TFT-LCD Timing Control Based on FPGA

Chang-Jiu Chen and Bing-Chia Peng
Department of Computer Science, National Chiao Tung University
cjchen@cs.nctu.edu.tw

ABSTRACT

Due to that different panel module implementations have different characteristics, panel module will affect designs on timing control signals. If we want to implement a new panel module we have to design new timing controller.

In this paper, we use Verilog to implement flexible TFT-LCD timing controller. We use EEPROM memory to save associated control timing values, and we can reuse timing controller for different panel modules and reduce design cycle of panel system. Finally, we use XILINX FPGA to verify control timing and use TFT-LCD panel to verify frame rate control algorithm.

1: INTRODUCTIONS

In the past, the timing controller of the TFT-LCD is used for specific panel process. Therefore TFT-LCD system designers have to create many specific for timing control in different panel process, and it needs more efforts on system design cycle.

If we can design flexible timing controller for different panel model, it will be convenient for panel system engineer and will have shorter develop cycles to achieve time to market.

Furthermore, in early, TFT-LCD timing controller is using TTL level to transmit image stream for column driver, but it usually has EMI emissions issue and large wire trace on print circuit board. Therefore by reducing image data stream, it can reduce large wire trace and EMI emissions issue. Thus normally we use 6-bits image data format for display, and in that case, we can know that reducing image data format will affect image quality. In this paper we improve these issues.

In this paper we describe TFT-LCD panel system in section 2. We elaborate the design of TFT-LCD timing controller and frame rate control algorithm in section 3. In section 4 we use FPGA-based platform to verify our timing controller design and use TFT-LCD panel to check frame rate control algorithm. Finally in section 5 we illustrate conclusions of TFT-LCD panel timing controller.

2: TFT-LCD Panel System

In flat panel display application, we can divide the luminance to non-emission and emission group. See Figure 1. In this paper, we focus on active matrix driving methods in TFT-LCD non-emission device. There are two types of material used in TFT-LCD manufacturing. One is a-Si and the other is Poly-Si. In a-Si, there are some advantages like cheap and reliable technology for very large IC area but it has very slow electron mobility and sensitive to head and light, and it also reduces brightness. The advantages of Poly-Si material are higher speed, inherent stability, brighter and higher resolution. On the other hand, it's expensive due to the higher processing temperature and larger OFF state current leakage.

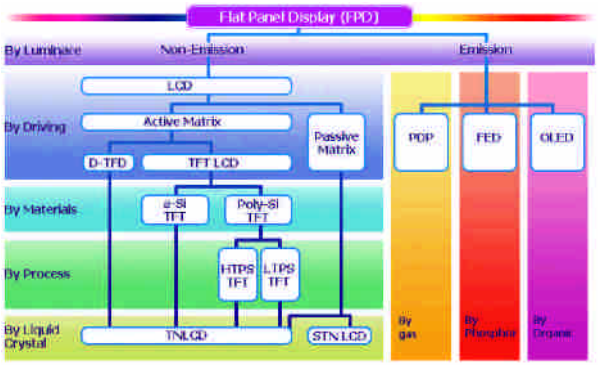

Figure 1. Classification of FPD.

The TFT substrate consists of a TFT array and a gray of external terminals in which column driver and row driver are bonded to drive the TFT-LCD panel. These drivers are directly bonded to the glass with TCP (Tape Carrier Package) connectors, and they provide each pixel of the panel with video signals that are transferred to the panel via a video signal processor and timing controller. A schematic diagram of TFT-LCD module and controllers is shown in Figure 2.

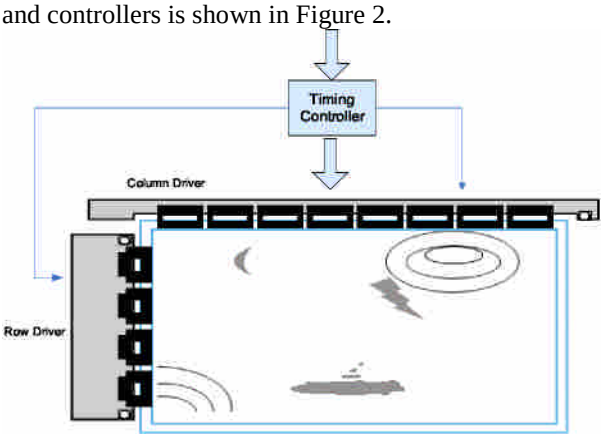

Figure 2. System block of TFT-LCD panel. [2]

In Figure 3 it shows that the brightness of the display module is much lower than that of the backlight illumination.

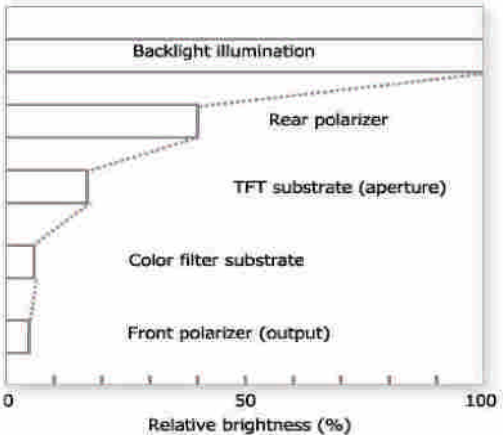

Figure 3. The brightness of the TFT-LCD module.

A TFT is formed at each intersection of these bus lines to turn on/off the voltage applied to the LCD cell. This cell is represented by an equivalent capacitance (C_{LC}) and in parallel this capacitor formed a storage capacitor (C_S). See Figure 4.

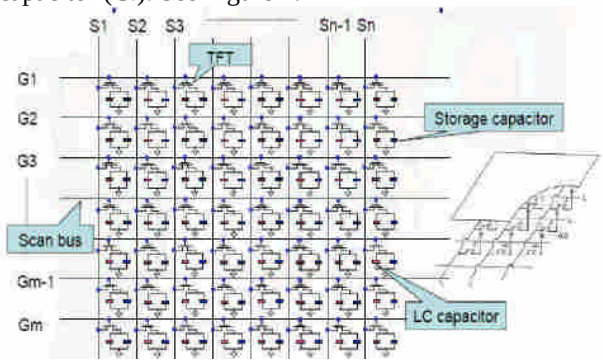

Figure 4. Schematic diagram of a TFT-LCD [5].

The color filter is formed in a striped R, G and B configuration. One pixel is formed by three dots. The display operates one line at a time, and then video signals are fed to the data bus lines simultaneously through a data buffer during the gate turn on time. The scan gate voltage pulse applied to a certain gate bus line opens the gates of the TFT connected to this bus line. The signal voltage is then applied to the pixel electrode of each dot on this gate bus line. The timing chart [1] for scanning the TFT-LCD is shown in Figure 5.

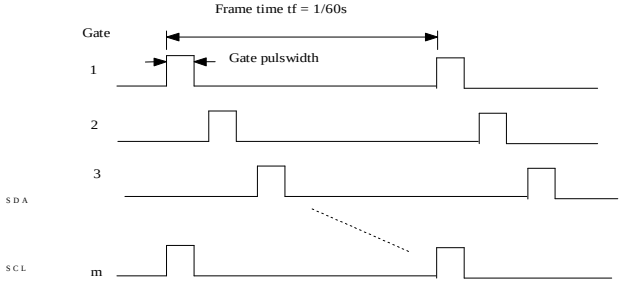

Figure 5. Timing chart of the 1024x768 resolution TFT-LCD panel.

3: TFT-LCD Control Timing Design

TCONs (Timing Controllers) are a key element in the "make-up" of LCD modules. They are essentially the "brains", and control center to the heart of a TFT LCD module [6]. In the past and in the current generation, TCONs have been implemented through the use of fully custom ASIC devices. These custom ASICs can rarely be re-used in other LCD panel designs. Along with TCON, discrete LVDS devices are also required for interface and control of the LCD module. A new family of highly functional (programmable) and highly integrated TCON have now been developed which can be used on multiple LCD panel designs.

They can support non-standard resolutions and different LCD panel configurations from various LCD manufacturers. These TCONs offer higher levels of integration resulting in smaller PCBs outlines and lower power consumption. This higher level of integration has been achieved by integrating an on chip LVDS Receiver. In addition to the higher levels of integration, a new high-speed low-voltage differential interface between the TCONs and column driver has also been developed. This new development, RSDS (Reduced Swing Differential Signaling), offers higher data transfer rates and enables higher display resolutions at lower EMI.

By providing a flexible and integrated TCON platform, LCD manufacturers can use the same device on multiple LCD Panel designs, resulting in shorter develop cycles and shorter time to market cycles.

In this paper, we do not implementation LVDS and RSDS analog interface in our timing controller. We focus on digital section. In digital circuit, we divided it into five modules in our implementation. See Figure 6.

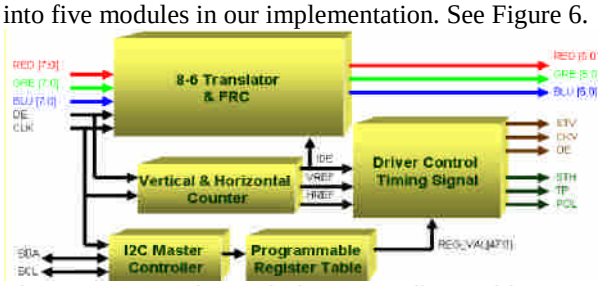

Figure 6. FPGA base timing controller architecture.

3.1: I2C Protocol and Master Control Module

The I2C bus uses a serial data line (SDA) and a serial clock line (SCL) for data transfer. All devices connected to these two signals must have open drain or open collector outputs. The logic AND function is exercised on both lines with external pull-up resistors [3].

Data is transferred between a Master and a Slave synchronously to SCL on the SDA line and a byte-by-byte basis. Each data byte is 8-bit long. There is one SCL clock pulse for each data bit with the MSB

being transmitted first. An acknowledge bit follows each transferred byte.

Within the procedure of the I2C bus, unique situation arise which is defined as START (S) and STOP (P) conditions. In START condition a high to low transition on the SDA line while SCL is high. In STOP condition a low to high transition on the SDA line while SCL is high. See Figure 7.

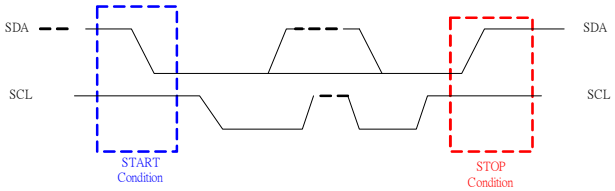

Figure 7. I2C START and STOP conditions.

SDA bus communication is established and 8-bit bytes are exchanged, each one being acknowledged by using a 9th data bit generated by the receiving party until the data transfer is completed. The bus is made free for use by other ICs when the 'master' releases the SDA line during a time when SCL is high. Apart from the two special exceptions of start and stop, no device is allowed to change the state of the SDA bus line unless the SCL line is low. If two masters try to start a communication at the same time, arbitration is performed to determine a "winner" (the master that keeps control of the bus and continue the transmission) and a "loser" (the master that must abort its transmission). The two masters can even generate a few cycles of the clock and data that 'match', but eventually one will output a 'low' when the other tries for a 'high'. The 'low' wins, so the 'loser' device withdraws and waits until the bus is freed again. There is no minimum clock speed; in fact any device that has problems to 'keep up the pace' is allowed to 'complain' by holding the clock line low. Because the device generating the clock is also monitoring the voltage on the SCL bus, it immediately 'knows' there is a problem and has to wait until the device releases the SCL line.

I2C address scheme is shown Figure 8. Any I2C device can be attached to the common I2C bus, they talk with each other, and passing information back and forth. Each device has a unique 7-bit or 10-bit I2C address. For 7-bit devices, typically the first four bits are fixed and the next three bits are set by hardware address pins (A0, A1, and A2) that allow the user to modify the I2C address allowing up to eight of the same devices to operate on the I2C bus. These pins are held highly to VCC, sometimes through a resistor or held low to GND. The last bit of the initial byte indicates that if the master is going to send (write) or receive (read) data from the slave, each transmission sequence must begin with the start condition and end with the stop condition. On the 8th clock pulse, SDA is set 'high' if data is going to be read from the other device, or 'low' if data is going to be sent (write). During its 9th clock, the master releases SDA line to accomplish the Acknowledge phase. If the other device is connected to the bus and has decoded and recognized its 'address', it will acknowledge by pulling the SDA line low. The responding chip is called the bus 'slave'. See Figure 9.

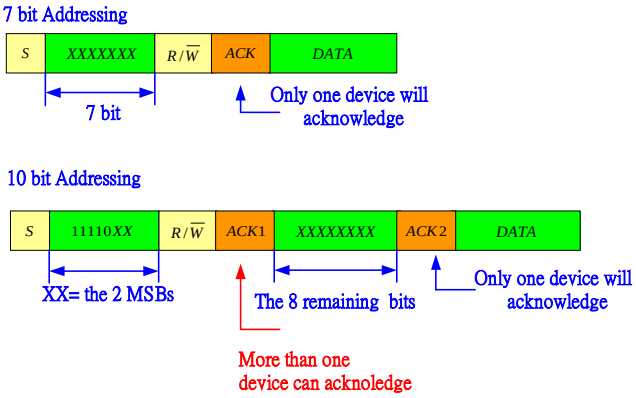

Figure 8. I2C address scheme.

3.2: Column and Row Driver Timing Control Signal

In driver timing control, we use 10-bit binary counter to count vertical and horizontal value for line and pixel reference. For example, if resolution is SXGA (1280x1024) it means 1280 pixels per one line period and 1024 lines per one frame. Therefore, we can follow this rule to build internal DE and associate driver control timing. There are six control signals for column driver and row driver. There are STH, TP and POL control signals in column driver. STH is start plus for synchronization first column driver, and TP is transfer signal and it informs column driver to latch the data register contents with leading edge. TP also transfers the data to D/A converter and outputs the gradation voltage with the trailing edge. Due to bus-line delay in different panel process, TP has different timing. Therefore we use 16 bits binary counter to control TP signal. POL is polarity reversal signal for pixel inversion control in column driver. When POL is low, the odd number of higher gamma voltage in column driver is output and even the numbers of lower gamma voltage outputs are reference power supplies. On the other hand, when POL is high, the odd-number of lower gamma voltages in column driver are output and even-number of higher gamma voltage outputs are reference power supplies.

There are three control signals in row driver, which are STV, CPV and OE. STV is start plus for synchronization first row driver and output at the falling of CKV. CKV is vertical shift clock plus for row driver and OE signal cascade start plus and read at rising edge of CKV. Due tot gate bus-line delay problem, OE signal is used to control turn-off of gate line, so that prevent two-gate bus-line turn on simultaneously. Therefore, CKV and OE have different control timing for different process, so we use 16 bits binary counter to control. See Figure 10.

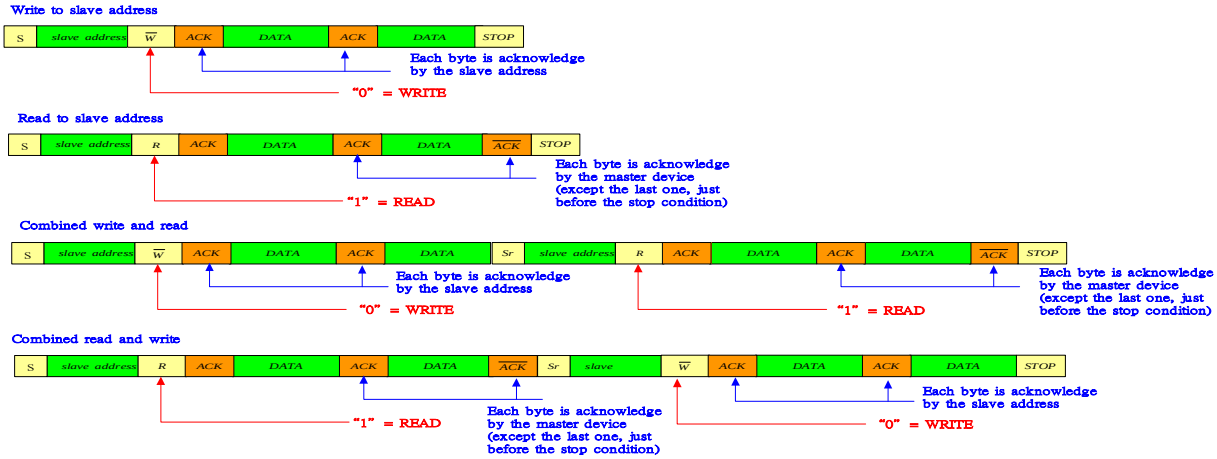

Figure 9. I2C read and write operation.

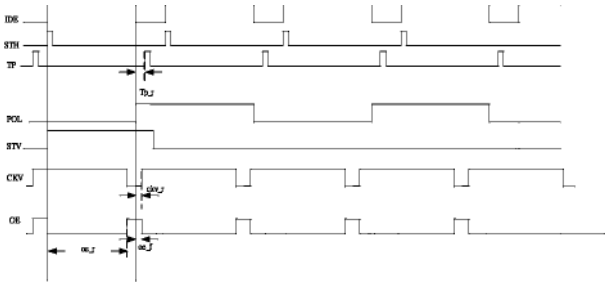

Figure 10. Driver control timing waveform.

3.3: Image Enhance using Frame Rate Control Algorithm

Generally, electronic displays make available the minimum number of levels to represent maximum levels by using digital processing algorithms. TFT-LCD panels usually for monitor or notebook application are using 6-bit driver IC to generate 64 analog levels, so range is from 0 to 63. Again, image data input to TFT-LCD is 8-bit data stream but driver IC is 6-bit data output, so if we using frame rate control algorithm we can get 0 to 252 in 8-bit expression [4] [7].

Frame rate control algorithm is also called dithering algorithm. This algorithm is usually used in printing and electronic display, but they have two major differences as bellow:

(1) The spatial resolution in the case of electronic display is very close to the eye's resolution (to make the display as cheap as possible).

(2) Unlike printing, electronic display is a time component that is available to the algorithm designer to play with.

Each of these factors is exploited to enhance the number of levels created for each pixel. Because the spatial resolution is just at the limit of visibility, and something similar to the spatial modulation encountered in the printing case is adapted for TFT-LCD displays, which is called error diffusion. Based on these characteristics of error diffusion we create 2 units of 2x2 windows for 4 conditions spatial field. It chooses lower 2-bit from 8-bit image data input. In temporal field we extend 4 frames to average 6-bit image data for 8-bit expression. See Figure 11.

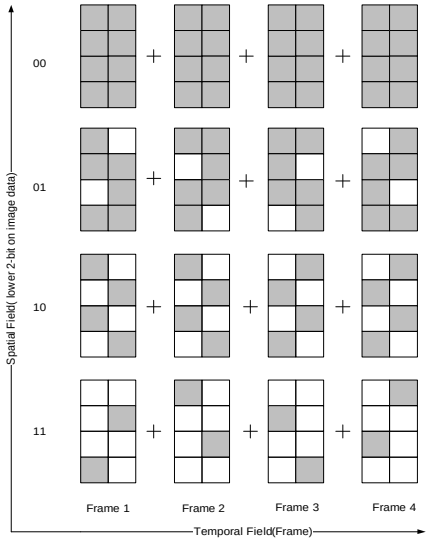

Figure 11. Spatial dithering and temporal averaging pattern on frame rate control

For instance if input image data is 130d (10000010b) gray level, the lower 2-bit is 10b. In Figure 12 it shows that the sequence of the upper left pixel data is 128d gray level. From frame 1 to frame 4, the temporal average value of pixel 1 is (128+128+132+132)/4=130d. The other pixel has different data sequences, but it has some temporal average value of 130d.

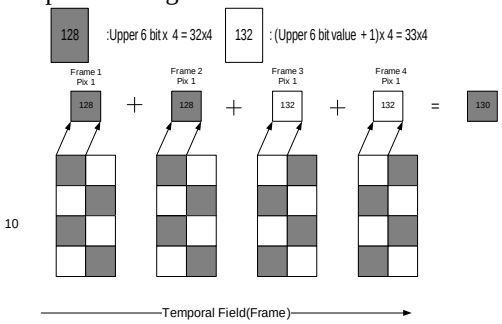

Figure 12. Temporal average of FRC to display luminance of gray level 130d

In Figure 13 it shows 2 units of 2x2 windows which have some average value for 130d. The total spatial average is 130d too. Through all frames, the spatial averaging of 2 units of 2x2 windows is maintained as 130d.

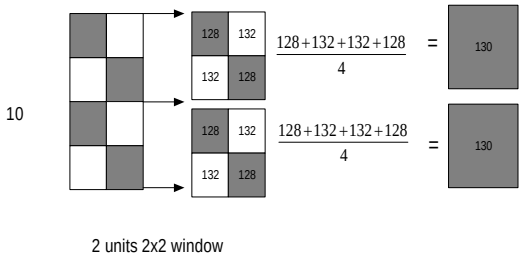

Figure 13. Spatial average of FRC to display luminance of gray level 130d

4: Simulation and Verification Results

In this section, we separate two subsections to illustrate TFT-LCD control timing and frame rate control algorithm. First in RTL simulation we follow section 3 to design column and row driver control timing signal. In section 4.2 we create FPGA-based verification flow to verify frame rate control algorithm in TFT-LCD panel.

4.1: Simulation Results

In Figures 14 and 15, we show that I_2C master module access external EEPROM data from timing controller. After downloading external control timing value, it generates STV, CKV and OE signal to control row driver and STH, TP and POL to control column driver. See Figure 16.

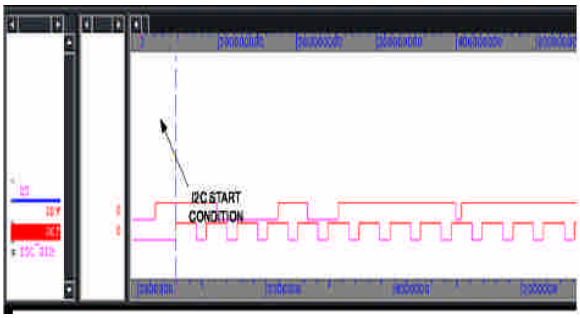

Figure14. I_2C master start condition simulation result.

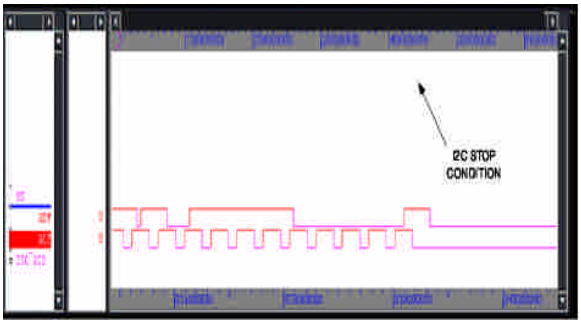

Figure 15. I_2C master stop condition simulation result.

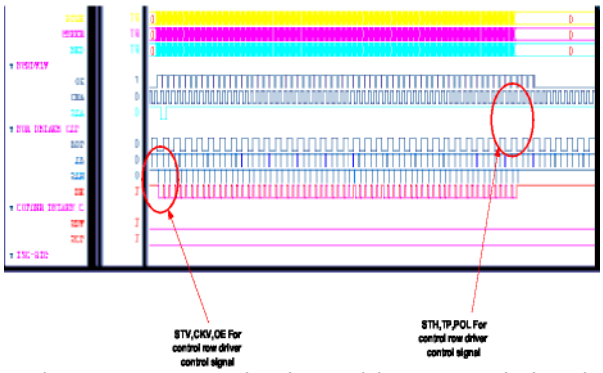

Figure16. Row and column driver control signal simulation results.

4.2: Verification Flow and Results

In this subsection we illustrate the verification flow via Figure 17. We use digital oscillator scope to probe STV, CKV and OE for verification row driver control timing and then probe STH, TP and POL for column driver control timing. See Figure 18.

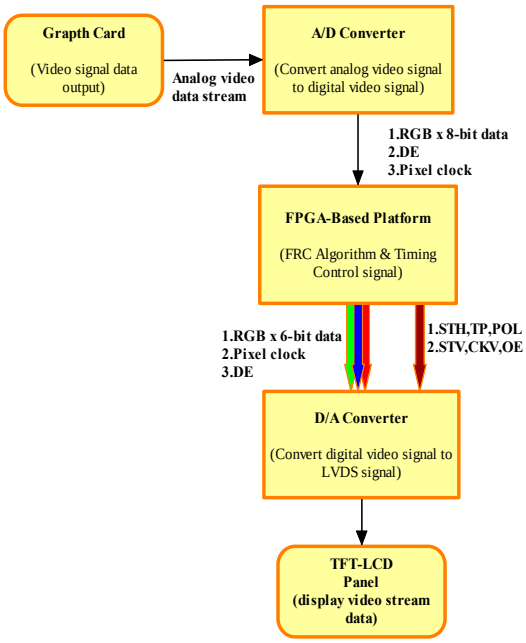

Figure 17. FPGA-Based verification flow.

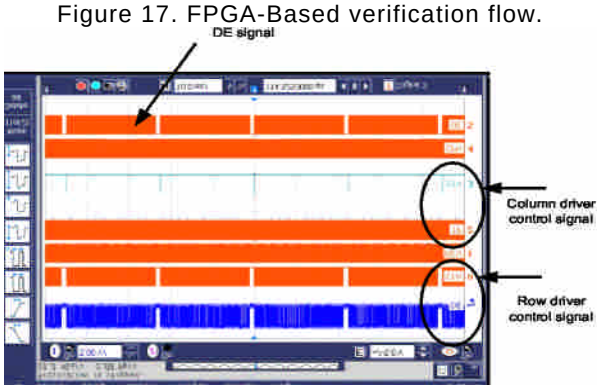

Figure18. Row and column driver control signal for FPGA verification.

We separate two experiments to verify FRC algorithm performance and display image. First we use color analyzer to measure luminance of FRC enable and disable. In Figure 19 we can see when FRC enabled we can get smoothly luminance on TFT-LCD display.

The other experiment we generate vertical 256 gray levels pattern from graph card to verify frame rate control algorithm on panel. If we disable frame rate control algorithm, the RGB data only can display from 0 to 63 gray levels. On the other hand we enable frame rate control algorithm to obtain 8-bit expression image quality.

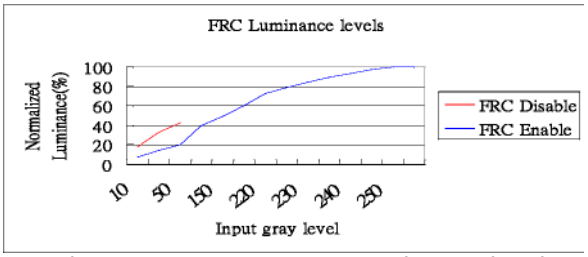

Figure19. Compare FRC Luminance levels

In the experiment we use two sets of TFT-LCD panels to compare with the enable and disable of the frame rate control algorithm. In Figure 20 we show the comparison of the panel. The left is the one that FRC algorithm is disabled, and on the right it is the panel directing 8-bit 256 vertical grays from video card. The image quality is rather different. Again we redo experiment and enable FRC algorithm to get the panel image on the left in Figure 21, which is almost the same as the panel on the right.

Figure 20. Disable frame rate control algorithm on TFT-LCD panel.

Figure 21. Enable frame rate control algorithm on TFT-LCD panel.

5: Conclusions

In this paper, we improve traditional timing controller problems and implement new timing controller that are flexible and reusable for different panel processes. Besides, in image quality, we use frame rate control algorithm to upgrade 262K colors to 16.2M colors.

REFERENCES

[1] Toshihisa Tsukada, "TFT/LCD: Liquid-Crystal Displays Addressed By Thin-Film Transistors", CRC Press, 1996.

[2] Willem den Boer., "Active Matrix Liquid Crystal Displays: Fundamentals and Applications.", Newnes, 2005.

[3] Philips Semiconductor, THE I^2C-BUS SPECIFICATION VERSION 2.1, JANUARY 2000.

[4] Gheorghe Berbecel, "Digital Image Display Algorithm and Implementation," John Wiley & Sons, 2003.

[5] Ya-Hsiang Tai, "Tutorial for Introduction to Display Technology," 2004 Fall.

[6] National Semiconductor, FPD85308-Panel Timing Controller datasheet, 2001.

[7] Seung-Woo Lee, Sang-Soo Kim, "A Novel Dithering Algorithm for High Color Depth and High Color Performance: Hi-FRC," SID'03.