from the SRAM data and sets them as V data.
- Pick b, which takes the amount of bits from the SRAM data and sets them as U data.
- Pick y, which takes the amount of bits from the SRAM data and sets them as Y data.
- Pick -, which does not take any data. However, this command can be used to shift the SRAM data additionally. Because the maximum SRAM data shift value is 6 and it is possible to take 8 bits for Y, an extra SRAM data shift cycle is needed to keep the SRAM data in synchronization.

As mentioned above, the bits select, how many bits of SRAM data is used for each operation. U and V data can be from 1 to 6 bits, Y data can be from 1 to 8 bits. Shifts are done according to program to keep the SRAM data synchronized. The tables below show how U and V data and Y data are organized depending on bit depth before sending to Color Modulator. Y value is an unsigned integer and U and V are signed integers.

Bits	Proto/Normal	U & V Data Bit Organization					
		5(MSB)	4	3	2	1	0
1	Normal	0	0	“0”	“0”	“0”	“0”
2	Normal	1	1	0	“0”	“0”	“0”
3	Normal	2	2	1	0	“0”	“0”
4	Proto	3	2	1	0	“0”	“0”
4	Normal	3	3	2	1	0	“0”
5	Normal	4	4	3	2	1	0
6	Normal	5	4	3	2	1	0

The Y value to color modulator is filled with MSB when less than 8 bits are used. The shorter Y data is aligned to MSB part and the lower bits are filled with MSB. The purpose of this to get maximum amount of separate luminance levels with each data width of Y. The very slight negative effect is that in the middle of luminance range there is one step that is twice the normal amount.

Bits	Proto/Normal	Y Data Bit Organization							
		7(MSB)	6	5	4	3	2	1	0
1	Normal	0	0	0	0	0	0	0	0
2	Normal	1	0	1	1	1	1	1	1
3	Normal	2	1	0	2	2	2	2	2
4	Normal	3	2	1	0	3	3	3	3
5	Normal	4	3	2	1	0	4	4	4
6	Normal	5	4	3	2	1	0	5	5
7	Normal	6	5	4	3	2	1	0	6
8	Proto/Normal	7	6	5	4	3	2	1	0

A protoline is a line of fixed UVY type (4 bits U, 4 bits V and 8 bits Y) and therefore it has a hardwired program. The microcode for the protoline is the example on previous page. The program length for protoline is eight VClk cycles.

There are still some other parameters affecting the video picture:

- Program length, this tells after how many VClk cycles the Video Display Controller microcode program is run again. The range is from 2 to 16.
- Index Start parameter is used to define the address where line indexes start in the SRAM.
- Select PAL mode, this control enables the V phase alteration on odd lines in the Color Modulator for the PAL system.
- Translate U and V (TRUV bit), this mode enables the use of four element tables for U and V values.
- UV Skip control, this can be used to skip the lines of microcode that pick U and/or V values. The value tells in how many code runs the U and V commands are not executed. The range is from 0 to 7.
- Y filter enable is for enabling the low-pass Y filter.
- PLL controls are needed to enable the 8x PLL and to select it as a clock source
- DAC control is for selecting the small or large current mode of Video DAC.
- Digital Output Control is used to select PIO outputs as digital video control outputs. This is useful for example for generating video synchronization signals. In protoline area V data can be selected as digital output by setting U data to minimum value (8h).

There are three readable parameters considering the Video Display Controller:

- Current Line value tells the line number at which the Video Display Controller is generating the video. The value is updated before SPI starts to output the data via SO. The range is from 0 to 1023.

  The Current Line value is read using the read Current Line Value & PLL Lock command.

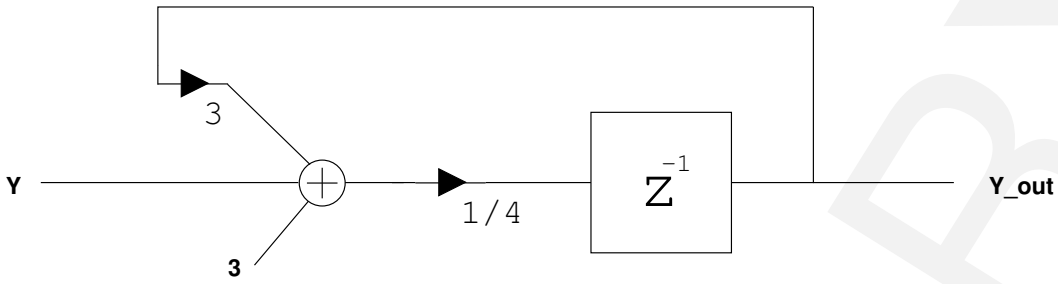


Figure 7: Switchable low-pass Y filter

- PLL Lock bit signals if the 8x PLL is locked to incoming VXTAL frequency and that its output frequency is correct.

  The PLL Lock is read using the read Current Line Value & PLL Lock command or using the read GPIO State register command.

- Block Move Active bit is high when Video Display Controller block move is active.

  The Block Move Active is read using the read GPIO State register command.

5.3 Memory Organization

In Video Display Controller mode SRAM array is divided into a couple of special sections. All the SRAM accesses are done by the Video Display Controller automatically according to user selectable register parameters.

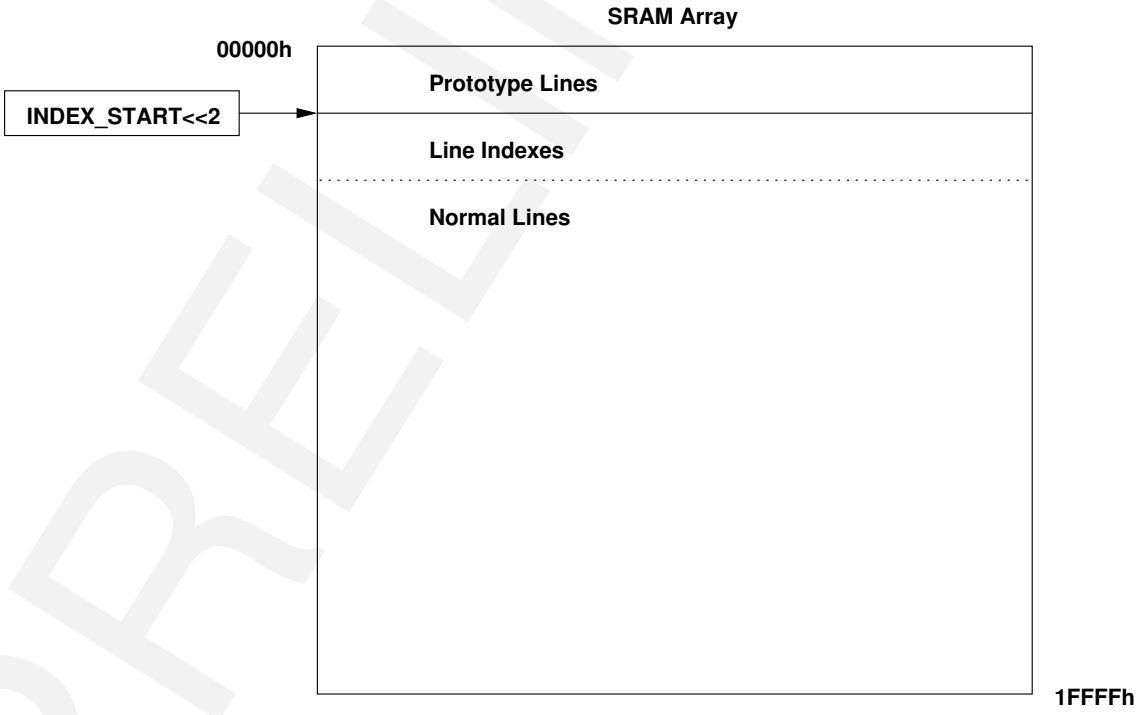


Figure 8: Video mode SRAM organization

INDEX_START value gives the byte address from where to fetch the index address for the first line. The eventual SRAM byte address, where the first index address is fetched, is INDEX_START shift left by two. Index address is fetched from the SRAM at the beginning of each line. Index address tells from which address the picture data for that line starts. If the index address is smaller than INDEX_START then line will be a prototype line, otherwise it is a normal picture line. If line is a normal line, then the beginning and the end of line are from the prototype line as defined by Line End and Line Start registers. There can be several protolines for different needs. The beginning address of the protoline is generated using the proto offset. The start byte address of the protoline is proto offset shift left by 9.

SRAM address	SRAM data byte	
	7 4	3 0
INDEX_START << 2	Line0 Index 3:0	Line0 Proto offset
+1	Line0 Index 11:4	
+2	Line0 Index 19:12	
+3	Line1 Index 3:0	Line1 Proto offset
+4	Line1 Index 11:4	
+5	Line1 Index 19:12	
+6	Line2 Index 3:0	Line2 Proto offset
+7	Line2 Index 11:4	
+8	Line2 Index 19:12	

Figure 9: Index address organization

Protoline has a fixed form and microcode. First is picked 4 bits V, then 4 bits U and last is taken 8 bit Y. If V value of protoline is set to minimum (8h) then it is not used as a new V signal. Setting V to minimum passes U value to 4-bit digital output instead of setting it as a new U. 4-bit digital control output is directed to PIO outputs by setting VGP bit of Line Length register high. The direction of the PIO pins is set by GPIO Control Register also in this mode. So PIO pins that are used in this mode have to be set as outputs separately. Note, that 8-bit parallel interface overrides the VGP bit selection, if XCSPAR pin is set to low for some parallel operation. The following table summarizes the modes of the PIO7-4 pins.

PIO7-4 Function Priority		
Priority	Mode	Control
$1st$	8-bit parallel mode	XCSPAR pin low
$2nd$	Digital video control output	VGP bit high & GPIO Control Register
$3rd$	GPIO pins	GPIO Control Register

On normal line the organization of U, V and Y data depends on the microcode program. For example, for a 8 bit pixel, there could be 2 bits for U, 2 bits for V and finally 4 bits for Y like is shown in Figure 12. There are not much limitations to data organization, for example, if there is a program where is taken first 3 V bits, then 4 U bits and finally 7 Y bits, then the organization

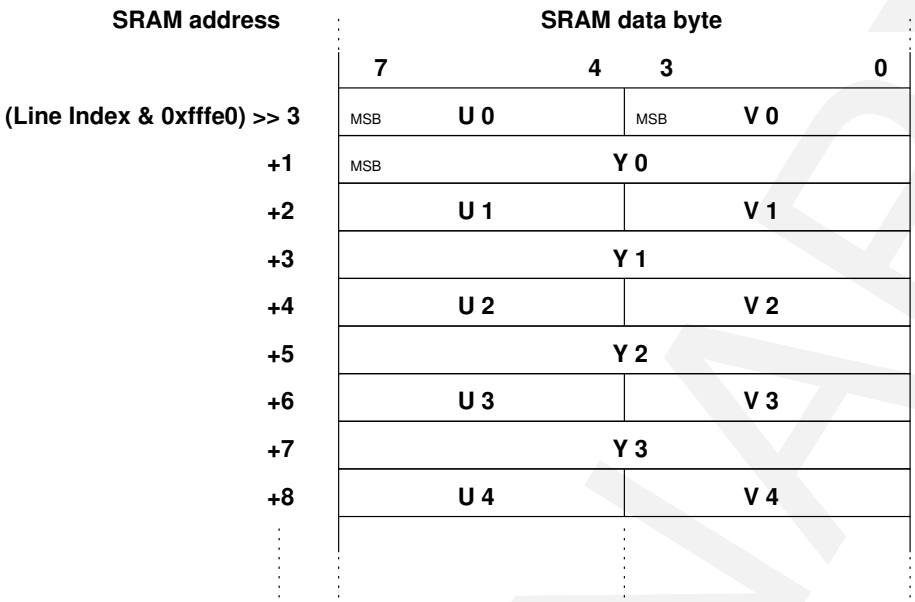


Figure 10: Protoline data organization, when whole line is protoline

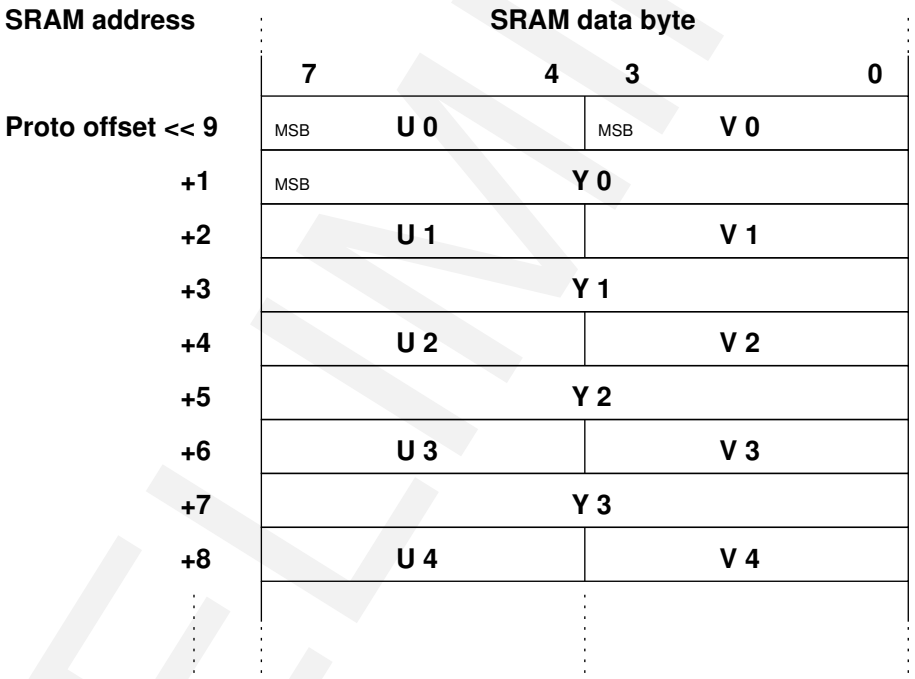


Figure 11: Protoline data organization for a picture line (Note that the starting address is formed differently than in previous picture)

is as shown in the Figure 13. The index address is a bit address, so the byte address of the pixel data is index address shift right by 3. Additionally the bit position of the MSB of the first video data is given by the three LSBs of the index address, the start position is 7-value of three LSBs. The organization of the data in SRAM bytes is optimized for generating the data using a barrel shifter of a master micro-controller.

On normal line it is possible to pass U and V data picking from SRAM by setting UVSkip to a non-zero value. In Figure 14 is shown an example of a program, where U and V are two bits

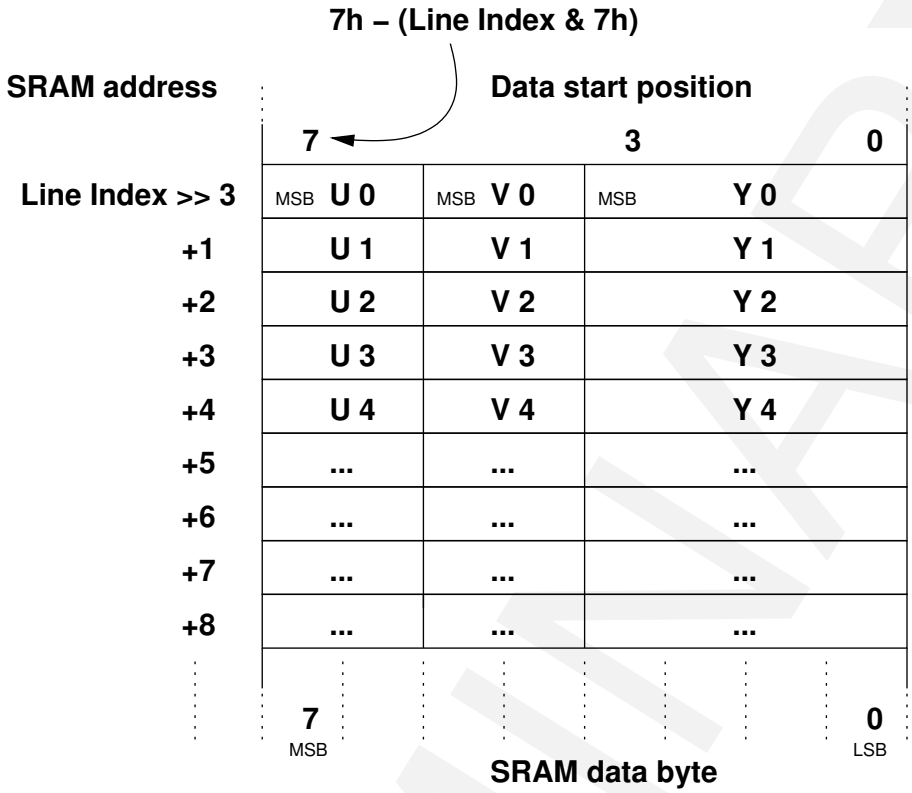


Figure 12: Normal line data organization example, 2 bits U, then 2 bits V and 4 bits Y for a pixel

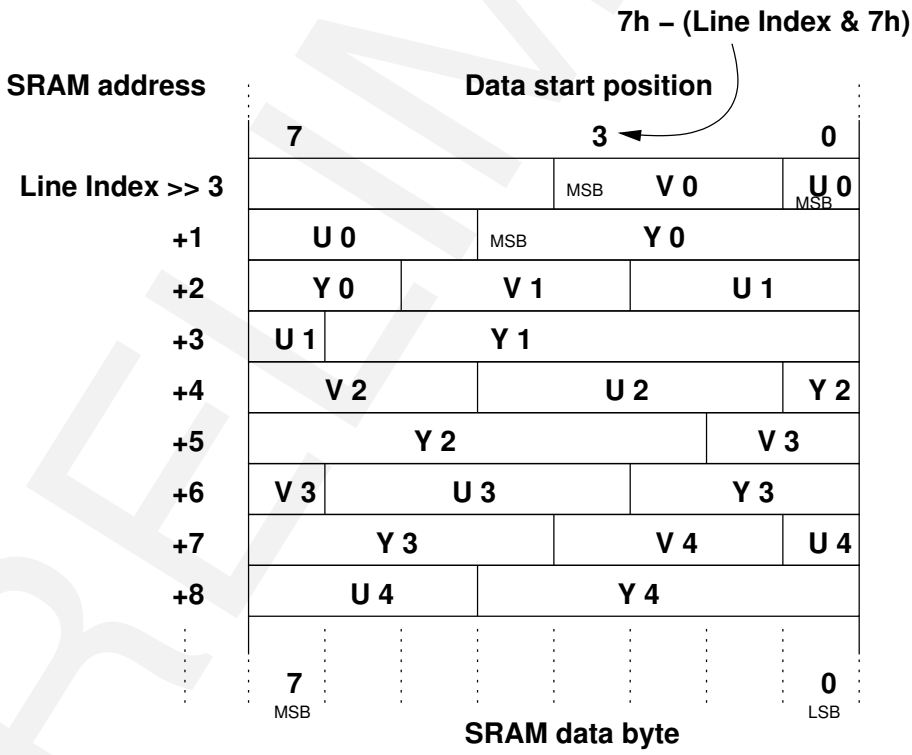


Figure 13: A more untypical, but possible normal line data organization example, 3 bits V, 4 bits U and 7 bits Y for a pixel.

long and Y is 6 bits. UVSkip is set to 4.

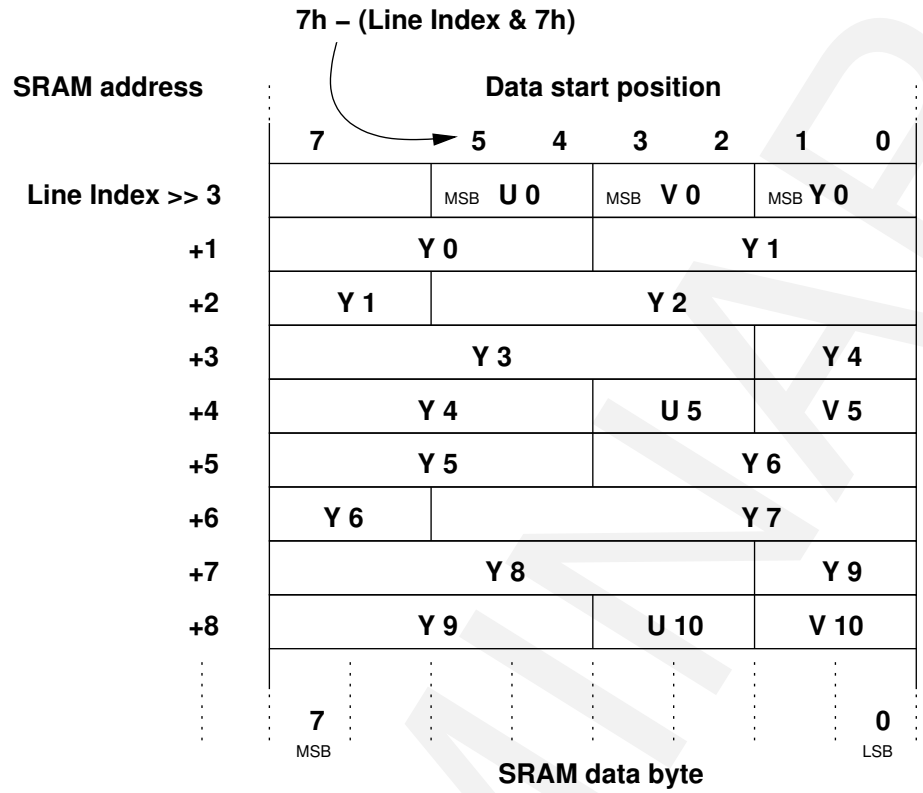


Figure 14: Normal line data organization example, 2 bits U and V, 6 bits Y and UVSkip value 4

5.4 8x PLL and Clock Switch

VClk, clock for the Video Display Controller is generated by the 8x PLL and Clock Switch block. VClk can be selected to come straight from the VXTALIN and VXTALOUT crystal oscillator pins. In that case VXTAL frequency has to be 8 times the color subcarrier frequency of the used video format. The other possibility is to use on-chip 8x PLL to generate the VClk. In this case VXTAL frequency is equal to the color subcarrier frequency of the used video format.

After power-up crystal oscillator is selected as VClk. 8x PLL can be selected as VClk by first enabling it and after 8x PLL is locked to incoming VXTAL signal, it can be selected as VClk. The sequence is described in detail in the datasheet of VS23S010D-L. However, the PLL locks very fast and securely, so this can be omitted and PLL selected as VClk without testing the lock too.

If a Multi-IC VS23S010D-L system is used for video generation, it is desired to get all VS23S010D-Ls operating in synch to each other (max +/- 1 VClk cycle phase error is possible). This can be achieved by enabling the PLL and checking that in all VS23S010D-Ls in system PLL is locked to incoming clock. After all PLLs are locked to VXTAL input, setup and enable the Video Display Controller. This same procedure should be used regardless of the selected clocking method (PLL or VXTAL clock).

5.5 Color Modulator

The Color Modulator is enabled always when Video Display Controller is active. If Y data belongs to picture area (i.e. normal lines) then an additional offset of 102 is added to it before Color Modulator. The Color Modulator generates its output using an eight VClk cycles long pattern. The output is an approximation of the formula $out = Y + Usin(2\pi x/8) + Vcos(2\pi x/8)$. The following table shows how the approximation is realized. The output frequency is F_{CSClk}.

Cycle	Output to DAC
0	Y + U
1	Y + 0.75×U + 0.75×V
2	Y + V
3	Y - 0.75×U + 0.75×V
4	Y - U
5	Y - 0.75×U - 0.75×V
6	Y - V
7	Y + 0.75×U - 0.75×V

When PAL mode is enabled the U data is inverted on odd lines. The maximum values from color modulator to DAC are 300 (255 - 0.75 * -32 - 0.75 * -28) for protoline and 405 (102 + 255 - 0.75 * -32 - 0.75 * -32) for picture line. The minimum values are 0 for protoline (usually used for video synch) and 102 for picture line. In direct DAC mode color modulator is bypassed and the maximum value is 510.

To convert RGB video to YUV format the following formula can be used. The color burst has to be set to E2h or similar value (first negative A (V), then positive B (U) value) in the protoline area. If the color burst value is changed then the formula may need to be adjusted too for best results. The conversion is from 8-bit RGB (8 bits for each R, G and B) to 8-bit YUV values:

- $Y = (76 \times R + 150 \times G + 29 \times B) >> 8$
- $U = (R << 7 - 107 \times G - 20 \times B) >> 8$
- $V = (-43 \times R - 84 \times G + B << 7) >> 8$

5.6 Block Move

In Video Display Controller it is possible to move a “rectangular” area of pixel data in SRAM from one position to another position. The principle of block move and parameters are shown in Figure 15. It is possible to move 4 bytes in 5 VClk cycles if there is no simultaneous Video Display Controller memory operation. The Video Display Controller fetches and SPI or parallel interface memory operations override always block move operations and block move operations continue when the memory is not used by any other.

The main parameters of the block move are (see Figure 15):

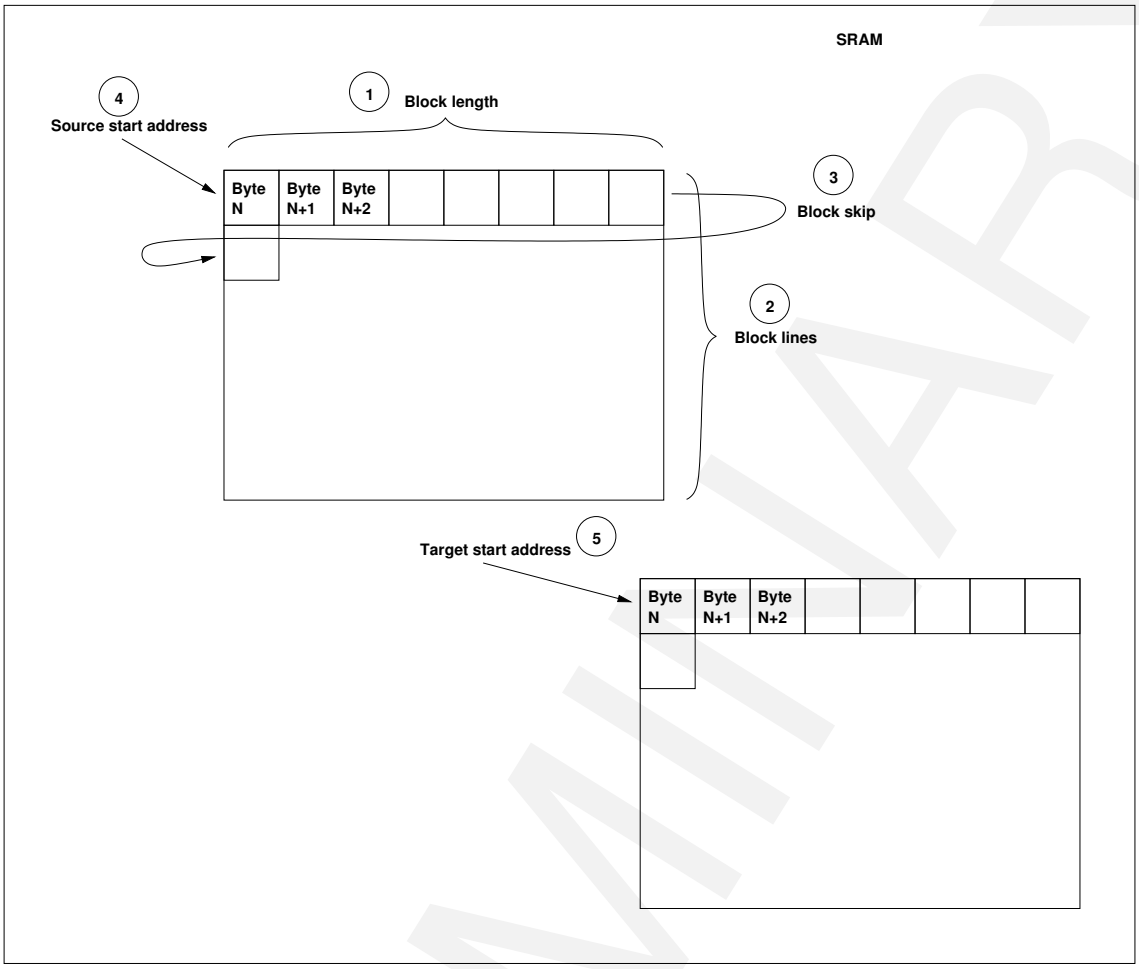


Figure 15: Block move parameters

1. Block length is given in bytes. The range is from 0 to 255.

   Block length is set by Write Block Move Control2 command.

2. Block lines tells how many lines are there in the block. The range is from 1 to 256.

   Block lines is set by Write Block Move Control2 command.

3. Block skip is the amount of bytes between the two lines of block. The range is from 1 to 2048.

   Block skip is set by Write Block Move Control2 command.

4. Source start address is the byte address of the first byte which is transferred to target location. The source address is a 17-bit value ranging from 00000h to 1FFFFh.

   Source start address is set by Write Block Move Control1 command.

5. Target start address is the byte address of the first byte at target location. The target address is a 17-bit value ranging from 00000h to 1FFFFh.

   Target start address is set by Write Block Move Control1 command.

There is still one additional control bit to block move. The direction of the move can be selected. The direction can be from the first byte to last or from the last byte to the first in SRAM. If the

direction is from last to first then the Source and Target start addresses are the last addresses of the block.

The block move is enabled by a single byte SPI command after parameters are set.

5.7 Direct DAC Mode

Direct DAC mode is a simple method of utilizing the VS23S010D-L DAC for other purposes than Video Display Controller. In Direct DAC mode there is possible to use much slower data rates than in Video Display Controller mode. Also the Color Modulator is by-passed. In Direct DAC Mode 8-bit unsigned data is the only supported format. The data is organized in SRAM from a defined start address (INDEX_START shift left by one, this has to be greater than 0h) in increasing order. Line Length value defines the length of data buffer in Direct DAC mode ranging from 1 to 4096.

SRAM address	SRAM data byte
	7 … 0
INDEX_START << 1	DAC0
+1	DAC1
+2	DAC2
+3	DAC3
+4	DAC4
+5	DAC5
+6	DAC6
+7	DAC7
+8	DAC8
⋮	⋮

Figure 16: Direct DAC data organization

In Direct DAC mode eight data bits are sent to MSBs of 9-bit Video DAC. The LSB is always “0”. Note, that the 10 VClk period at the beginning of the line affects also in Direct DAC mode. During those cycles the data to DAC is not updated and value remains the same.

The summary of registers for Direct DAC mode is shown in the following table.

Register	Bit	Description
Video Display Controller Control2	ENA	Enables Video Display Controller
Video Display Controller Control1	DIRDAC	Selects Direct DAC mode
Video Display Controller Control1	PLLENA	Enables Video DAC analog biases
Video Display Controller Control1	DACDIV	Clock Divider in Direct DAC mode
Picture Index Start Address		DAC data buffer start address, > 0h
Line Length		DAC data buffer length

5.8 Operating The Video Display Controller

Video Display Controller is controlled via SPI. First fill SRAM with data, then set Video Display Controller control registers to desired values. The last SPI write is to the register which enables the Video Display Controller. The SRAM data can be updated when Video Display Controller is enabled.

The Video Display Controller logic is reset by setting the XRESET pin low. Setting the XRESET pin high exits the reset state. Entering the reset state is done immediately asynchronously and exiting the reset state requires three VClk cycles. Figure 17 shows the timing of the XRESET pin and active-low, on-chip reset signal. There are two important notes considering the Video Display Controller reset:

- It is not allowed to reset the Video Display Controller, when SPI or parallel interface SRAM operation is in progress so that the correct state of the SRAM is maintained.
- The XRESET pin resets only the Video Display Controller logic and operation. The Video Display Controller control registers are in the SPI block and they are not affected by the XRESET pin. For example, if Video Display Controller is reset when it is active, it will restart again after XRESET is released and VClk is given to VS23S010D-L.

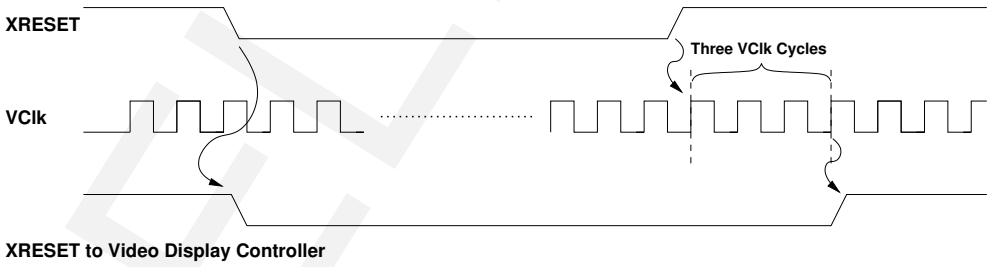


Figure 17: Timing of on-chip reset signal

5.9 Advice for Picture/Proto Border

Depending on Video Display Controller operating parameters (picture end, line length, program length, picture data width etc.) there may be some visible artefacts at the picture/proto border i.e. where line changes from picture mode to proto mode Note, that this is not the case always and the border can be all right.

The problem occurs when more than one simultaneous fetch from the video memory would be needed. The problem can be avoided by aligning the pixel data so that the pixel data for the line does not end at a 32-bit boundary. For example, having the pixel data for each line start at an odd byte address will fix the issue for most common video modes.

For example, a video mode of 320 x 240 pixels, 8 bits per pixel will have the issue if the frame buffer starts at location 2000h. Changing the frame buffer start address to 2001h will fix the issue.

For example, if the first protoline word value after picture area is C4CFh (Ch for V, 4h for U and CFh for Y) then write the extra bytes after the end of picture line as follows: First write the Y byte, CFh and after that the VU byte, C4h. Repeat that three times. The previous example is for a byte wide picture data. For a two byte wide picture data the order of extra bytes is switched, first is written C4h and then CFh. In some cases there is no need for extra bytes in every line. The previous cases are examples and for each parameter set and protolines correct values can be obtained.

5.10 U Table Usage

If TRUV bit is set U and V values for picture area are taken from the U and V Table registers. U table register works so that four selected register bits are put to LSB part of the six bit U output to color modulator. So the output is always positive which limits the usable color space a bit. The V table works as a normal four bit V value.

5.11 Video Example

Following there are figures and tables showing Video Display Controller parameters and some of their possible variations. PAL video is selected as an example.

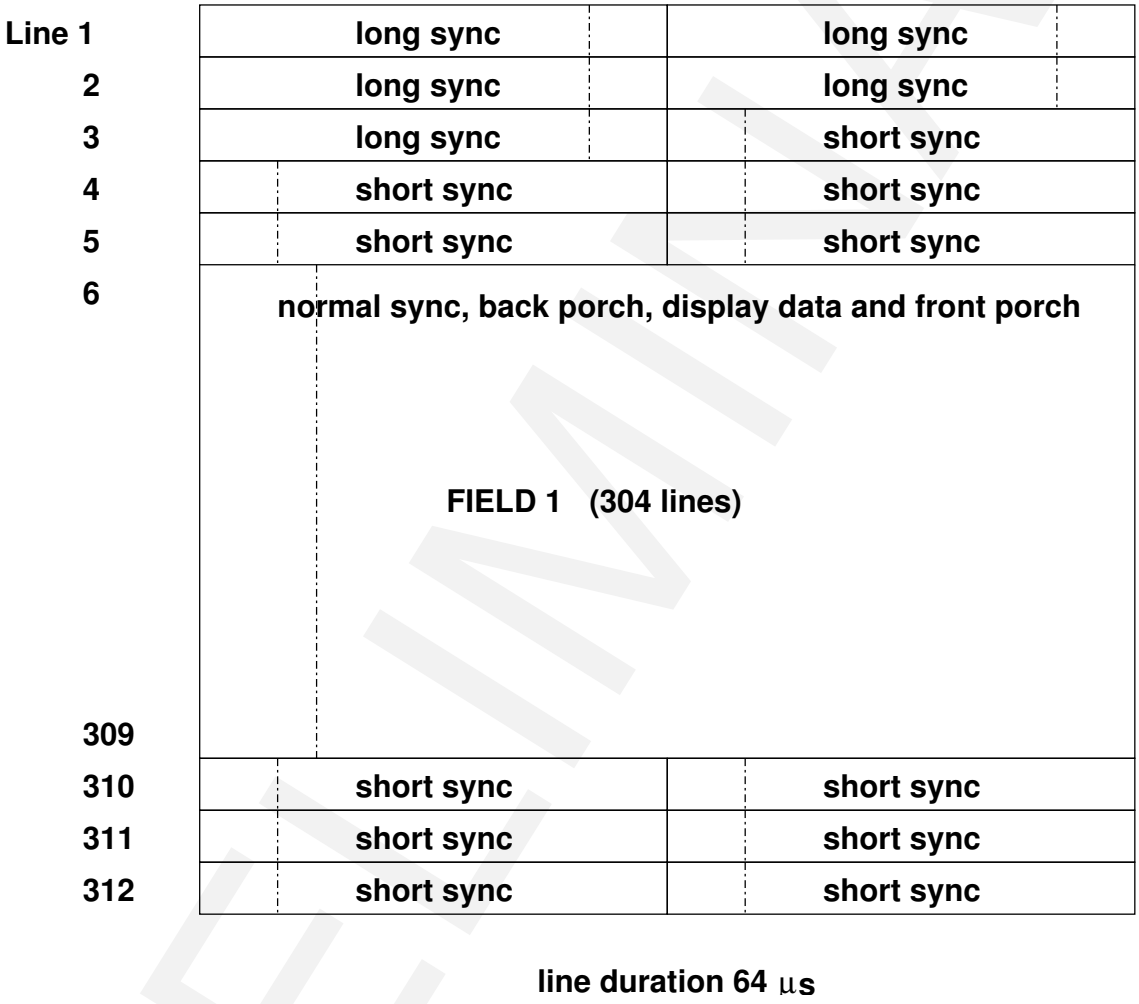


Figure 18: Progressive PAL video frame timing

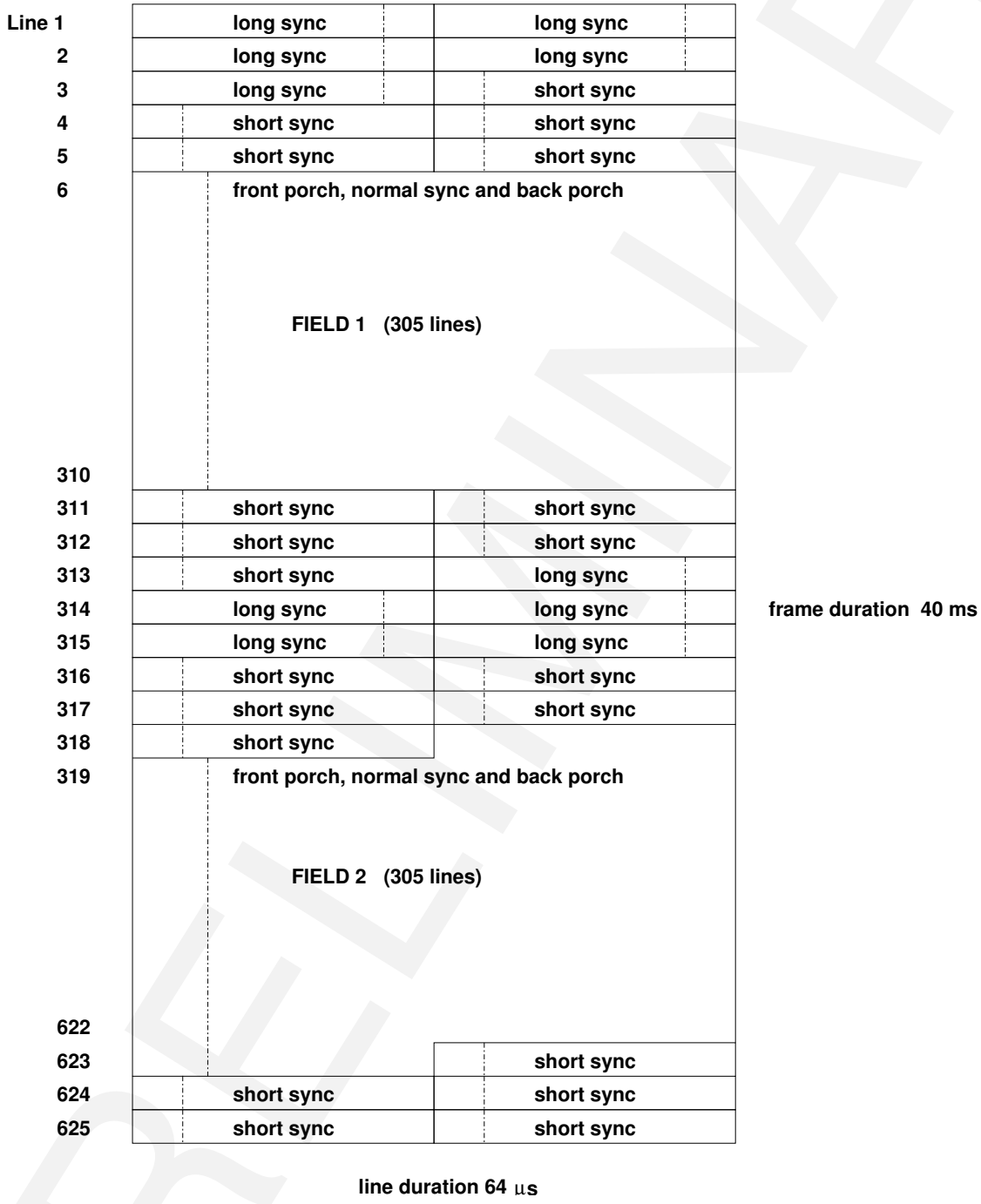


Figure 19: Interlaced PAL video frame timing

Field synchronization of PAL signal can be done using seven different Prototype lines:

- Long sync, long sync line
- Long sync, short sync line
- Short sync, short sync line
- Short sync, long sync line
- Short sync only
- Normal sync line, short sync line
- Normal sync line

Additional Protolines can be used for generating background images for video etc.

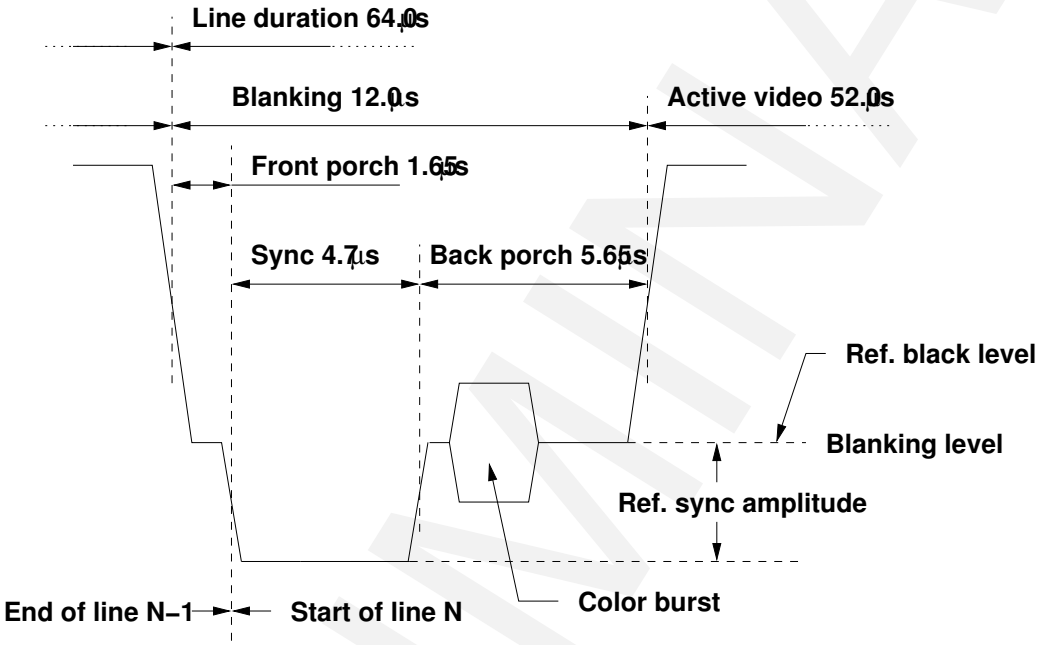


Figure 20: PAL video line timing principle (timing tolerances not shown)

Some PAL video timing parameters are shown on the following table also without timing tolerances.

Format	PAL analog
Field Rate	50 Hz
Frame Rate	25 Hz
Line Count of Picture	625
Vertical Lines Visible	576
Line Count of Frame (theoretical)	312.5
Visible Lines in Frame	288
Line Duration	64 μs
Front Porch	1.65 μs
Sync Pulse Width	4.7 μs
Back Porch	5.65 μs
Color Burst Duration	2.25 μs
Long Sync Width	27.3 μs
Short Sync Width	2.35 μs
Line Frequency	15625 Hz

Following tables are just shown to illustrate how the parameters can be calculated. In the table are used 5, 7 or 3 bits per pixel and they are maybe not the most obvious parameter selections for a video image.

Video Display Controller Parameters for Interlaced PAL	
One Field in the SRAM version 1	
CSClk Frequency	4.43361875 MHz
VClk Frequency	35.46895 MHz
Line Count	312
Visible Lines	288
Line Length	round(64*35.46895)-10 = 2260
Picture Start (Sync + Back Porch)	round(((4.7+5.65)*35.46895-10)/8) = 45
Picture End (Front Porch)	round(((64-1.65)*35.46895-10)/8) = 275
Program Length	3
U & V Bits	1
Y Bits	3
Colors	$2^{(1+1+3)}$ = 32
Visible Pixels per Line	8*(275-45)/3 = 640
Bits Used for Protolines (minimum)	ceiling(2260/8)*16*7 = 31640
Bits Used for Line Indexes	312*3*8 = 7488
Bits Used for Visible Area of Field	(640*(1+1+3))*288 = 921600
Time for Updating the Whole Visible Area	19.92 ms
Write Frequency for Byte	> 5.78 MHz
One Field in the SRAM version 2	
Same as Above Except	
Program Length	4
U & V Bits	2
Y Bits	3
Colors	128
Visible Pixels per Line	480
Bits Used for Visible Area of Field	967680
Write Frequency for Byte	> 6.07 MHz
One Frame in the SRAM	
Same as Top Except	
Line Count	625
Visible Lines	576
Program Length	4
U & V Bits	1
Y Bits	1
Colors	8
Visible Pixels per Line	480
Bits Used for Line Indexes	15000
Bits Used for Visible Area of Frame	829440
Time for Updating the Whole Visible Area	39.91 ms
Write Frequency for Byte	> 2.60 MHz

In the following table is shown how the maximum picture area can be calculated for a 8-bit pixel. The program length is selected as 4 VClk cycles,

Video Display Controller Parameters for Progressive PAL	
Maximum picture area with 8-bit pixel and 4 VClk long program	
CSClk Frequency	4.43361875 MHz
VClk Frequency	35.46895 MHz
Line Count	312
Line Length	2260
Picture Start (Sync + Back Porch)	45
Picture End (Front Porch)	275
Program Length	4
U & V Bits	2
Y Bits	4
Colors	256
Visible Pixels per Line	460
Bits Used for Protolines (minimum)	ceiling(2260/8)*16*5 = 22640
Bits Used for Line Indexes	7488
Bits Free for Picture Area	1048576-22640-7488 = 1018448
Lines for Picture Area (maximum)	floor(1018448/(460*8)) = 276
Bits Used for Visible Area	(460*(2+2+4))*276 = 1015680
Time for Updating the Whole Visible Area	19.92 ms
Write Frequency for Byte	> 6.37 MHz

The video data for Video Display Controller can be generated by a master micro-controller. The video data organization in the SRAM is such that data can be handily formulated by a barrel shifter to a suitable format. Video data generation principle is as follows:

1. Select your program based on your video format, for example. 3 bits U, 3 bits V and 5 bits Y
2. When your video is in the required format, first put the U bits to LSB part of the barrel shifter input register.
3. Next shift left 3 bits and then or the V bits to existing barrel shifter input register value.
4. Next shift left 5 bits and then or the Y bits to existing barrel shifter input register value.
5. Repeat the procedure described here (from 2 to 5) until barrel shifter is full. Then take the 8 MSBs of the barrel shifter and initiate a write to index start address of the current video line. Send the byte to Video Display Controller.
6. Then generate additional bytes using the procedure described in steps 2 to 6.
7. After a line is transferred to Video Display Controller transfer the rest of the lines as described before.

5.12 Microcode Program Example

In Chapter 5.3 is explained how indexes and video pixel data of proto and normal picture lines are arranged in the SRAM of the VS23S010D-L. Following is an example of microcode program use:

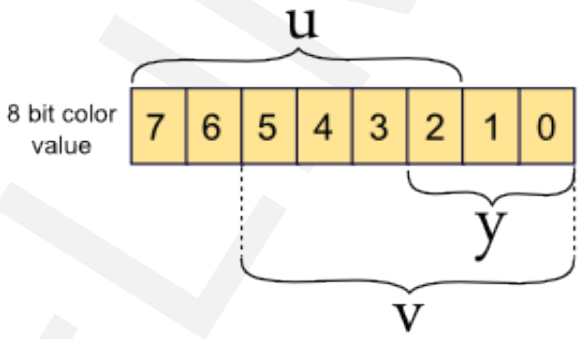
Here's an example and explanation on how to set up a 8-bit color palette on the VS23S010D-L in an efficient way.

VS23S010D-L doesn't have a color look-up table, or "palette memory" as it's often called, because adding one would have been prohibitively expensive. Instead, it has a versatile microcode engine that picks bits from the video memory and assigns those to Y, U and V coefficients in the video modulator. Here's one example of usign the microcode engine in an interesting way to make a nice 8-bit palette (256 colours).

5.12.1 Assigning Bits from Pixel Color Value to Color Coefficients

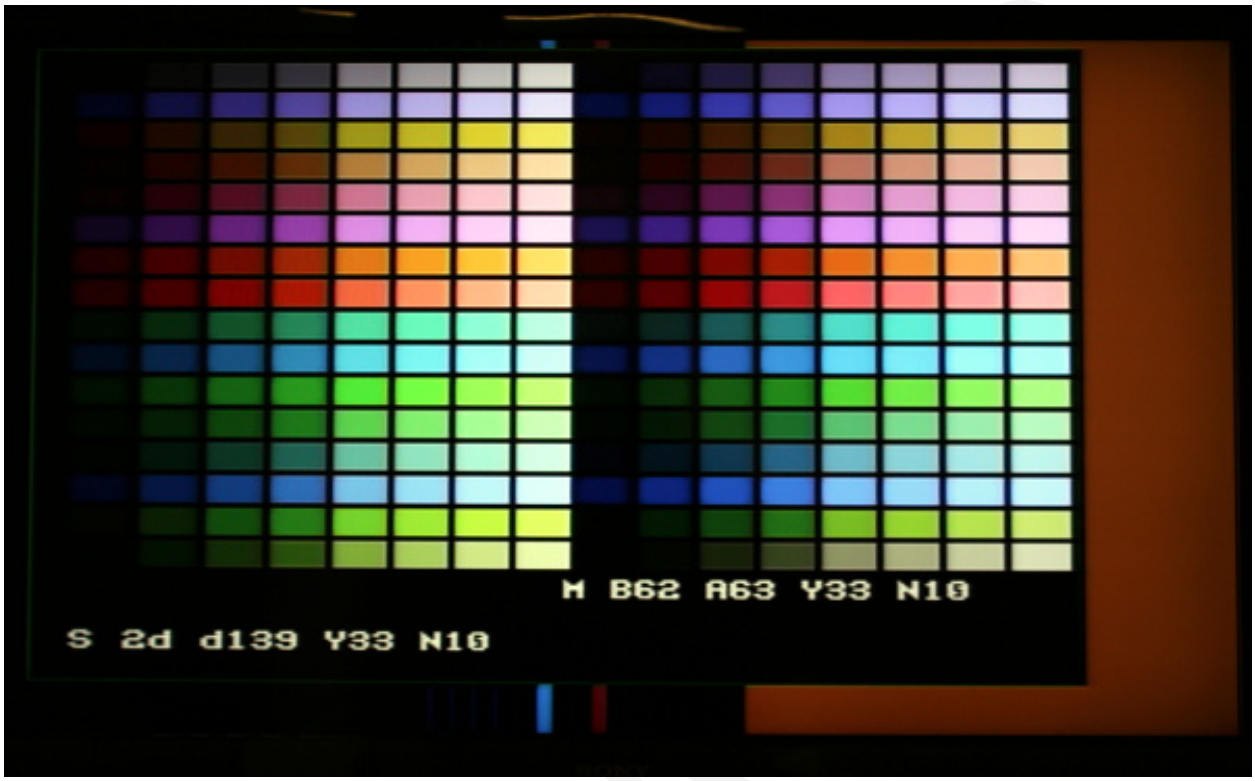
8-bit palettes are attractive from programmer's point of view, because 8-bit colors will have separate pixels in separate bytes (1 byte per pixel), so each pixel is uniquely addressable in the framebuffer memory. When VS23S010D-L is drawing the picture, it will load pixel data from memory into a shift register inside the microcode engine. The microcode engine picks bits from the shift register into color coefficients Y, A and B. Y bits modify luminance, e.g. set the pixel's brightness. A and B commonly modify the V and U components in PAL YUV colorspace, or Q and I components in NTSC YIQ colorspace (depending on the phase value of the color burst).

A clever and surprising feature of the microcode engine is that bit assignment and shifting are done independently from each other, e.g. bits of the pixel color value can be assigned into more than one coefficient. For example, see the assignment below:



This assignment strategy takes, from the 8-bit pixel color value, 6 bits to u color coefficient, 6 bits to v color coefficient and 3 bits to luminance. That results in a large amount of different hues and also dark and bright luminances for most colors.

Below is a picture of the palette, photographed from a modern LCD television:

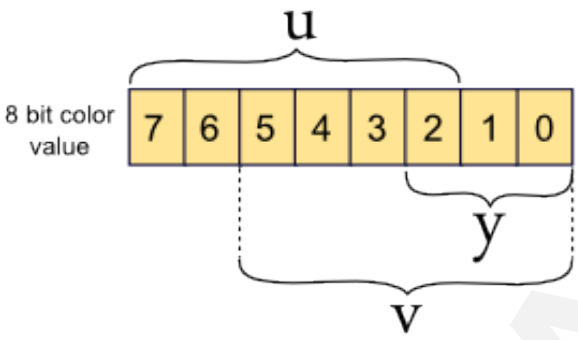


The large span of variation in the color coefficients results in a nice spread of different hues, which are useful for bright user interfaces, games, animations and such. Also a lot of darker tones are available. The first 8 values form a very near grayscale. The next 8 values have a slightly bluish tone and the last 8 values form a grayscale with a seepia tone.

With the burst value of 0xB5, the microcode that forms this palette can be seen in the picture: "B62, A63, Y33, N10", which reads out:

```
- For B coefficient (u) pick 6 bits, then shift out 2 bits.
- For A coefficient (v) pick 6 bits, then shift out 3 bits.
- For Y coefficient (luminance) pick 3 bits, then shift out 3 bits.
- For the final step, pick 1 bit for No coeffient and don't shift any bits.
```

From the picture below, you can see how this microcode correlates to the bit assignments:



The last step of the microcode is needed, because the engine always executes 4 steps from the microcode memory, followed by additional No-Operation steps until the pixel width setting is satisfied.

Colours on the VS23S010D-L are not limited to 8 bits, or even 16 bits; the maximum color depth is 20 bits per pixel (8 bits for Y, 6 bits for U, 6 bits for V). But 8-bit or 16-bit colors are nice because they allow individually addressable pixels.

5.13 Further Examples in VS23S010 Forum

Software examples about VS23S010D-L usage and related discussions can be found on the VSDSP Forum (*http://www.vsdsp-forum.com/*) in the VS23S010 Forum section.

6 Document Version Changes

This chapter describes the most important changes to this document.

Version 0.1, 2016-12-13

- Created VS23S010D-LGuide by taking some parts of the VS23S010D-LDatasheet and adding new information.

7 Contact Information

VLSI Solution Oy
Entrance G, 2nd floor
Hermiankatu 8
FI-33720 Tampere
FINLAND

URL: http://www.vlsi.fi/
Phone: +358-50-462-3200
Commercial e-mail: sales@vlsi.fi

For technical support or suggestions regarding this document, please participate at
http://www.vsdsp-forum.com/
For confidential technical discussions, contact
support@vlsi.fi