DESIGN OF VGA DISPLAY SYSTEM BY USING FPGA

TABLE OF CONTENTS

ABSTRACT

As a video image display interface standard, VGA interface has been widely used in the embedded system. Due to the lack of professional VGA display controller in most embedded systems, splash screen, even blank screen problems may appear while displaying the high-resolution video image. The design illustrates the implement method of VGA display controller, By using The FPGA board the design stores data that gets from the User and then reads them out in form of VGA display interface standard into D/A converter which will convert the data into analog signals to display. The whole operation above is controlled by USER by writing the code . So such a design can effectively solve the problems caused by insufficient bandwidth in the displaying, what's more, it can reduce the pressure of the CPU. The article gives the internal detailed design of VGA, which includes generation of VGA timing signal, finite state machine and logic control, but also presents analysis of practical tests rendering. This entire code is working on the SPARTAN 3E Fpga board .

CHAPTER 1
INTRODUCTION

With the development of embedded system and its popularization in each industry, the requirements of the market for display also become higher and higher, and terminal video image display has become an important part of the development of the system. In fact, VGA interface displayer is an important terminal display device of computer and various intelligent instrument Equipments.

Due to the unified standard and relatively mature technology of VGA interface, it has always been used as the connected interface between the computer and external displayer. Now most of the embedded processor only integrated the LCD controller without VGA controller, therefore, we need to covert LCD signal to VGA interface signal. For now, the most widely used program chooses an appropriative conversion chip to handle. The advantage of this scheme is that it's easy to use, but at the same time, there also exists defect. First, when we want to show a 16-bit true color image, which has resolution of 1024*768 and 60 Hz field frequency, general processor may appear screen dithering, and even temporary blank screen when displaying the picture in above mode. This attributed to the internal of general processor do not have graphic processing module, and it cannot load such huge image data's transmission. Another point is that it's not easy to upgrade and update, once we want to show different resolution pictures, it needs to design the hardware circuit again and again, and sometimes even needs to replace conversion chip, then development cycle will get longer, and in a certain extent, cost will increase, which is not conducive to upgrade of the product.

In view of the above question, this design put forward a method which combined CPLD with SRAM to realize the control of VGA display. In other words, pulsing this module between the processor and displayer so that it could store the data ,which transfer from the processor, to high speed of SRAM memory, and each frame data that VGA interface displayer received are read from SRAM according to the standard, consequently, it can completely satisfy the requirement of VGA video display standard.

This system is suitable for terminal display of all kinds of embedded systems. The advantages of the design are simple circuit structure, flexible design, high stability and transplantation convenience etc.

CHAPTER 2
LITERATURE REVIEW

2.1GENERAL:

Video Graphics Array (**VGA**) refers specifically to the display hardware first introduced with the IBM PS/2 line of computers in 1987, but through its widespread adoption has also come to mean either an analog computer display standard, the 15-pin D-subminiature VGA connector or the 640x480 resolution itself. While this resolution was superseded in the personal computer market in the 1990s, mobile devices have only caught up in the last few years.

VGA was the last graphical standard introduced by IBM that the majority of PC clone manufacturers conformed to, making it today the lowest common denominator that almost all post-1990 PC graphics hardware can be expected to implement. For example, the Microsoft Windows splash screen, in versions prior to Windows Vista, appears while the machine is still operating in VGA mode, which is the reason that this screen always appears in reduced

resolution and color depth. Windows Vista and newer versions can make use of the VESA BIOS Extension support of newer graphics hardware to show their splash screens in a higher resolution than VGA allows.

VGA was officially followed by IBM's Extended Graphics Array (XGA) standard, but it was effectively superseded by numerous slightly different extensions to VGA made by clone manufacturers that came to be known collectively as Super VGA.

CHAPTER 3
DESIGN

INTRODUCTION:

The design is a VGA display system based on the CPLD and SRAM, and it will be applied to a variety of embedded display terminals, especially digital language teaching system terminal interface display. The hardware circuit diagram of the system is shown in Figure.1. It is mainly composed of VGA controller, D/A converter and SRAM. To display images or videos, it should have a data source, and the data of system come from the application programs of computer. So it needs to program the corresponding application program, and then transmit data to the system through the USB interface. Intermediate using special USB conversion chip CH341 to convert the data format to parallel format data, and so that CPLD collect data conveniently according to the agreement. VGA display image data volume is usually huge, and the built-in ROM of CPLD is difficult to meet such a large amount of data, so there needs external RAM to store the data. Input signal of the VGA display is an

analog signal (line and field signal is digital signal), so before getting into the VGA interface the data collected from parallel port must be transferred by the D/A converter.

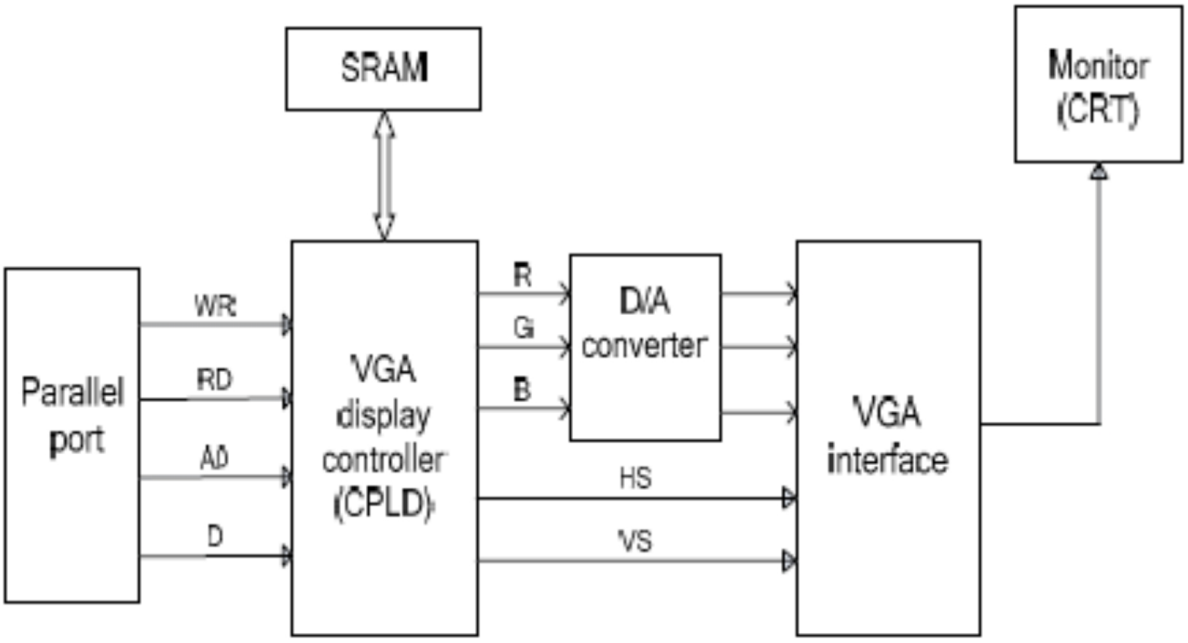
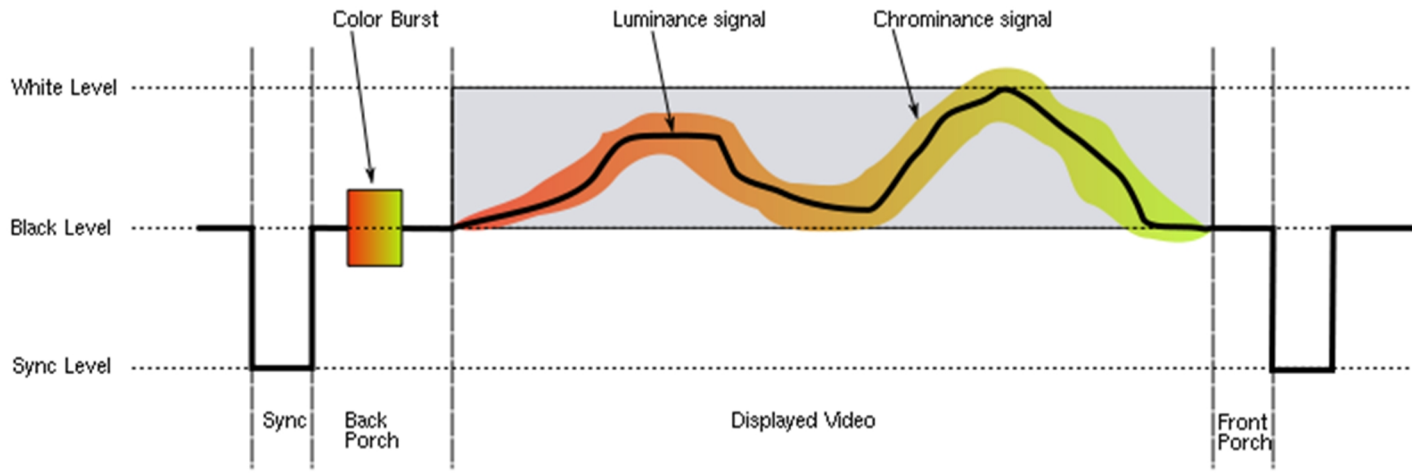


Figure.1 System Block diagram

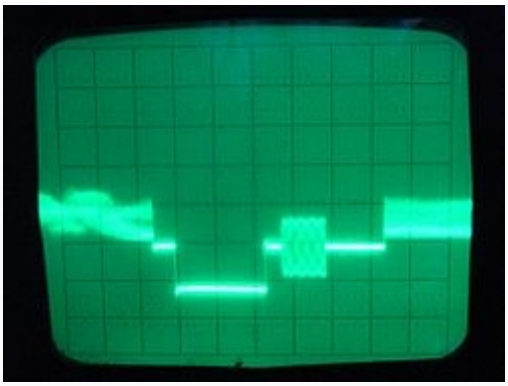
The computer sends the picture data, whose field frequency is less than 30Hz and resolution is the 800*600, is based on the 16-bit RGB model. So, through this we can verify the feasibility of the system program.

Structure of a video signal[edit]

The video carrier is demodulated to give a composite video signal; this contains luminance, chrominance and synchronization signals;[5] this is identical to the video signal format used by analog video devices such as VCRs or CCTV cameras. Note that the RF signal modulation is inverted compared to the conventional AM: the minimum video signal level corresponds to maximum carrier amplitude, and vice versa. To ensure good linearity (fidelity), consistent with affordable manufacturing costs of transmitters and receivers, the video carrier is never shut off altogether. When intercarrier sound was invented later in 1948, not completely shutting off the carrier had the side effect of allowing intercarrier sound to be economically implemented.



Each line of the displayed image is transmitted using a signal as shown above. The same basic format (with minor differences mainly related to timing and the encoding of color) is used for PAL, NTSC and SECAM television systems. A monochrome signal is identical to a color one, with the exception that the elements shown in color in the diagram (the color burst, and the chrominance signal) are not present.



Portion of a PAL video signal. From left to right: end of a video scan line, front porch, horizontal sync pulse, back porch with color burst, and beginning of next line

The *front porch* is a brief (about 1.5 microsecond) period inserted between the end of each transmitted line of picture and the leading edge of the next line sync pulse. Its purpose was to allow voltage levels to stabilise in older televisions, preventing interference between picture lines. The *front porch* is the first component of the horizontal blanking interval which also contains the horizontal sync pulse and the *back porch*.[6][7]

The *back porch* is the portion of each scan line between the end (rising edge) of the horizontal sync pulse and the start of active video. It is used to restore the black level (300 mV) reference in analog video. In signal processing terms, it compensates for the fall time and settling time following the sync pulse.[6][7]

In color television systems such as PAL and NTSC, this period also includes the colorburst signal. In the SECAM system it contains the reference subcarrier for each consecutive color difference signal in order to set the zero-color reference.

In some professional systems, particularly satellite links between locations, the audio is embedded within the back porch of the video signal, to save the cost of renting a second channel.

Synchronization[edit]

Synchronizing pulses added to the video signal at the end of every scan line and video frame ensure that the sweep oscillators in the receiver remain locked in step with the transmitted signal, so that the image can be reconstructed on the receiver screen.[6][7][8]

A *sync separator* circuit detects the sync voltage levels and sorts the pulses into horizontal and vertical sync. (see section below – Other technical information, for extra detail.)

Horizontal synchronization[edit]

The horizontal synchronization pulse (*horizontal sync HSYNC*), separates the scan lines. The horizontal sync signal is a single short pulse which indicates the start of every line. The rest of the scan line follows,

with the signal ranging from 0.3 V (black) to 1 V (white), until the next horizontal or vertical synchronization pulse.

The format of the horizontal sync pulse varies. In the 525-line NTSC system it is a 4.85 μs-long pulse at 0 V. In the 625-line PAL system the pulse is 4.7 μs synchronization pulse at 0 V . This is lower than the amplitude of any video signal (*blacker than black*) so it can be detected by the level-sensitive "sync stripper" circuit of the receiver.

Vertical synchronization[edit]

For the graphic option provided by video games, see Frame rate and Refresh rate.

Vertical synchronization (Also *vertical sync* or *VSYNC*) separates the video fields. In PAL and NTSC, the vertical sync pulse occurs within the vertical blanking interval. The vertical sync pulses are made by prolonging the length of HSYNC pulses through almost the entire length of the scan line.

The *vertical sync* signal is a series of much longer pulses, indicating the start of a new field. The sync pulses occupy the whole of line interval of a number of lines at the beginning and end of a scan; no picture information is transmitted during vertical retrace. The pulse sequence is designed to allow horizontal sync to continue during vertical retrace; it also indicates whether each field represents even or odd lines in interlaced systems (depending on whether it begins at the start of a horizontal line, or midway through).

The format of such a signal in 525-line NTSC is:

- pre-equalizing pulses (6 to start scanning odd lines, 5 to start scanning even lines)
- long-sync pulses (5 pulses)
- post-equalizing pulses (5 to start scanning odd lines, 4 to start scanning even lines)

Each pre- or post- equalizing pulse consists in half a scan line of black signal: 2 μs at 0 V, followed by 30 μs at 0.3 V.

Each long sync pulse consists in an equalizing pulse with timings inverted: 30 μs at 0 V, followed by 2 μs at 0.3 V.

In video production and computer graphics, changes to the image are often kept in step with the vertical synchronization pulse to avoid visible discontinuity of the image. Since the frame buffer of a computer graphics display imitates the dynamics of a cathode-ray display, if it is updated with a new image while the image is being transmitted to the display, the display shows a mishmash of both frames, producing a page tearing artifact partway down the image.

Vertical synchronization eliminates this by timing frame buffer fills to coincide with the vertical blanking interval, thus ensuring that only whole frames are seen on-screen. Software such as video games and computer aided design (CAD) packages often allow vertical synchronization as an option, because it delays the image update until the vertical blanking interval. This produces a small penalty in latency, because the program has to wait until the video controller has finished transmitting the image to the display before continuing.Triple buffering reduces this latency significantly.

Two timing intervals are defined – the *front porch* between the end of displayed video and the start of the sync pulse, and the *back porch* after the sync pulse and before displayed video. These and the sync pulse itself are called the *horizontal blanking* (or *retrace*) *interval* and represent the time that the electron beam in the CRT is returning to the start of the next display line.

Horizontal hold and vertical hold[edit]

The lack of precision timing components in early television receivers meant that the timebase circuits occasionally needed manual adjustment. If their free-run frequencies were too far from the actual line and field rates, the circuits would not be able to follow the incoming sync signals. Loss of horizontal synchronization usually resulted in an unwatchable picture; loss of vertical synchronization would produce an image rolling up or down the screen.

The adjustment took the form of *horizontal hold* and *vertical hold* controls, usually on the front panel along with other common controls. These adjusted the free-run frequencies of the corresponding timebase oscillators.

By the early 1980s the efficacy of the synchronization circuits, plus the inherent stability of the sets' oscillators, had been improved to the point where these controls were no longer necessary.

3.2 DESIGN OF VGA CONTROLLER

3.2.1 VGA SEQUENCE

VGA (Video Graphics Array) is a video transmission standard, which launched with PS/2 by the IBM in 1987, and it has the advantages of high resolution, fast display speed and rich color, so it is widely used in area of color display.VGA display also needs a standard timing signal, only in this way image data can be displayed accurately on the monitor. VGA interface signal mainly have five signals, they are line synchronous signal HS, field synchronous signal VS and three primary color signals of RGB [3], and line and field synchronizing signals is the main control signal. This design uses the programmable logic device CPLD to produce corresponding pulse signal according to the VGA timing standard. In the verilog program, we can use the counter and trigger combination way to generate different resolution line, field synchronous clock signal.

VGA interface sequence is shown in Figure.2. The above is the row (HSYNC) scan sequence diagram, it consists of sync pulse (ha), Back porch (hb), Display interval (hc) and Front porch (hd) four parts. The sync pulse (ha) is to tell the displayer that this line will begin to show,

while display interval (hc) is line effective display section. In the chart, the line blanking refers to not effective display space, and this range of display data is invalid. The structure of following field sequential is similar to the row, but the only different between the two is the length of time. From the above analysis, the image available display area is combined line with field effective areas, so when writing the programs we can use line and field counters to determine the marker bit of effective area of image display. In this way, you can program line, field counters to determine the image display effective area, so as to produce the read address.

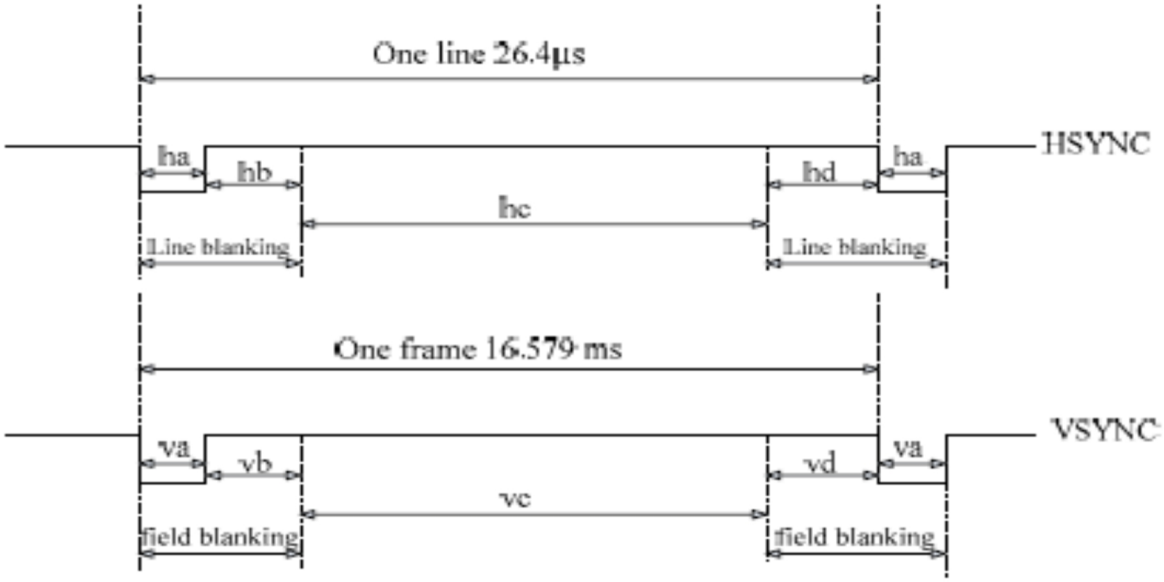


Figure.2 VGA interface timing diagram

	parameters	Time	Total time
Line	ha	128pixels/3.2μs	26.4μs
	Hb	88pixels/2.2μs	
	Hc	800pixels/20μs	
	hd	40pixels/1.0μs	
Field	Va	4lines/106μs	16.579ms
	Vb	23lines/607μs	
	Vc	600lines/15840μs	
	Vd	1lines/26μs	

TABLE .1 VGA TIMING PARAMETERS TABLE

This design uses a 16-bit true color image of 800*600resolution and 60Hz field frequency to transfer. First of all, determining the frequency of the master clock according to the refresh rate, then using the clock frequency and resolution of the image to calculate the various parameters of line and field sequential. By checking the standard, you can find pixel master clock is 40 MHz, and the system clock of this design is 80 MHZ, so that binary frequency can get the pixel clock which is needed. Then the system clock of the 80MHz can output directly to SRAM chip which asks for higher speed of reading and writing. The pixel clock period multiplied by the number of pixels can get the row total time, while the line total time multiplied by the total number of line to get field total time, the other by parity of reasoning, thereby the table 1's parameters can be come out.

VGA monitor is displayed in a progressive scanning way which is scanned from left to right, from top to bottom. When scanning each row, there will have a period of blanking, and during this period, the displayer do not do any of the display, so do not send effective data to

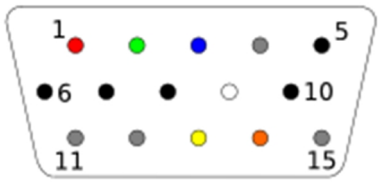
display in this meantime, otherwise the image display errors would be arise, so as to the field blanking.

Colour display

VGA relies on 3 analogue signals to define the colour of the current pixel, one each for red, green and blue. They are typically defined as between 0 and 0.7 volts although 0-1 volt often works fine. In truly analogue displays the value can vary by any arbitrarily small amount, giving no definite limit to the number of colours. Typically however the number of colours is defined by the number of bits assigned to each colour channel. The input impedance of these analogue lines is 75Ω (like composite video) this can be used in simple DAC circuits to turn ordinary logic level outputs into analogue values in the right voltage range.

The VGA display uses the horizontal sync period at the end of each line to calibrate the colour. During this period all of the colour lines need to be at their minimum value, if they're higher than the minimum then that colour will have a reduced dynamic range during the following line. Note that it isn't possible to have a negative intensity of a pixel in the screen so if the voltage on a colour line drops after the sync signal that colour will remain at its minimum intensity.

Connector



Connector Pinout:

Pin	Function
1	Red
2	Green
3	Blue
4	IDN2
5	Ground

6	Red Ground
7	Green Ground
8	Blue Ground
9	Connector signal
10	Sync Ground
11	IDN0
12	IDN1
13	H_SYNC
14	V_SYNC
15	IDN3

The image displays a minimal wiring for the VGA 15-pin high density D-type connector. This is drawn from the solder-side on the generating device. Note that each of the three colour signals have their own ground return signal, and there is a joint ground signal for the two sync lines. In most systems it is sufficient to just join all these to the system ground along with the general ground pin. If you have a proper DAC solution however you may have a separate analogue ground which may improve signal quality especially for higher resolutions. Pin 9 is typically connected to a 5V signal on the host to identify when the VGA connection is in use, this is used by a lot of monitors to switch to the appropriate input and without it you may find the monitor does not turn on.

There are also a set of return pins which can identify the monitor's capabilities (marked IDNx on the diagram). In the most basic implementations these pins are simply shorted to ground or not to indicate whether the monitor can display colour or higher resolutions. These simple qualifications are out of date now and not really useful, the pins can act as a form of synchronous serial bus similar to i^2c which can provide more detailed monitor information about make, model and capabilities. I've not experimented with these capabilities and have no idea what the practical implications of using these pins are. For a simple VGA display you can safely ignore them, just leave them disconnected.

3.2.2MEMORY MODULE

There are many choices for the external memory of CPLD, such as ROM, SRAM, SDRAM, FLASH etc. Among these, the SRAM is easier to control timing
sequence of reading and writing, what's more, the speed of reading and writing is fast. Unlike the SDRAM need complex timing to control reading and writing, and it needs to refresh in a certain period of time, as a result, it will takes up a lot of CPU resources, then after comprehensive consideration, taking the system memory
size into account, the design use SRAM as an external storage device.

The design use a SRAM as a memory, so that it can simplify system design and reduce the cost, at the same time, it also increase the difficulty of controlling the SRAM timing sequence of reading and writing. At this time, we can consider to use spare time section of SRAM bus of line sequence and frame sequence to update the data of the SRAM memory when the image display is on. The updating rate of this method is closely related to the speed of memory's data writing, faster the speed of writing data, higher the update rate is. This design uses the IS61LV51216 chip of the ISSI company as the external memory chip of the system, and its fastest theory speed of reading and writing is 10ns, which is sufficient to meet the system design requirements. In order to control easily, CS and other signals like CS could be connected directly by hardware circuit, when designing the circuit, and only leaving reading and writing signals as the control signal of SRAM.

As in the VGA display, each pixel corresponds to a position on the screen, so when displaying the image, if it cannot send corresponding data to each pixel, it will lead to inaccurate display. So it is necessary to strictly control the timing between pixel and data. In SRAM each pixel is assigned a fixed address, so that we can read the corresponding coordinate data according to the address, and then do correct display.

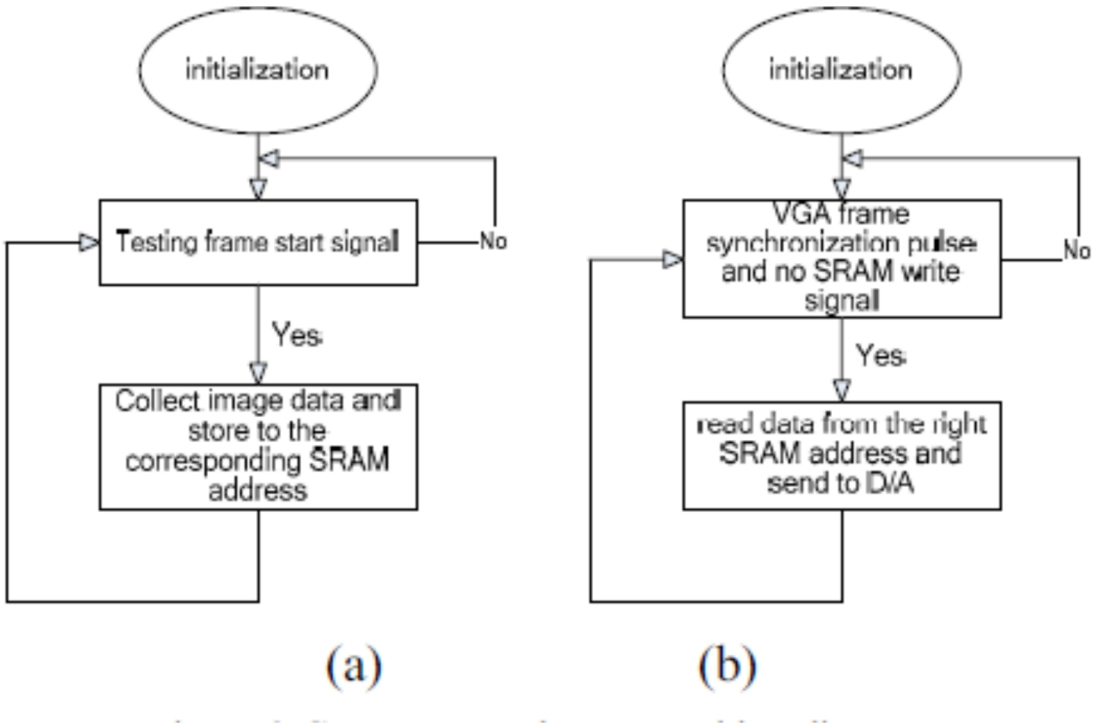


Figure.3 CPLD Internal state machine diagram

Reading and writing timing sequence of the SRAM is a key, but it also a difficulty. Because this design only use one piece of SRAM chip, so reading and writing are on the same chip address and reading and writing cannot be carried out simultaneously. This design uses the Verilog to write two state machines to achieve this control as shown in Figure 3. State machine (a) always checks in whether there is a parallel data down, and it writes the collected data as soon as it get to corresponding address of SRAM. State machine (b) is always detect frame synchronization in pulse of VGA and writing flag of SRAM, When SRAM is not writing state and VGA pulse is in the scope of the display area, it would read the corresponding data and send them to the D/A converter. State machine (a) and state machine (b) are performed at the same time.

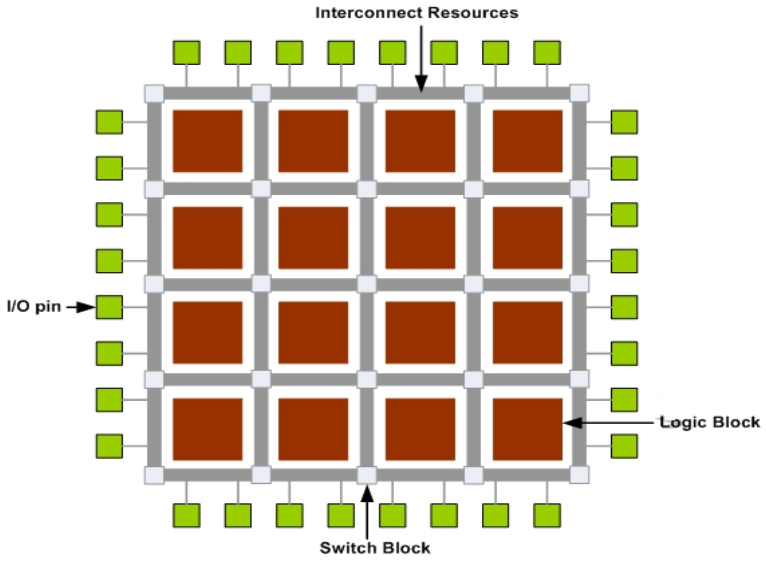
CHAPTER 4

FPGA DESIGN FLOW

4.1 FPGA DESIGN FLOW

FPGA contains a two dimensional arrays of logic blocks and interconnections between logic blocks. Both the logic blocks and interconnects are programmable. Logic blocks are programmed to implement a desired function and the interconnects are programmed using the switch boxes to connect the logic blocks. To be more clear, if we want to implement a complex design (CPU for instance), then the design is divided into small sub functions and each sub function is implemented using one logic block. Now, to get our desired design (CPU), all the sub functions implemented in logic blocks must be connected and this is done by programming the interconnects.

Internal structure of an FPGA is depicted in the following figure.



FPGAs, alternative to the custom ICs, can be used to implement an entire System On one Chip (SOC). The main advantage of FPGA is ability to reprogram. User can reprogram an FPGA to implement a design and this is done after the FPGA is manufactured. This brings the name "Field Programmable."

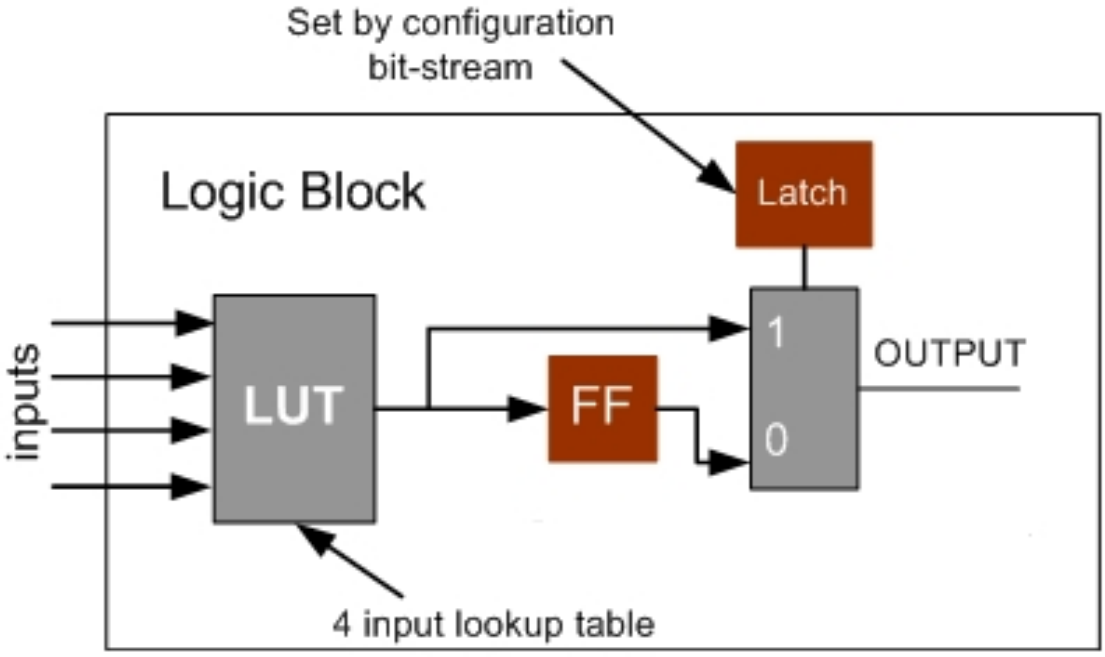
Custom ICs are expensive and takes long time to design so they are useful when produced in bulk amounts. But FPGAs are easy to implement with in a short time with the help of Computer Aided Designing (CAD) tools (because there is no physical layout process, no mask making, and no IC manufacturing).

Some disadvantages of FPGAs are, they are slow compared to custom ICs as they can't handle vary complex designs and also they draw more power.

Xilinx logic block consists of one Look Up Table (LUT) and one FlipFlop. An LUT is used to implement number of different functionality. The input lines to the logic block go into the LUT and enable it. The output of the LUT gives the result of the logic function that it implements and the output of logic block is registered or unregistered out put from the LUT.

SRAM is used to implement a LUT.A k-input logic function is implemented using 2^k * 1 size SRAM. Number of different possible functions for k input LUT is 2^2^k. Advantage of such an architecture is that it supports implementation of so many logic functions, however the disadvantage is unusually large number of memory cells required to implement such a logic block in case number of inputs is large.

Figure below shows a 4-input LUT based implementation of logic block.



LUT based design provides for better logic block utilization. A k-input LUT based logic block can be implemented in number of different ways with trade off between performance and logic density.

An n-LUT can be shown as a direct implementation of a function truth-table. Each of the latch

holds the value of the function corresponding to one input combination. For Example: 2-LUT can be used to implement 16 types of functions like AND , OR, A+not B .... etc.

A	B	AND	OR	NAND	.....	....
0	0	0	0	1		
0	1	0	1	1		
1	0	0	1	1		
1	1	1	1	0		

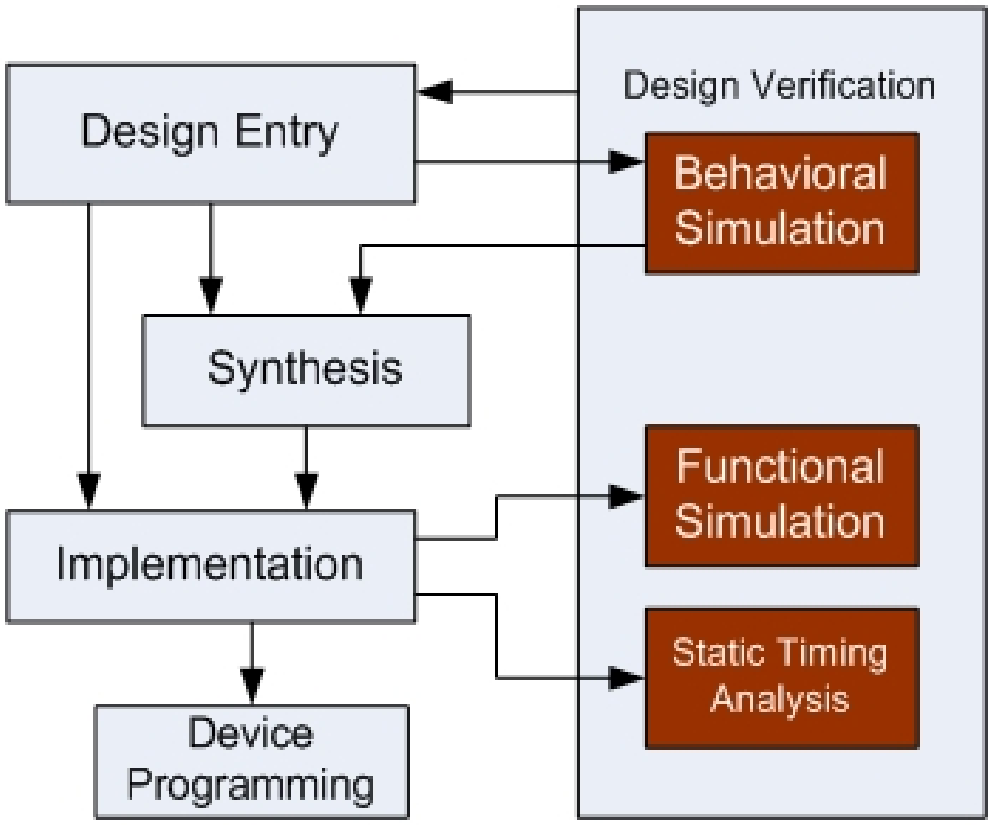
Interconnects

A wire segment can be described as two end points of an interconnect with no programmable switch between them. A sequence of one or more wire segments in an FPGA can be termed as a track.

Typically an FPGA has logic blocks, interconnects and switch blocks (Input/Output blocks). Switch blocks lie in the periphery of logic blocks and interconnect. Wire segments are connected to logic blocks through switch blocks. Depending on the required design, one logic block is connected to another and so on.

FPGA DESIGN FLOW

In this part of tutorial we are going to have a short intro on FPGA design flow. A simplified version of design flow is given in the flowing diagram.



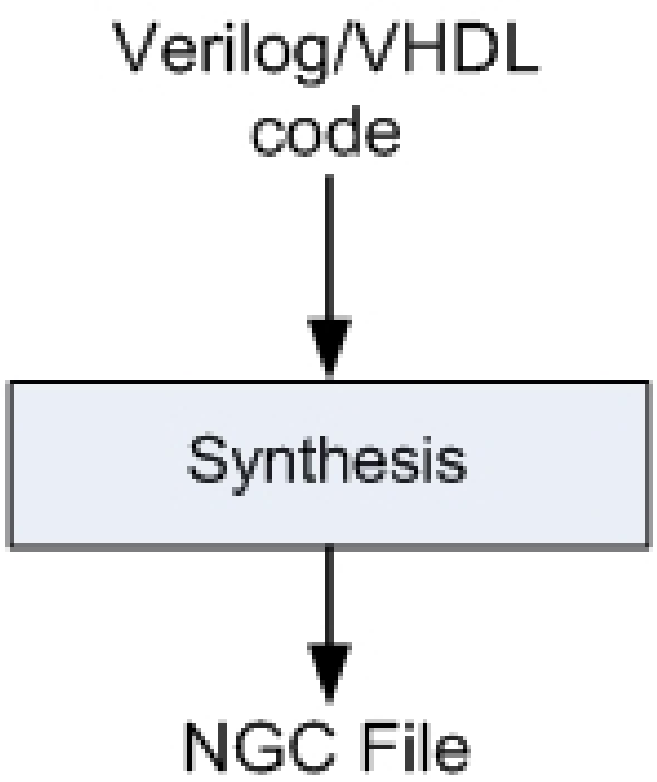
4.1.1 Design Entry

There are different techniques for design entry. Schematic based, Hardware Description Language and combination of both etc. . Selection of a method depends on the design and designer. If the designer wants to deal more with Hardware, then Schematic entry is the better choice. When the design is complex or the designer thinks the design in an algorithmic way then HDL is the better choice. Language based entry is faster but lag in performance and density.

HDLs represent a level of abstraction that can isolate the designers from the details of the hardware implementation. Schematic based entry gives designers much more visibility into the hardware. It is the better choice for those who are hardware oriented. Another method but rarely used is state-machines. It is the better choice for the designers who think the design as a series of states. But the tools for state machine entry are limited. In this documentation we are going to deal with the HDL based design entry.

4.1.2 Synthesis

The process which translates VHDL or Verilog code into a device netlist formate. i.e a complete circuit with logical elements(gates, flip flops, etc…) for the design.If the design contains more than one sub designs, ex. to implement a processor, we need a CPU as one design element and RAM as another and so on, then the synthesis process generates netlist for each design element Synthesis process will check code syntax and analyze the hierarchy of the design which ensures that the design is optimized for the design architecture, the designer has selected. The resulting netlist(s) is saved to an NGC(Native Generic Circuit) file (for Xilinx® Synthesis Technology (XST)).



4.1.3. Implementation

This process consists a sequence of three steps

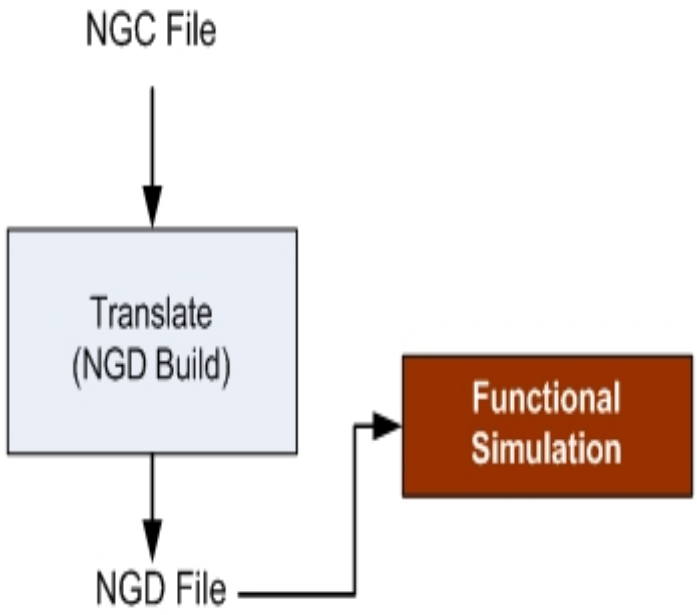
1. Translate

2. Map

3. Place and Route

4.1.3.1 Translate

This process combines all the input netlists and constraints to a logic design file. This information is saved as a NGD (Native Generic Database) file. This can be done using NGD Build program. Here, defining constraints is nothing but, assigning the ports in the design to the physical elements (ex. pins, switches, buttons etc) of the targeted device and specifying time requirements of the design. This information is stored in a file named UCF (User Constraints File). Tools used to create or modify the UCF are PACE, Constraint Editor etc.

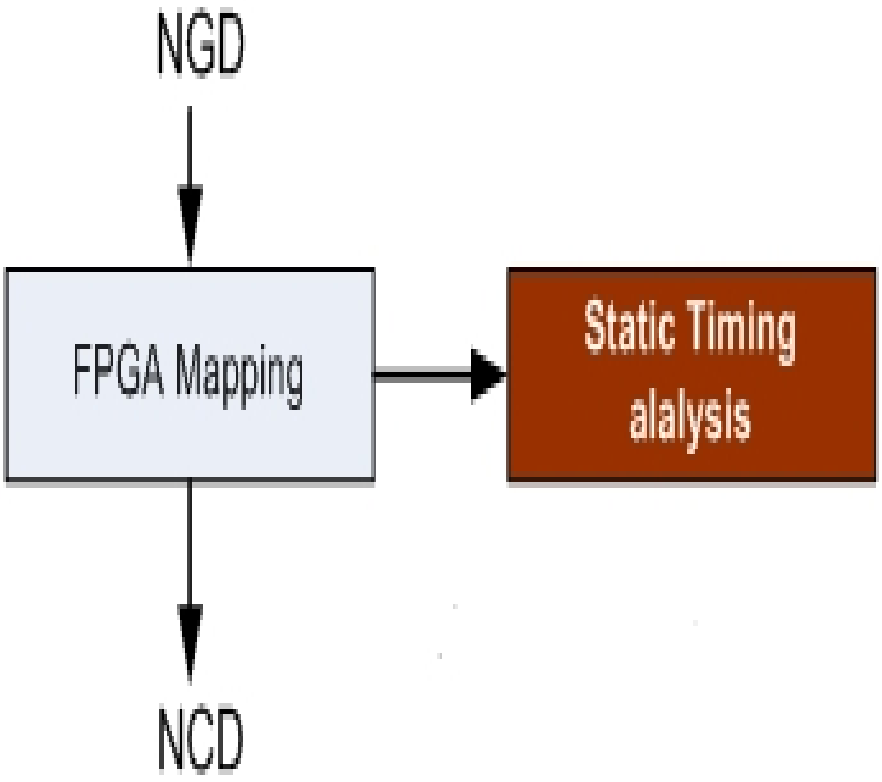


4.1.3.2 Map

This process divides the whole circuit with logical elements into sub blocks such that they can be fit into the FPGA logic blocks. That means map process fits the logic defined by the NGD file into the targeted FPGA elements (Combinational Logic Blocks (CLB), Input Output Blocks (IOB)) and generates an NCD (Native Circuit Description) file which physically represents the design mapped to the components of FPGA. MAP program is used for this purpose.

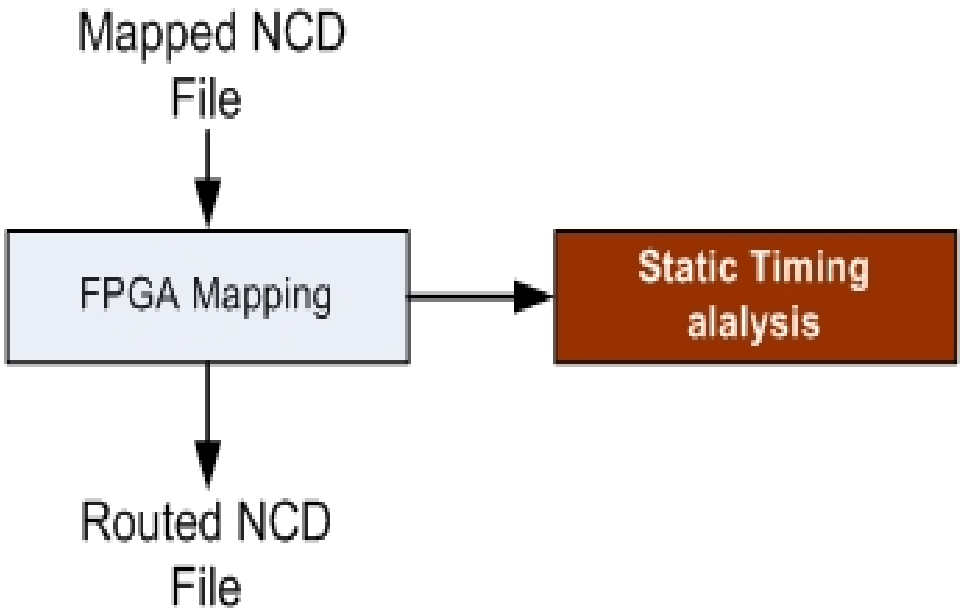
4.1.3.3 Place and Route

PAR program is used for this process. The place and route process places the sub blocks from the map process into logic blocks according to the constraints and connects the logic blocks. Ex. if a sub block is placed in a logic block which is very near to IO pin, then it may save the time but it may effect some other constraint. So trade off between all the constraints is taken account by the place and route process. The PAR tool takes the mapped NCD file as input and produces a completely routed NCD file as output. Output NCD file consists the routing information.



Place and Route PAR program is used for this process. The place and route process places the sub blocks from the map process into logic blocks according to the constraints and connects the logic blocks. Ex. if a sub block is placed in a logic block which is very near to IO pin, then it may save the time but it may effect some other constraint. So trade off between all the constraints is taken account by the place and route process. The PAR tool takes the mapped NCD

file as input and produces a completely routed NCD file as output. Output NCD file consists the routing information.



4.1.4 Device Programming

Now the design must be loaded on the FPGA. But the design must be converted to a format so that the FPGA can accept it. BITGEN program deals with the conversion. The routed NCD file is then given to the BITGEN program to generate a bit stream (a .BIT file) which can be used to configure the target FPGA device. This can be done using a cable. Selection of cable depends on the design.

4.1.5 Design Verification

Verification can be done at different stages of the process steps.

4.1.6 Behavioral Simulation (RTL Simulation) This is first of all simulation steps; those are encountered throughout the hierarchy of the design flow. This simulation is performed before synthesis process to verify RTL (behavioral) code and to confirm that the design is functioning as intended. Behavioral simulation can be performed on either VHDL or Verilog designs. In this

process, signals and variables are observed, procedures and functions are traced and breakpoints are set. This is a very fast simulation and so allows the designer to change the HDL code if the required functionality is not met with in a short time period. Since the design is not yet synthesized to gate level, timing and resource usage properties are still unknown.

4.1.7 Functional simulation (Post Translate Simulation) Functional simulation gives information about the logic operation of the circuit. Designer can verify the functionality of the design using this process after the Translate process. If the functionality is not as expected, then the designer has to made changes in the code and again follow the design flow steps.

4.1.8. Static Timing Analysis This can be done after MAP or PAR processes Post MAP timing report lists signal path delays of the design derived from the design logic. Post Place and Route timing report incorporates timing delay information to provide a comprehensive timing summary of the design.

CHAPTER 7

CONCLUSION

The experiments show that the Color stripes is correctly displayed on the monitor, and it has good stability, good visual effects, and the naked eye can not feel that the image is refreshed.. Using this scheme can reduce the pressure of embedded system of CPU, what's more, it effectively solve the original conversion system monitors problems of jitter, flash screen, and even blank screen problems. This system also applies to the microcontroller to control the display, while interface can also use the parallel port. The FPGA can use Spartan 3E XC3S 500E to control VGA port to display. This design has certain practicability and value of popularization.