International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 4, April - 2013

FPGA Based Implementation of Flat Panel Display Controller with DVI Interface

Divya K S

4th SEM M.Tech in VLSI
KVGCE Sullia, D K

***Abstract*—Digital displays are a fast-growing market comprising LCD, plasma, and rear projection television technologies as well as smaller displays for mobile handsets and automobiles, in addition to many other applications. Digital image processing enhances the overall viewing aesthetics of the displayed image and can differentiate our product. This paper deals about the design and development of Flat panel display controller design using Xilinx Spartan-3AN Development board and intended for display panel applications to assist in developing products for this market. The display solution FPGA consists of a DVI Input interface, colour temperature correction, precise gamma correction, an image dithering engine, and Low-Voltage Differential Signaling (LVDS) Transmit (TX) output interface.**

I. INTRODUCTION

In this project work, the objective is to design and validate a LCD flat panel display controller on a Xilinx Spartan-3 FPGA. The LCD Flat panel display controller would consist of DVI Receiver, which accepts input video signals in RGB format, Colour Temperature Correction (CTC) module which does the correction of the input video frames into user set colour temperature, Precise Gamma Correction which corrects the pixels to the required Gamma value, Image Dithering Engine (IDE) which does the Dithering of the Colour temperature corrected pixels to a dithered 3x8 pixel stream, and an output display interface consisting of DVI interface to drive the LCD module.

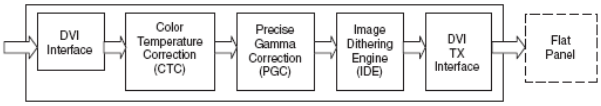

Fig. 1. Display Panel Design Flow

TABLE I
BLOCK DESCRIPTIONS

Block Name	Block Description
DVI Receiver	Accepts input video in DVI format
Color Temperature Correction (CTC)	Incoming RGB values of the entire frame are corrected to the user set color temperature
Precise Gamma Correction (PGC)	Accepts the pixel stream, processes it as per the required gamma value, and sends the modified pixel stream out to the next processing module
Image Dithering Engine (IDE)	Receives the 3x10-bit gamma corrected pixel stream from the precise gamma correction module and dithers it to 3x8-bit pixel stream, without losing the video quality
Display Interface	Allows the DVI transmitter to directly drive LCD modules with LVDS interfaces

A. Introduction to Colour Temperature

White light can be described by colour temperature. To determine the colour temperature of a light source, its output is compared with a theoretical "black-body radiator" at a certain temperature in Kelvin. Specifically, 5000K to 5500K is seen in typical daylight, 2000K is red/orange, and 15000K is bluish. Different light sources and different display technologies show differing colour temperatures. For example, as the sun crosses the sky, it may appear to be red, orange, white, or blue, depending on its position. In digital displays, white colour is realized by a superposition of the R, G, and B colour emitted from the R, G, and B cells of the specific display.

RGB data can be transformed to fit the CIE x-y colour space, the colour space where many calculations are performed in this algorithm. The term "White Point" is loosely defined as colour temperature. On a CIE chromaticity diagram (Figure 2), a white point at 5500K is near the point x = y = 0.33. For example, plasma display panels (PDPs) show a low colour temperature, especially compared to a conventional cathode ray tube (CRT) display.

The low colour temperature of the PDP is caused by an inherent low blue luminance. Furthermore, the end customer cannot arbitrarily vary the colour temperature once the PDP cell structure and the related driving scheme are fixed.

A digital display output such as a PDP must be colour corrected to achieve better image quality.

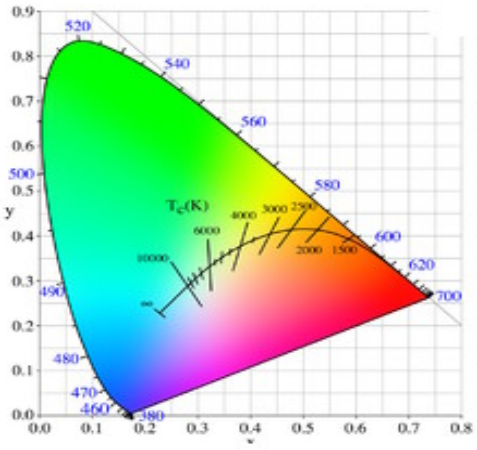

Fig. 2. Colour Temperature on CIE Chromaticity Diagram

B. CTC Algorithm

In this algorithm implementation, the white point or colour temperature of the frame is changed to the desired colour temperature of that particular frame. First, a CIE reference white point is selected in x and y format from a user-selected temperature input. For every incoming frame, a white point is estimated.

The input frame of the 8- or 10-bit RGB data is converted into x-y format using a colour-mixing program. Next, the incoming frame temperature and the reference temperature are checked to see if they are the same.

A recursive calculation controls the delta temperature value and selects the reference white point to give an exact colour temperature correction. If the colour temperature is the same (within the delta), processing is not done. If the colour temperature is different, the frame is changed to the desired colour temperature. A ratio method is used to modify the colour temperature. The CIE temperature to incoming frame temperature ratio is then applied to individual RGB data of the input frame to set the desired colour temperature. At this point, correlated colour temperature conversion is then achieved.

The value of the input colour temperature register indicates a standard temperature value (6500K, 8000K, or 9300K) as the target colour temperature. The values are selected by an external DIP switch. Then CIE standard white point values are determined. These respective RGB white point values are used for further operation. By default, colour temperature is not changed; it will bypass the input data. After selecting the color temperature at the start of the next frame, reference white point values are updated in the colour correction module.

C. Frame White Point Estimation

To estimate the white point, the following algorithm is used. First, the pixel with the maximum sum of RGB over the entire frame is located. Related RGB values that represent this sum are considered as white points for that particular frame. Then this white point is updated at the start of the next incoming frame.

D. Design of Colour Temperature Correction Module

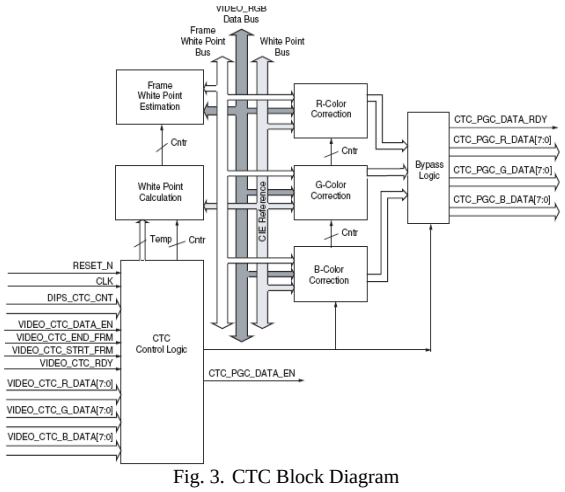

Fig. 3. CTC Block Diagram

In the CTC module, the incoming RGB values of the entire frame are corrected to the new temperature. For correction of these values, the ratio multiplication method is used. The curves show the gamma-corrected pixel input. For example, for the R-value of a pixel the correction in Equation 1 is used.

$$R_{in} \times \frac{R_{wt}}{R_{wf}} = R_{out} \qquad \text{Equation 1}$$

Where:
Rin = Present R value of input pixel
Rwt = CIE white point value for a standard temperature
Rwf = R value white point of the previous frame
Rout = R value of CTC corrected output pixel

The ratio Rwt / Rwf remains constant over the entire frame interval. The same procedure is also applied to the G and B values. At every clock cycle, pixels are simultaneously manipulated. CTC_PGC_DATA_RDY flag is enabled when the colour temperature correction module is ready to send data to the next module, the precise gamma correction block.

E. Precise Gamma Correction

A gamma characteristic is an exponential relationship that approximates the relationship between the encoded luminance in a display system and the desired image brightness. Mathematically, a generic function is: Output = Input Function ^ (Gamma), or the inverse function to pre-correct the data before it is displayed.

Many displays show a nonlinear relationship between input and brightness output; hence, the need for gamma correction. Correction can also be applied to enhance perceived image quality. Uncorrected images or the incorrect gamma can cause poor contrast, poor colour balance, and an improper overall light level. In addition, it is difficult to correct these image deficiencies with other colour adjustments. Therefore, it is important to first encode the proper gamma for all images. For example, in the case in Figure 4, the linear encoded (uncorrected) input signal Vs shows a large jump in perceived brightness from 0.1 to 0.4 and a much smaller increase from 8 to 10.

Basically, the gamma function is applied to the input to achieve linear output intensity for each input step for this display, as is desired with output I.

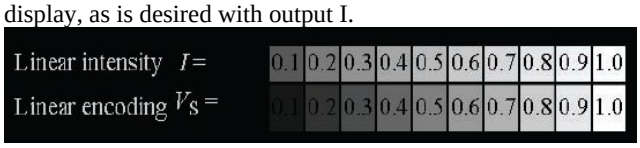

Fig. 4. Uncorrected (Vs) and a Gamma-Corrected Intensity (I)

F. Gamma Correction Implementation

A gamma curve of a 10-bit width was chosen to increase the accuracy of the output because the 10-bit output can be approximated to the nearest integer with more degrees of freedom. The 8-bit and 10-bit gamma curves are shown in Figure 5 and Figure 6, respectively, for comparison.

Luminance value with respect to the gray curve has the 8- or 10-bit resolution and the black curve is the desired response.

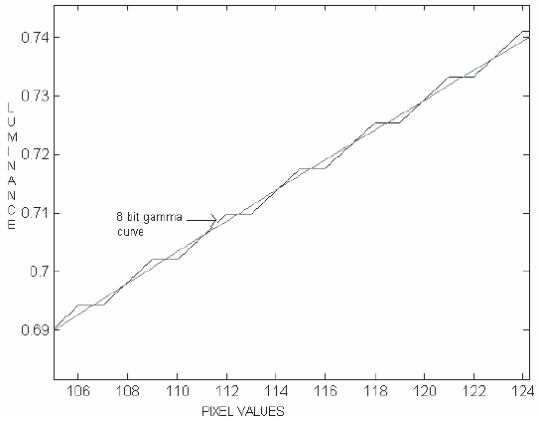

Fig. 5. 8-bit Gamma Curve versus a Desired Linear Response

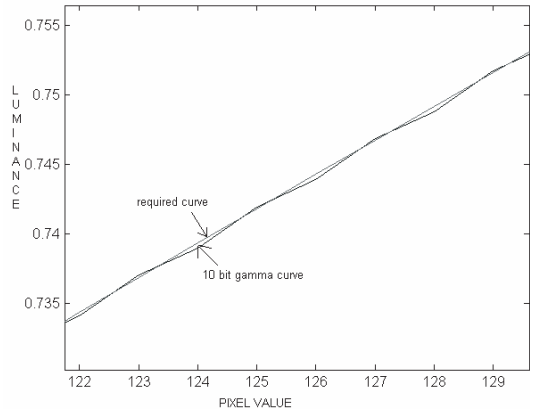

Fig. 6. 8-bit Gamma Curve versus a Desired Linear Response 10-bit Gamma Curve versus a Desired Linear Response

The figures clearly show that the 8-bit output produces a step-like function and the 10-bit output produces the desired smoothness and fit to the output. Equation 2 is used for 10-bit gamma correction.

$$Y = ROUND\left[1023 \times \left(\frac{X}{256}\right) \wedge \left(\frac{1}{Gamma}\right)\right] \qquad \text{Equation 2}$$

Where:
X = an RGB input (integer representation an individual 8-bit R, G, or B value)
Y = the 10-bit gamma corrected R', G', B' output
Gamma = the gamma factor (programmable) Round (Y) takes the nearest integer if the value of the decimal value is greater than 0.5; otherwise, it truncates the decimal part. The output is in a 10-bit format.

G. Design of PGC Module

External DIP switches are set to the value of gamma to be applied to the input data. The gamma functions are applied individually to each R, G, B colour through a series of look-up tables (LUTs). The output data is in the significant bits (LSBs) of the three R, G, B colours contain form of 3x10-bit data for R'G'B'. When precise gamma correction is complete, data-ready flags are enabled for the next module: the image dithering engine. It is also possible to bypass this block via an external DIP switch.

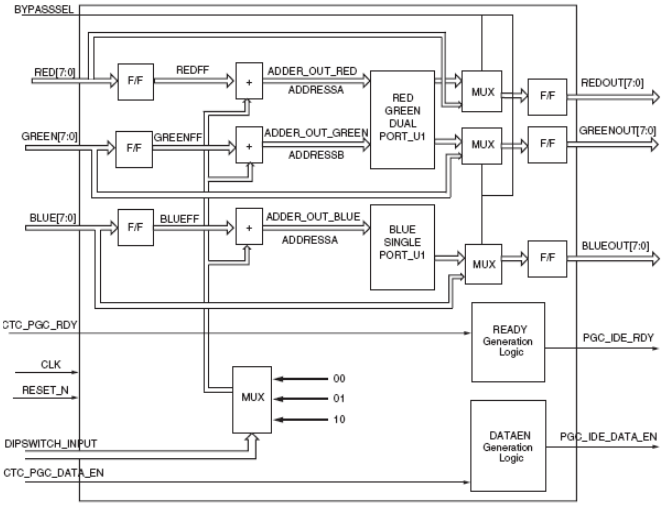

Fig. 7. Block Diagram of Precise Gamma Correction

H. Image Dithering Engine (IDE):

Dithering is a technique used to create the illusion of colour depth in displays with limited colour depth. In a dithered image, colours that are not available are approximated by a mix of coloured pixels from within the colours that are available. The human eye perceives the mixture as a different colour. For example, a display with only black or white colours can be used to create an image with gray colours by use of dithering (see Figure 8). The interlaced black and white pixels create the illusion of gray.

Fig. 8. Creating New Colours by Use of Dithering

Some display devices have colour depth less than the colour depth of the input data (for example, eight-bit input data and six-bit display colour depth). The input data is either truncated or rounded, but this approach usually produces both a loss of detail and may produce large, banded areas of a single colour that differs significantly from the original image. Dithering is used to enhance these images.

I. Design of Image Dithering Engine Module

The IDE module receives 30-bit (10 bits of data x3 for R'G'B') pixel streams from the PGC module. The image dithering engine operates only on the active pixels. It uses a spatial dithering technique with a 2x2 dithering matrix for 10- to 8-bit dithering. When a 3x10 bit data stream comes in, for

example, the two least the most “fine” colour information and are selected away from the eight remaining MSB bits.

The two LSB truncated bits effectively are an “error.” These two bits can have any value from 00 to 11 (binary) providing four finer colour levels (0%, 25%, 50%, and 75%) to the remaining eight bits. This error is spread over adjacent pixels.

The following example shows the weights in the 2x2 matrix:

0	1
2	3

The following example shows how the above weight table is spread across the display space.

0	1	0	1	…..
2	3	2	3	…..
0	1	0	1	…..
2	3	2	3	…..
.	.	.	.	…..

Each weight is actually a threshold value. If the last two-bit value of the input pixel is greater than the entry in the table for that position, the energy level represented by those two LSBs is added to the remaining bits, while making sure that the resultant eight-bit number does not overflow.

To eliminate a noticeable pattern, the weight table can be rotated every two lines or every frame, which adds a temporal dithering over the spatial dither. Also, the matrix weight can be randomly rotated to provide randomness over the dithering behaviour, which is good for a video screen display. When dithering with the least significant two bits of input data, the IDE uses spatial, tempo-spatial, and random spatial dithering with 2x2 pixel blocks. When complete, the 24-bit dithered data (eight bits for each R, G, and B) are sent to the LVDS/DVI TX interface along with a dither_data_ready signal.

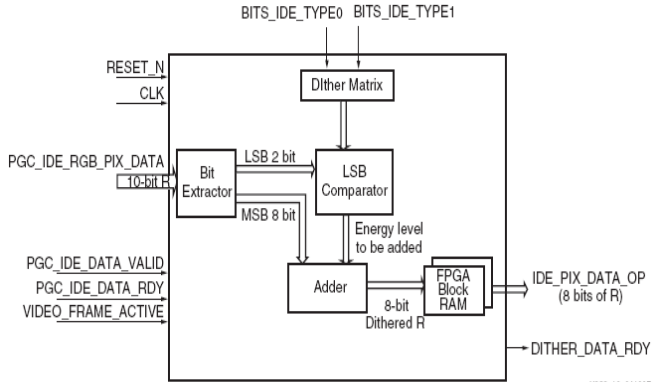

Fig. 9. Functional Description of Image Dithering Engine

II. SIMULATION RESULTS

VHDL has been chosen as the HDL language for implementation onto FPGA. Modelsim Simulator from Mentor Graphics has been used for the simulation of the developed codes.

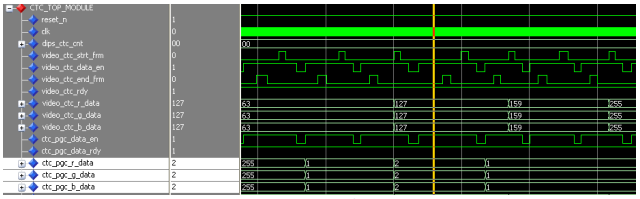

Fig. 10. Simulation Results of CTC Module

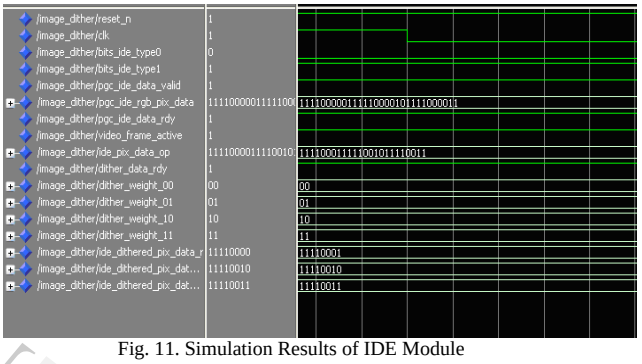

Fig. 11. Simulation Results of IDE Module

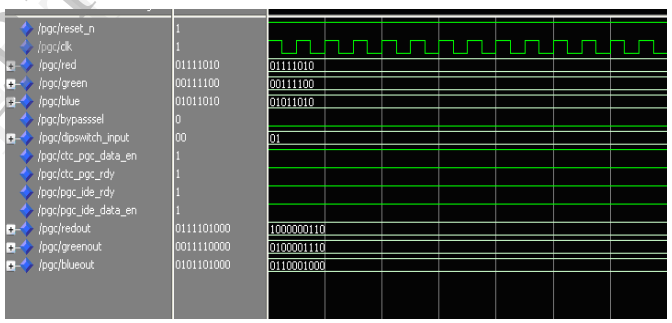

Fig. 12. Simulation Results of PGC Module

III. IMPLEMENTATION

At present the CTC module has been implemented on the FPGA and sample pixel data from a JPEG image has been converted from MATLAB and the textual form of Pixel information is stored in External ROM present on FPGA Board. The input pixels are read from the ROM and processed inside the FPGA.

The outputs are extracted from FPGA (By using Chipscope Pro tools) and the outputs are written to a text file. The outputs are again imported into MATLAB and converted to JPEG image to observe the differences between input image and output images processed by the IDE module.

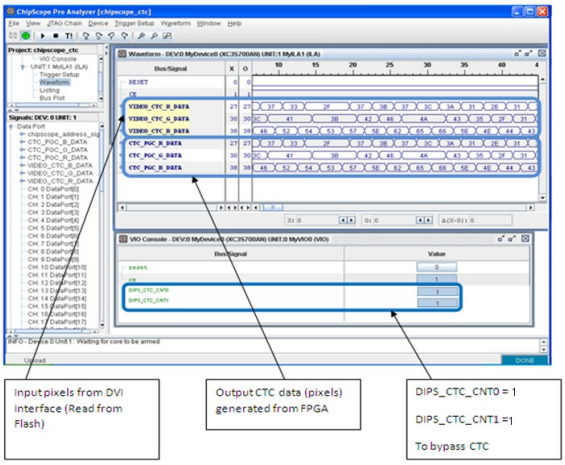

Fig. 13. Chipscope Results for the IDE module depicting the signal values on Inputs and Outputs

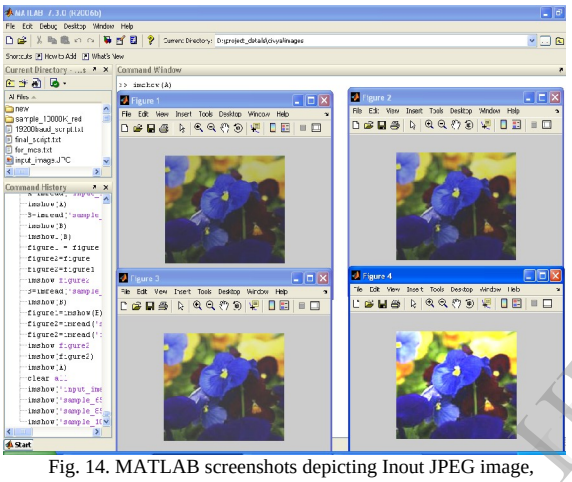

Fig. 14. MATLAB screenshots depicting Inout JPEG image, CTC module outputs at 6500K, CTC module outputs at 8500K and CTC module outputs at 10000K

IV. Conclusion

The design presented here presents a series of video processing algorithms on a Xilinx powered development board to quickly start a display development program. Any of the individual blocks can be customized to your needs on any target Xilinx device. In addition, new video processing blocks can be added to this system to quickly validate your display video enhancement algorithms.

Acknowledgment

I would like to thank my internal guide, Mr. Sai Venkatramana Prasada G S, Asst professor, Department of Electronics and Communication Engineering, KVGCE Sullia for the enormous help being delivered to me at all points of the design cycle.

I would also take any opportunity to thank all the authors mentioned in the reference papers, without which there would have been no progress.

References

[1] IEEE: **Color Temperature Conversion for Video on TV or PC Reflecting Human's Display Preference Tendency;** Sang-Kyun Kim; Du-Sik Park; Won-Hee Choi; Seong-Deok Lee; Convergence Information Technology, 2007. International Conference on 21-23 Nov. 2007 Page(s):861 – 867 Digital Object Identifier 10.1109/ICCIT.2007.12

[2] IEEE: Design Considerations Between Color Gamut and Brightness for Multi-Primary Color Displays; Mang Ou-Yang; Shih-Wei Huang; Display Technology, Journal of Volume 3, Issue 1, March 2007; Page(s):71-82; Digital Object Identifier 10.1109/ JDT.2006.890701.

[3] IEEE: **Gamma correction with revised piecewise curve and edge directed error diffusion;** Mingyu Liu; Yong Ding; Xiang Wang; Xiaolang Yan; Wireless Communications & Signal Processing, 2009. WCSP 2009. International Conference on 13-15 Nov. 2009; Page(s):1-4; Digital Object Identifier 10.1109/ .2009.5371641.

[4] IEEE: **Low-voltage low-power LVDS drivers;** Mingdeng Chen; Silva-Martinez, J.; Nix, M.; Robinson, M.E.; Solid-State Circuits, IEEE Journal of Volume 40, Issue 2, Feb. 2005 Page(s):472 - 479 Digital Object Identifier 10.1109/JSSC.2004.840955

[5] Spartan 3AN Data Sheet: www.xilinx.com

[6] Modelsim User Guide: www.xilinx.com

[7] Xilinx ISE User Guide: www.xilinx.com

[8] Chipscope Pro User Guide: www.xilinx.com