1 Experiment Introduction

HDMI as a video output input interface has been widely used for a long time, mainly through TMDS differential encoding transmission. This experiment demonstrates the timing of the video and the representation of the video color, by displaying the color bar and the input and output loop-through experiment on the HDMI screen, to provide a basis for the subsequent video processing experiments.

2 Experiment Principle

2.1 Hardware Introduction

HDMI uses the same transmission principle as DVI - TMDS (Transition Minimized Differential signal), which minimizes the transmission of differential signals.

The TMDS transmission system is divided into two parts: the transmitting end and the receiving end. The TMDS transmitting end receives 24-bit parallel data representing the RGB signal transmitted from the HDMI interface (TMDS encodes the RGB three primary colors of each pixel by 8 bits respectively, that is, the R signal has 8 bits, the G signal has 8 bits, and the B signal has 8 bits） . Then, the data is encoded and parallel/serial conversion, and the data representing the three RGB signals are respectively distributed to separate transmission channels for transmission. The receiving end receives the serial signal from the transmitting end, decodes it and serial/parallel conversion, and then transmits it to the control terminal of the display. At the same time, the clock signal is also received to achieve synchronization.

The principle of TMDS

Each TMDS link includes three data channels for transmitting RGB signals and one channel for transmitting clock signals. Each data channel uses an encoding algorithm to convert 8-bit video and audio data into 10-bit data that minimizes transmission and DC balance. This makes data transmission and recovery more reliable. Minimizing the transmission of differential signals is to convert the original 8-bit signal data into 10 bits by XOR and XOR logic, The 8-bit signal data is converted into 10 bits, the first 8 data is obtained from the original signal, the 9th bit indicates the operation mode, and the 10th bit is used to correspond to the DC balance.

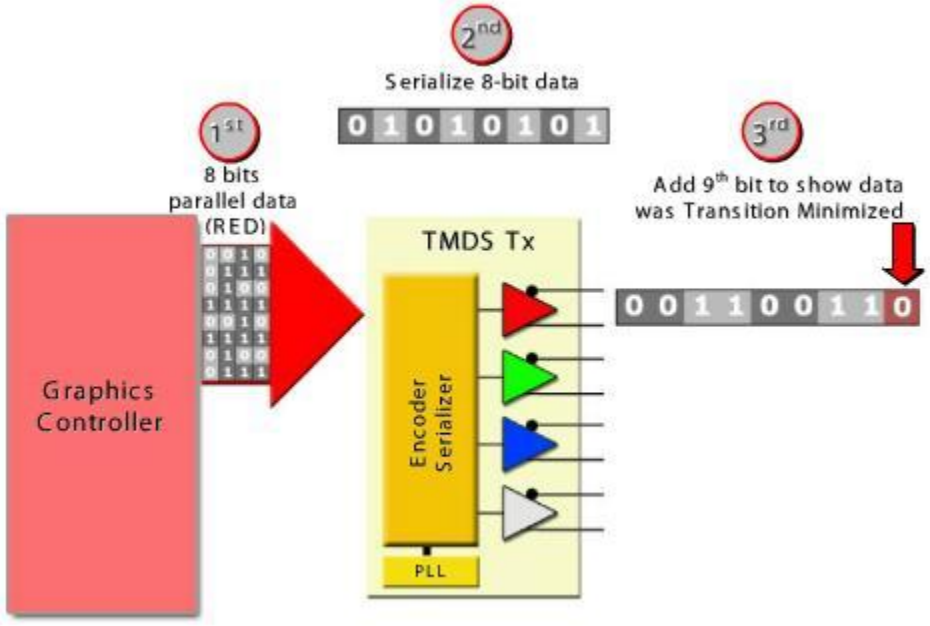
Generally, the encoding format of the HDMI transmission includes video data, control data, and data packets (the data packet contains audio data and additional information data, such as error correction codes, etc.). Each channel of TMDS should contain 2bit control data, 8bit video data or 4bit data packets when transmitting. In the HDMI information transmission process, it can be divided into

three phases: a video data transmission cycle, a control data transmission cycle, and a data island transmission cycle, which respectively correspond to the above three data types.

The techniques used in TMDS are described below:

1. Minimize transmission

The 8-bit data is encoded and DC balanced to obtain 10-bit minimized data, which seems to increase redundant bits and higher bandwidth requirements for the transmission link, but in fact, the 10-bit data obtained by this algorithm is longer. The reliability of transmission in coaxial cable is enhanced. The figure below is an example of encoding an 8-bit parallel RED data and parallel/serial conversion.



Step 1: Send 8-bit parallel RED data to the TMDS sender

Step 2: Serial/Parallel conversion

Step 3: Minimize the transfer process, plus the 9th bit, which is the encoding process. The 9th bit of data is called the coded bit.。

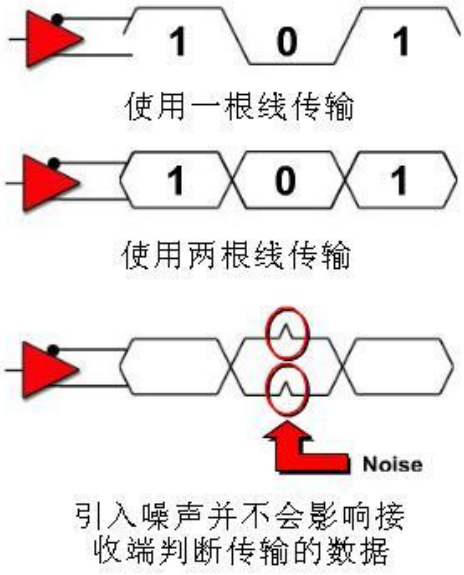
2. DC-balanced

DC-balanced means that the DC offset in the channel is guaranteed to be zero during the encoding process. The method is to add the 10th data after the original 9-bit data, and the transmitted data tends to be DC balanced, so that the electromagnetic interference of the signal to the transmission line is reduced, and the reliability of the signal transmission is improved.

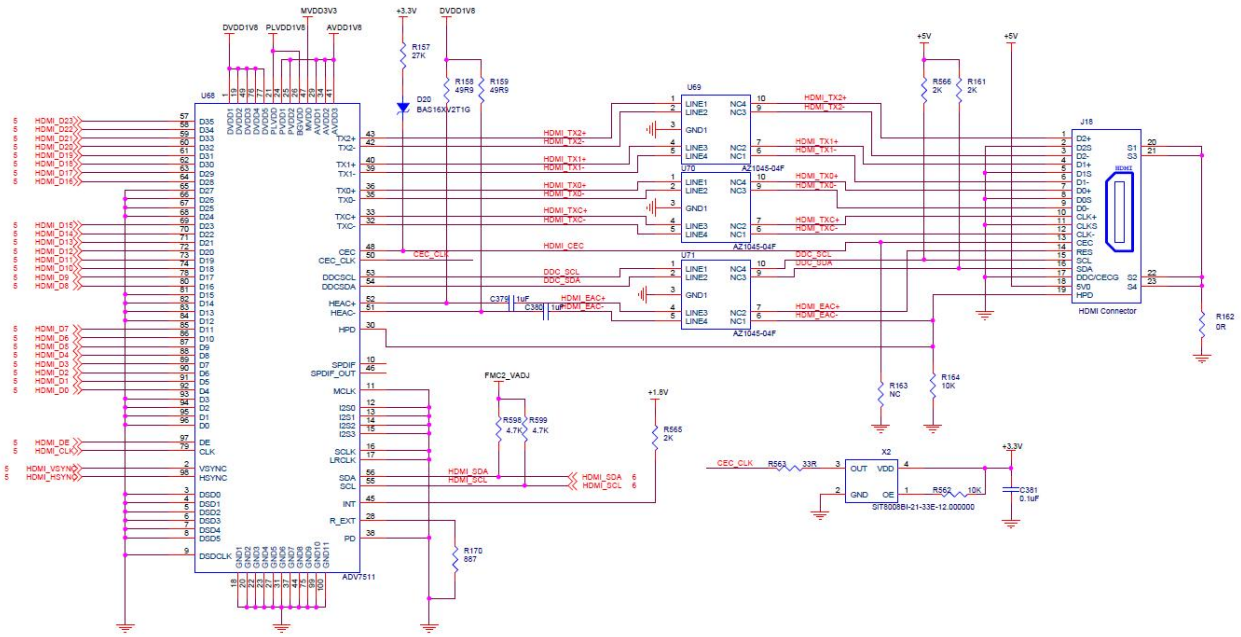
3. Differential signal

TMDS differential drive technology is a technology that uses a voltage difference between two pins to transmit a signal. The value of the transmitted data ("0" or "1") is determined by the positive and negative polarity and magnitude of the voltage between the two legs. That is, two lines are used to transmit signals, one line transmits the original signal, and the other line transmits the opposite signal to the original signal. In this way, the receiving end can shield the electromagnetic interference by subtracting the signal on one line from the signal on the other line, thereby obtaining the correct signal.

As shown below:



In addition, there is a display data channel (DDC) which is a signal line for reading extended display identification data (EDID) indicating display capability such as display resolution of the receiving end. The DDC line is also used to authenticate the cryptographic keys between the transmitting and receiving devices equipped with HDCP (High-bandwidth Digital Content Protection).



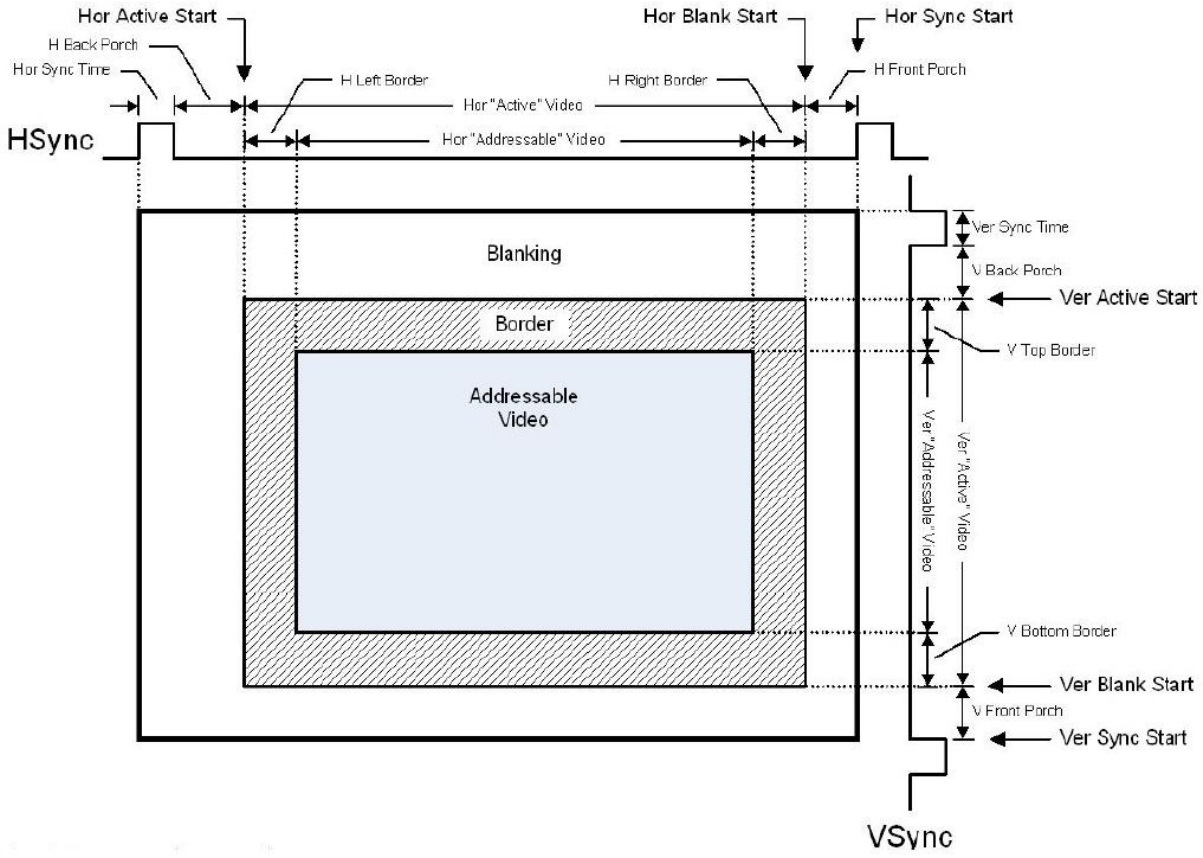
AXKU040 FPGA Board HDMI Interface Schematic

2.2 Video timing standard

The HDMI display scanning mode starts from the top left corner of the screen and scans from left to right point by point. After each line is scanned, the electron beam returns to the beginning of the next line on the left side of the screen. During this period, the CRT blanks the electron beam. At the end of each line, the line sync signal is used for synchronization; when all the lines are scanned, a frame is formed, the field sync signal is used for field sync, and the scan is returned to the upper left of the screen, and make the scan back to the top left of the screen, while performing field blanking, start the next frame.

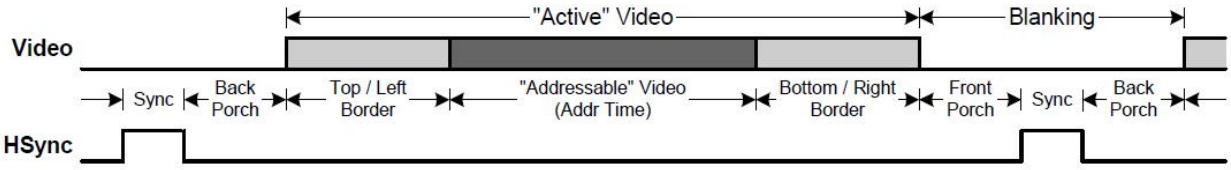
The time to complete a line scan is called horizontal scan time, and the reciprocal is called line frequency; the time to complete one frame (full screen) scan is called vertical scan time, and the reciprocal is called field frequency, that is, the frequency of refreshing one screen, Common are 60Hz, 75Hz and so on. The standard display has a field rate of 60 Hz.

Clock frequency: Take “1024x768@59.94Hz (60Hz)” as an example. Each field corresponds to 806 line periods, of which 768 is the display line. Each display line includes a 1344-point clock, of which 1024 points are valid display areas. It can be seen that the clock frequency needs to be “806*1344*60” and about 65MHz.



Video Timing

"VGA" scan, the basic element is line scan, multiple lines make up one frame, the following figure shows the timing of one line, where "Active" Video is the effective pixel of one line of video, Top/Left Border and Bottom / Right Border are 0 in most resolution clocks. "Blanking" is the synchronization time of one line. "Blanking" time plus "Active "Video time is one line of time. "Blanking" is divided into three parts: "Front Porch", "Sync" and "Back Porch".



Line synchronization timing

2.3 Common resolution timing

Timing Name	= **640 x 480 @ 60Hz;**						
Hor Pixels	= **640;**	// Pixels					
Ver Pixels	= **480;**	// Lines					
Hor Frequency	= 31.469;	// kHz	=	31.8 usec	/	line	
Ver Frequency	= 59.940;	// Hz	=	16.7 msec	/	frame	
Pixel Clock	= **25.175;**	// MHz	=	39.7 nsec	± 0.5%		
Character Width	= **8;**	// Pixels	=	317.8 nsec			
Scan Type	= **NONINTERLACED;**			// H Phase	=	2.0 %	
Hor Sync Polarity	= **NEGATIVE;**	// HBlank	=	18.0% of HTotal			
Ver Sync Polarity	= **NEGATIVE;**	// VBlank	=	5.5% of VTotal			
Hor Total Time	= 31.778;	// (usec)	=	100 chars	=	800 Pixels	
Hor Addr Time	= 25.422;	// (usec)	=	80 chars	=	640 Pixels	
Hor Blank Start	= 25.740;	// (usec)	=	81 chars	=	648 Pixels	
Hor Blank Time	= 5.720;	// (usec)	=	18 chars	=	144 Pixels	
Hor Sync Start	= 26.058;	// (usec)	=	82 chars	=	656 Pixels	
// H Right Border	= 0.318;	// (usec)	=	**1** chars	=	8 Pixels	
// H Front Porch	= 0.318;	// (usec)	=	**1** chars	=	8 Pixels	
Hor Sync Time	= 3.813;	// (usec)	=	**12** chars	=	96 Pixels	
// H Back Porch	= 1.589;	// (usec)	=	**5** chars	=	40 Pixels	
// H Left Border	= 0.318;	// (usec)	=	**1** chars	=	8 Pixels	
Ver Total Time	= 16.683;	// (msec)	=	525 lines	HT – (1.06xHA)		
Ver Addr Time	= 15.253;	// (msec)	=	480 lines	= 4.83		
Ver Blank Start	= 15.507;	// (msec)	=	488 lines			
Ver Blank Time	= 0.922;	// (msec)	=	29 lines			
Ver Sync Start	= 15.571;	// (msec)	=	490 lines			
// V Bottom Border	= 0.254;	// (msec)	=	**8** lines			
// V Front Porch	= 0.064;	// (msec)	=	**2** lines			
Ver Sync Time	= 0.064;	// (msec)	=	**2** lines			
// V Back Porch	= 0.794;	// (msec)	=	**25** lines			
// V Top Border	= 0.254;	// (msec)	=	**8** lines			

640x480@60Hz Timing Parameter

Timing Name	= **800 x 600 @ 60Hz;**						
Hor Pixels	= **800;**	// Pixels					
Ver Pixels	= **600;**	// Lines					
Hor Frequency	= 37.879;	// kHz	=	26.4 usec	/	line	
Ver Frequency	= 60.317;	// Hz	=	16.6 msec	/	frame	
Pixel Clock	= **40.000;**	// MHz	=	25.0 nsec		± 0.5%	
Character Width	= **8;**	// Pixels	=	200.0 nsec			
Scan Type	= **NONINTERLACED;**			// H Phase	=	2.3 %	
Hor Sync Polarity	= **POSITIVE;**	// HBlank	=	24.2% of HTotal			
Ver Sync Polarity	= **POSITIVE;**	// VBlank	=	4.5% of VTotal			
Hor Total Time	= 26.400;	// (usec)	=	132 chars	=	1056 Pixels	
Hor Addr Time	= 20.000;	// (usec)	=	100 chars	=	800 Pixels	
Hor Blank Start	= 20.000;	// (usec)	=	100 chars	=	800 Pixels	
Hor Blank Time	= 6.400;	// (usec)	=	32 chars	=	256 Pixels	
Hor Sync Start	= 21.000;	// (usec)	=	105 chars	=	840 Pixels	
// H Right Border	= 0.000;	// (usec)	=	**0** chars	=	0 Pixels	
// H Front Porch	= 1.000;	// (usec)	=	**5** chars	=	40 Pixels	
Hor Sync Time	= 3.200;	// (usec)	=	**16** chars	=	128 Pixels	
// H Back Porch	= 2.200;	// (usec)	=	**11** chars	=	88 Pixels	
// H Left Border	= 0.000;	// (usec)	=	**0** chars	=	0 Pixels	
Ver Total Time	= 16.579;	// (msec)	=	628 lines		HT – (1.06xHA)	
Ver Addr Time	= 15.840;	// (msec)	=	600 lines		= 5.2	
Ver Blank Start	= 15.840;	// (msec)	=	600 lines			
Ver Blank Time	= 0.739;	// (msec)	=	28 lines			
Ver Sync Start	= 15.866;	// (msec)	=	601 lines			
// V Bottom Border	= 0.000;	// (msec)	=	**0** lines			
// V Front Porch	= 0.026;	// (msec)	=	**1** lines			
Ver Sync Time	= 0.106;	// (msec)	=	**4** lines			
// V Back Porch	= 0.607;	// (msec)	=	**23** lines			
// V Top Border	= 0.000;	// (msec)	=	**0** lines			

800x600@60Hz Timing Parameter

Timing Name	= **1024 x 768 @ 60Hz;**					
Hor Pixels	= **1024;**	// Pixels				
Ver Pixels	= **768;**	// Lines				
Hor Frequency	= 48.363;	// kHz	=	20.7 usec	/	line
Ver Frequency	= 60.004;	// Hz	=	16.7 msec	/	frame
Pixel Clock	= **65.000;**	// MHz	=	15.4 nsec	± 0.5%	
Character Width	= **8;**	// Pixels	=	123.1 nsec		
Scan Type	= **NONINTERLACED;**			// H Phase =	5.1 %	
Hor Sync Polarity	= **NEGATIVE;**	// HBlank	=	23.8% of HTotal		
Ver Sync Polarity	= **NEGATIVE;**	// VBlank	=	4.7% of VTotal		
Hor Total Time	= 20.677;	// (usec)	=	168 chars	=	1344 Pixels
Hor Addr Time	= 15.754;	// (usec)	=	128 chars	=	1024 Pixels
Hor Blank Start	= 15.754;	// (usec)	=	128 chars	=	1024 Pixels
Hor Blank Time	= 4.923;	// (usec)	=	40 chars	=	320 Pixels
Hor Sync Start	= 16.123;	// (usec)	=	131 chars	=	1048 Pixels
// H Right Border	= 0.000;	// (usec)	=	**0** chars	=	0 Pixels
// H Front Porch	= 0.369;	// (usec)	=	**3** chars	=	24 Pixels
Hor Sync Time	= 2.092;	// (usec)	=	**17** chars	=	136 Pixels
// H Back Porch	= 2.462;	// (usec)	=	**20** chars	=	160 Pixels
// H Left Border	= 0.000;	// (usec)	=	**0** chars	=	0 Pixels
Ver Total Time	= 16.666;	// (msec)	=	806 lines	HT – (1.06xHA)	
Ver Addr Time	= 15.880;	// (msec)	=	768 lines	= 3.98	
Ver Blank Start	= 15.880;	// (msec)	=	768 lines		
Ver Blank Time	= 0.786;	// (msec)	=	38 lines		
Ver Sync Start	= 15.942;	// (msec)	=	771 lines		
// V Bottom Border	= 0.000;	// (msec)	=	**0** lines		
// V Front Porch	= 0.062;	// (msec)	=	**3** lines		
Ver Sync Time	= 0.124;	// (msec)	=	**6** lines		
// V Back Porch	= 0.600;	// (msec)	=	**29** lines		
// V Top Border	= 0.000;	// (msec)	=	**0** lines		

1024x768@60Hz Timing Parameter

Timing Name	=	**1280 x 720 @ 60Hz;**						
Hor Pixels	=	**1280;**	// Pixels					
Ver Pixels	=	**720;**	// Lines					
Hor Frequency	=	45.000;	// KHz	=	22.2 usec	/	line	
Ver Frequency	=	60.000;	// Hz	=	16.7 msec	/	frame	
Pixel Clock	=	**74.250;**	// MHz	=	13.5 nsec		± 0.5%	
Character Width	=	**1;**	// Pixels	=	13.5 nsec			
Scan Type	=	**NONINTERLACED;**			// H Phase	=	3.3 %	
Hor Sync Polarity	=	**POSITIVE;**	// HBlank	=	22.4% of HTotal			
Ver Sync Polarity	=	**POSITIVE;**	// VBlank	=	4.0% of VTotal			
Hor Total Time	=	22.222;	// (usec)	=	1650 chars	=	1650 Pixels	
Hor Addr Time	=	17.239;	// (usec)	=	1280 chars	=	1280 Pixels	
Hor Blank Start	=	17.239;	// (usec)	=	1280 chars	=	1280 Pixels	
Hor Blank Time	=	4.983;	// (usec)	=	370 chars	=	370 Pixels	
Hor Sync Start	=	18.721;	// (usec)	=	1390 chars	=	1390 Pixels	
// H Right Border	=	0.000;	// (usec)	=	**0** chars	=	0 Pixels	
// H Front Porch	=	1.481;	// (usec)	=	**110** chars	=	110 Pixels	
Hor Sync Time	=	0.539;	// (usec)	=	**40** chars	=	40 Pixels	
// H Back Porch	=	2.963;	// (usec)	=	**220** chars	=	220 Pixels	
// H Left Border	=	0.000;	// (usec)	=	**0** chars	=	0 Pixels	
Ver Total Time	=	16.667;	// (msec)	=	750 lines		HT – (1.06xHA)	
Ver Addr Time	=	16.000;	// (msec)	=	720 lines		= 3.95	
Ver Blank Start	=	16.000;	// (msec)	=	720 lines			
Ver Blank Time	=	0.667;	// (msec)	=	30 lines			
Ver Sync Start	=	16.111;	// (msec)	=	725 lines			
// V Bottom Border	=	0.000;	// (msec)	=	**0** lines			
// V Front Porch	=	0.111;	// (msec)	=	**5** lines			
Ver Sync Time	=	0.111;	// (msec)	=	**5** lines			
// V Back Porch	=	0.444;	// (msec)	=	**20** lines			
// V Top Border	=	0.000;	// (msec)	=	**0** lines			

1280x720@60Hz Timing Parameter

Timing Name	= **1280 x 800 @ 60Hz;**						
Hor Pixels	= **1280;**	// Pixels					
Ver Pixels	= **800;**	// Lines					
Hor Frequency	= 49.702;	// kHz	=	20.1 usec	/	line	
Ver Frequency	= 59.810;	// Hz	=	16.7 msec	/	frame	
Pixel Clock	= **83.500;**	// MHz	=	12.0 nsec		± 0.5%	
Character Width	= **8;**	// Pixels	=	95.8 nsec			
Scan Type	= **NONINTERLACED;**			// H Phase	=	3.8 %	
Hor Sync Polarity	= **NEGATIVE;**	// HBlank	=	23.8% of HTotal			
Ver Sync Polarity	= **POSITIVE;**	// VBlank	=	3.7% of VTotal			
Hor Total Time	= 20.120;	// (usec)	=	210 chars	=	1680 Pixels	
Hor Addr Time	= 15.329;	// (usec)	=	160 chars	=	1280 Pixels	
Hor Blank Start	= 15.329;	// (usec)	=	160 chars	=	1280 Pixels	
Hor Blank Time	= 4.790;	// (usec)	=	50 chars	=	400 Pixels	
Hor Sync Start	= 16.192;	// (usec)	=	169 chars	=	1352 Pixels	
// H Right Border	= 0.000;	// (usec)	=	**0** chars	=	0 Pixels	
// H Front Porch	= 0.862;	// (usec)	=	**9** chars	=	72 Pixels	
Hor Sync Time	= 1.533;	// (usec)	=	**16** chars	=	128 Pixels	
// H Back Porch	= 2.395;	// (usec)	=	**25** chars	=	200 Pixels	
// H Left Border	= 0.000;	// (usec)	=	**0** chars	=	0 Pixels	
Ver Total Time	= 16.720;	// (msec)	=	831 lines		HT – (1.06xHA)	
Ver Addr Time	= 16.096;	// (msec)	=	800 lines		= 3.87	
Ver Blank Start	= 16.096;	// (msec)	=	800 lines			
Ver Blank Time	= 0.624;	// (msec)	=	31 lines			
Ver Sync Start	= 16.156;	// (msec)	=	803 lines			
// V Bottom Border	= 0.000;	// (msec)	=	**0** lines			
// V Front Porch	= 0.060;	// (msec)	=	**3** lines			
Ver Sync Time	= 0.121;	// (msec)	=	**6** lines			
// V Back Porch	= 0.443;	// (msec)	=	**22** lines			
// V Top Border	= 0.000;	// (msec)	=	**0** lines			

1280x800@60Hz Timing Parameter

Timing Name	= **1280 x 960 @ 60Hz;**						
Hor Pixels	= **1280;**	// Pixels					
Ver Pixels	= **960;**	// Lines					
Hor Frequency	= 60.000;	// kHz	=	16.7 usec	/	line	
Ver Frequency	= 60.000;	// Hz	=	16.7 msec	/	frame	
Pixel Clock	= **108.000;**	// MHz	=	9.3 nsec	± 0.5%		
Character Width	= **8;**	// Pixels	=	74.1 nsec			
Scan Type	= **NONINTERLACED;**			// H Phase	=	6.0 %	
Hor Sync Polarity	= **POSITIVE;**	// HBlank	=	28.9% of HTotal			
Ver Sync Polarity	= **POSITIVE;**	// VBlank	=	4.0% of VTotal			
Hor Total Time	= 16.667;	// (usec)	=	225 chars	=	1800 Pixels	
Hor Addr Time	= 11.852;	// (usec)	=	160 chars	=	1280 Pixels	
Hor Blank Start	= 11.852;	// (usec)	=	160 chars	=	1280 Pixels	
Hor Blank Time	= 4.815;	// (usec)	=	65 chars	=	520 Pixels	
Hor Sync Start	= 12.741;	// (usec)	=	172 chars	=	1376 Pixels	
// H Right Border	= 0.000;	// (usec)	=	**0** chars	=	0 Pixels	
// H Front Porch	= 0.889;	// (usec)	=	**12** chars	=	96 Pixels	
Hor Sync Time	= 1.037;	// (usec)	=	**14** chars	=	112 Pixels	
// H Back Porch	= 2.889;	// (usec)	=	**39** chars	=	312 Pixels	
// H Left Border	= 0.000;	// (usec)	=	**0** chars	=	0 Pixels	
Ver Total Time	= 16.667;	// (msec)	=	1000 lines	HT – (1.06xHA)		
Ver Addr Time	= 16.000;	// (msec)	=	960 lines	= 4.1		
Ver Blank Start	= 16.000;	// (msec)	=	960 lines			
Ver Blank Time	= 0.667;	// (msec)	=	40 lines			
Ver Sync Start	= 16.017;	// (msec)	=	961 lines			
// V Bottom Border	= 0.000;	// (msec)	=	**0** lines			
// V Front Porch	= 0.017;	// (msec)	=	**1** lines			
Ver Sync Time	= 0.050;	// (msec)	=	**3** lines			
// V Back Porch	= 0.600;	// (msec)	=	**36** lines			
// V Top Border	= 0.000;	// (msec)	=	**0** lines			

1280x960@60Hz Timing Parameter

Timing Name	= **1280 x 1024 @ 60Hz;**						
Hor Pixels	= **1280;**	// Pixels					
Ver Pixels	= **1024;**	// Lines					
Hor Frequency	= 63.981;	// kHz	=	15.6 usec	/	line	
Ver Frequency	= 60.020;	// Hz	=	16.7 msec	/	frame	
Pixel Clock	= **108.000;**	// MHz	=	9.3 nsec		± 0.5%	
Character Width	= **8;**	// Pixels	=	74.1 nsec			
Scan Type	= **NONINTERLACED;**			// H Phase	=	5.9 %	
Hor Sync Polarity	= **POSITIVE;**	// HBlank	=	24.2% of HTotal			
Ver Sync Polarity	= **POSITIVE;**	// VBlank	=	3.9% of VTotal			
Hor Total Time	= 15.630;	// (usec)	=	211 chars	=	1688 Pixels	
Hor Addr Time	= 11.852;	// (usec)	=	160 chars	=	1280 Pixels	
Hor Blank Start	= 11.852;	// (usec)	=	160 chars	=	1280 Pixels	
Hor Blank Time	= 3.778;	// (usec)	=	51 chars	=	408 Pixels	
Hor Sync Start	= 12.296;	// (usec)	=	166 chars	=	1328 Pixels	
// H Right Border	= 0.000;	// (usec)	=	**0** chars	=	0 Pixels	
// H Front Porch	= 0.444;	// (usec)	=	**6** chars	=	48 Pixels	
Hor Sync Time	= 1.037;	// (usec)	=	**14** chars	=	112 Pixels	
// H Back Porch	= 2.296;	// (usec)	=	**31** chars	=	248 Pixels	
// H Left Border	= 0.000;	// (usec)	=	**0** chars	=	0 Pixels	
Ver Total Time	= 16.661;	// (msec)	=	1066 lines		HT – (1.06xHA)	
Ver Addr Time	= 16.005;	// (msec)	=	1024 lines		= 3.07	
Ver Blank Start	= 16.005;	// (msec)	=	1024 lines			
Ver Blank Time	= 0.656;	// (msec)	=	42 lines			
Ver Sync Start	= 16.020;	// (msec)	=	1025 lines			
// V Bottom Border	= 0.000;	// (msec)	=	**0** lines			
// V Front Porch	= 0.016;	// (msec)	=	**1** lines			
Ver Sync Time	= 0.047;	// (msec)	=	**3** lines			
// V Back Porch	= 0.594;	// (msec)	=	**38** lines			
// V Top Border	= 0.000;	// (msec)	=	**0** lines			

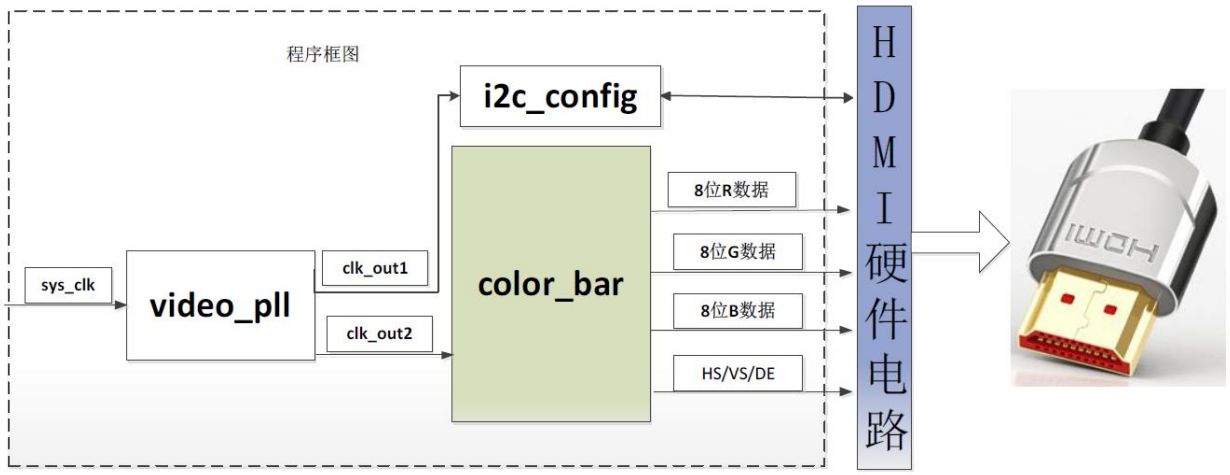
1280x1024@60Hz Timing Parameter

Timing Name	= **1920 x 1080 @ 60Hz;**						
Hor Pixels	= **1920;**	// Pixels					
Ver Pixels	= **1080;**	// Lines					
Hor Frequency	= 67.500;	// kHz	=	14.8 usec	/	line	
Ver Frequency	= 60.000;	// Hz	=	16.7 msec	/	frame	
Pixel Clock	= **148.500;**	// MHz	=	6.7 nsec		± 0.5%	
Character Width	= **4;**	// Pixels	=	26.9 nsec			
Scan Type	= **NONINTERLACED;**			// H Phase	=	1.4 %	
Hor Sync Polarity	= **POSITIVE**	// HBlank	=	12.7% of HTotal			
Ver Sync Polarity	= **POSITIVE**	// VBlank	=	4.0% of VTotal			
Hor Total Time	= 14.815;	// (usec)	=	550 chars	=	2200 Pixels	
Hor Addr Time	= 12.929;	// (usec)	=	480 chars	=	1920 Pixels	
Hor Blank Start	= 12.929;	// (usec)	=	480 chars	=	1920 Pixels	
Hor Blank Time	= 1.886;	// (usec)	=	70 chars	=	280 Pixels	
Hor Sync Start	= 13.522;	// (usec)	=	502 chars	=	2008 Pixels	
// H Right Border	= 0.000;	// (usec)	=	**0** chars	=	0 Pixels	
// H Front Porch	= 0.593;	// (usec)	=	**22** chars	=	88 Pixels	
Hor Sync Time	= 0.296;	// (usec)	=	**11** chars	=	44 Pixels	
// H Back Porch	= 0.997;	// (usec)	=	**37** chars	=	148 Pixels	
// H Left Border	= 0.000;	// (usec)	=	**0** chars	=	0 Pixels	
Ver Total Time	= 16.667;	// (msec)	=	1125 lines		HT – (1.06xHA)	
Ver Addr Time	= 16.000;	// (msec)	=	1080 lines		= 1.11	
Ver Blank Start	= 16.000;	// (msec)	=	1080 lines			
Ver Blank Time	= 0.667;	// (msec)	=	45 lines			
Ver Sync Start	= 16.059;	// (msec)	=	1084 lines			
// V Bottom Border	= 0.000;	// (msec)	=	**0** lines			
// V Front Porch	= 0.059;	// (msec)	=	**4** lines			
Ver Sync Time	= 0.074;	// (msec)	=	**5** lines			
// V Back Porch	= 0.533;	// (msec)	=	**36** lines			
// V Top Border	= 0.000;	// (msec)	=	**0** lines			

1920x1080@60Hz Timing Parameter

3 HDMI Output programming

This experiment will implement an HDMI output display with test image color bars displayed on the HDMI display. The program is implemented by three modules: clock module sys_pll, color bar generation module color_bar and HDMI interface configuration module i2c_config. The logical block diagram of the implementation is as follows:



1. Clock module video_pll

The video_pll module is a multiplier generation module. The input sys_clk clock has two sets of clock signals clk_out1 and clk_out2, clk_out2 outputs clock 100MHz, which is used as the clock of the HDMI interface configuration module; clk_out1 is used to clock the color_bar module and HDMI hardware circuit; Set to 148.5MHz to meet the HDMI display resolution of 1920x1080p.

2. Color bar generation module “color_bar.v”

“Color_bar.v” that produces 8 colors, which are white, yellow, cyan, green, purple, red, blue, and black. For VGA timing, row sync and field sync each use a counter, the line sync counter is used to generate line sync, the line is valid pixels, row valid pixels. And the field sync counter is used to generate field sync, field valid pixels. At the same time, horizontal (X) and vertical (Y) coordinates can be generated according to the value of the counter, and some graphics can be displayed in real time through the coordinate information. The HDMI1080P output display is implemented using the hardware chip ADV7511 in the HDMI output.

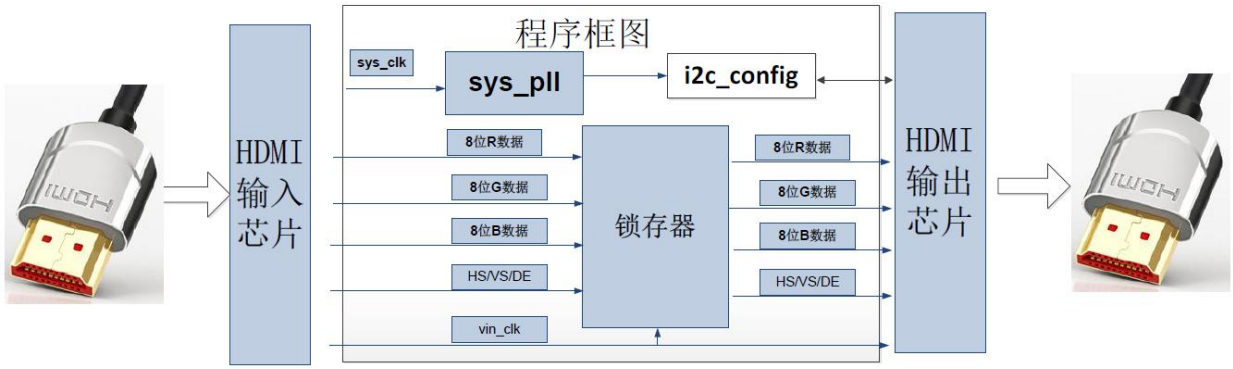
Several resolution timing parameters are preset in the program, including 2 LCD screens, which are prepared for subsequent LCD verification tests. For detailed settings, please refer to the related routines provided.

3. HDMI interface configuration module i2c_config.v

The module is configured with the register of the HDMI interface chip ADV7511, and the i2c_master_top module is called to implement the IIc interface communication for transmitting configuration parameters.

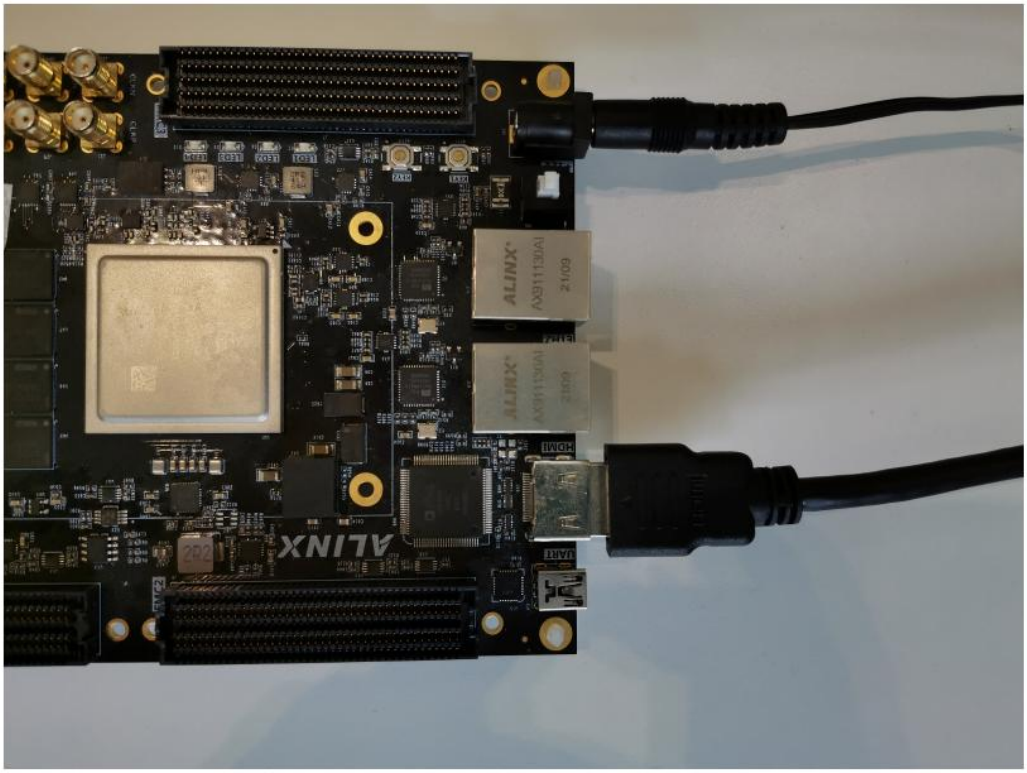
4 HDMI loop-throug design

The HDMI loop is to display the video image of the HDMI input through the FPGA loop and then output it to the display via HDMI. The program is based on the HDMI output program, the color_bar generation module (color_bar.v) is deleted, and the HDMI output signal directly uses the HDMI input signal. The logical block diagram of the implementation is as follows:



5 Experiment Result

As the HDMI output device, the FPGA development board can only be displayed through the HDMI display device. Don't try to display through the HDMI interface of the laptop, because the laptop is also an output device.



AXKU040 HDMI Display Connection

Color bar display