Camera and Multi-Touch Integration with DE2-115

Michael Barker, master student, MS in Electrical Engineering

Manaswi Yarradoddi, master student, MS in Electrical Engineering

Roshini Naidu, master student, MS in Embedded Systems

Contents

INTRODUCTION

Introduction

The goal of the project:

- Was to write a VHDL program that would connect a digital camera and a Multi- touch screen display to an FPGA board and capture live video from the digital camera. The captured image is then displayed on the touch screen display.
- Touching the image and bouncing and zooming it, moving it up and down, Right and left, diminishing the image and enlarging it.

Applications

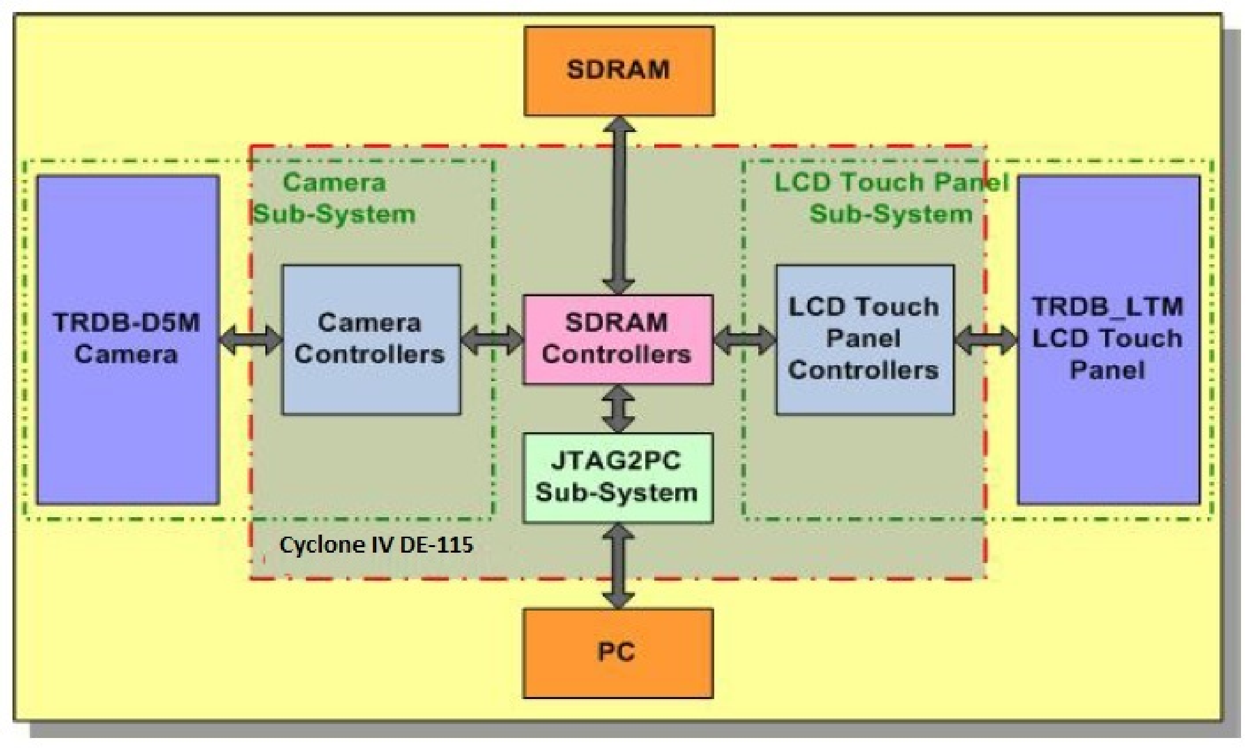
- Image and video processing are used widely in automotive multimedia applications. Examples of such applications are navigation aids and driver information systems

Hardware

Multi-Touch, Camera!!!

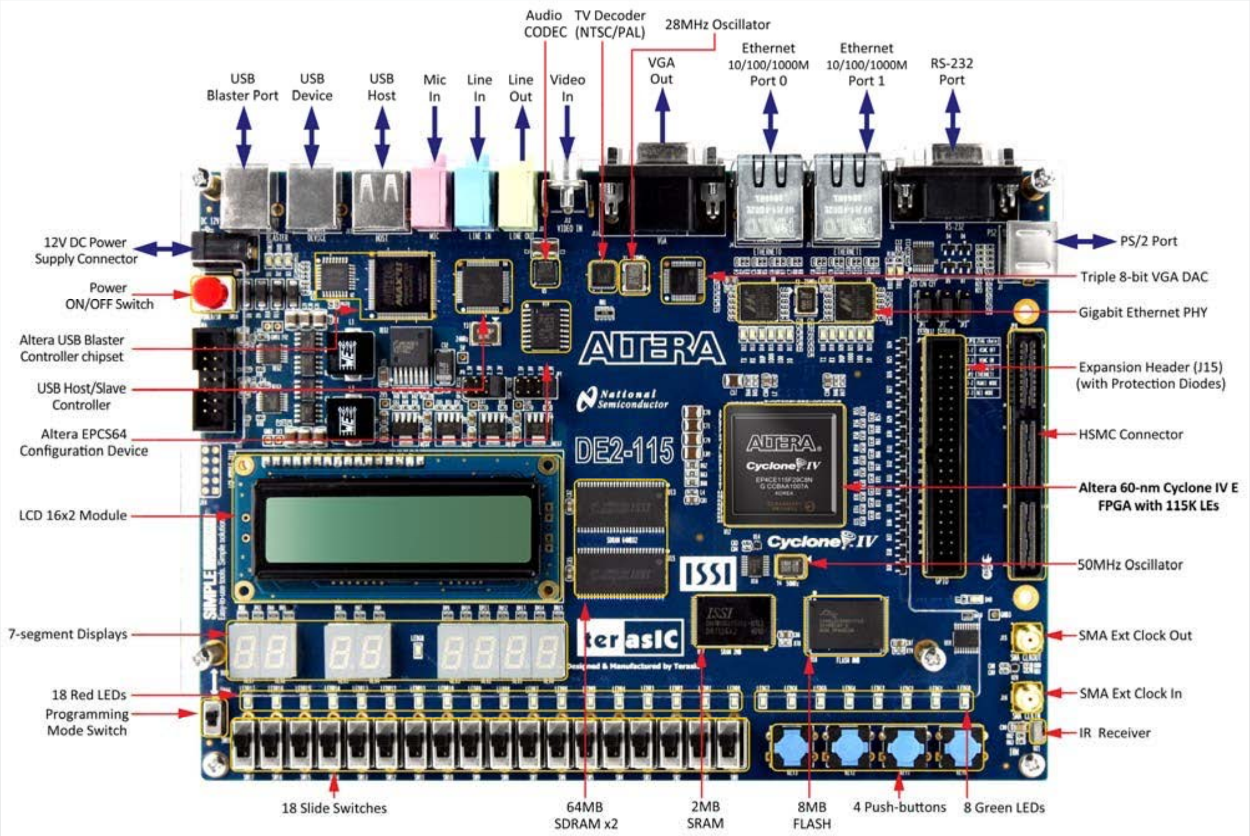
System Block Diagram

- The flexibility of FPGAs gives us the possibility to integrate additional applications and image processing algorithms to the system without any cost in hardware.
- Its offers advantages in terms of lower power consumption, lower cost, and abundance of logic, memory and digital signal processing capabilities.



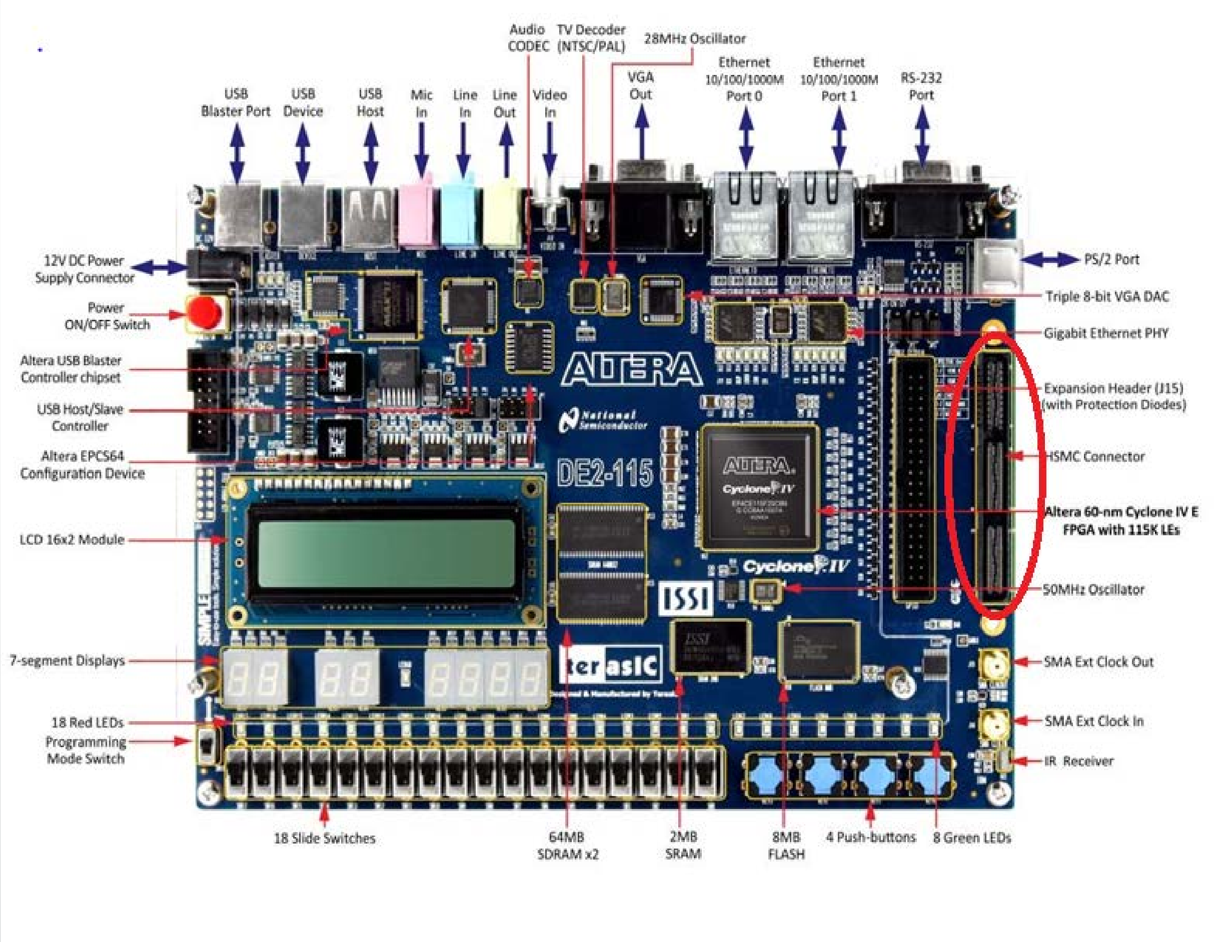
ALTERA DE-115 Board

- Responding to increased versatile low-cost spectrum needs driven by the demand for mobile video, voice, data access, and the hunger for high-quality images, the new DE2-115 offers an optimal balance of low cost, low power and a rich supply of logic, memory and DSP capabilities.



Interfacing FPGA to Camera & Touch Screen

- A High-Speed Mezzanine Card (HSMC) connector is provided to support additional functionality and connectivity via HSMC daughter cards and cables.
- For large-scale ASIC prototype development, a connection can be made with two or more FPGA-based boards by means of a HSMC cable through the HSMC connector.



Multi-Touch LCD

- The touch controller translates x,y coordinates of touch point into digital data. The diagonal length of the touch screen is 7 inches. Its resolution is 800x 3 RGB x 480. Its color arrangement is RGB-stripe.

CAMERA INTERFACE

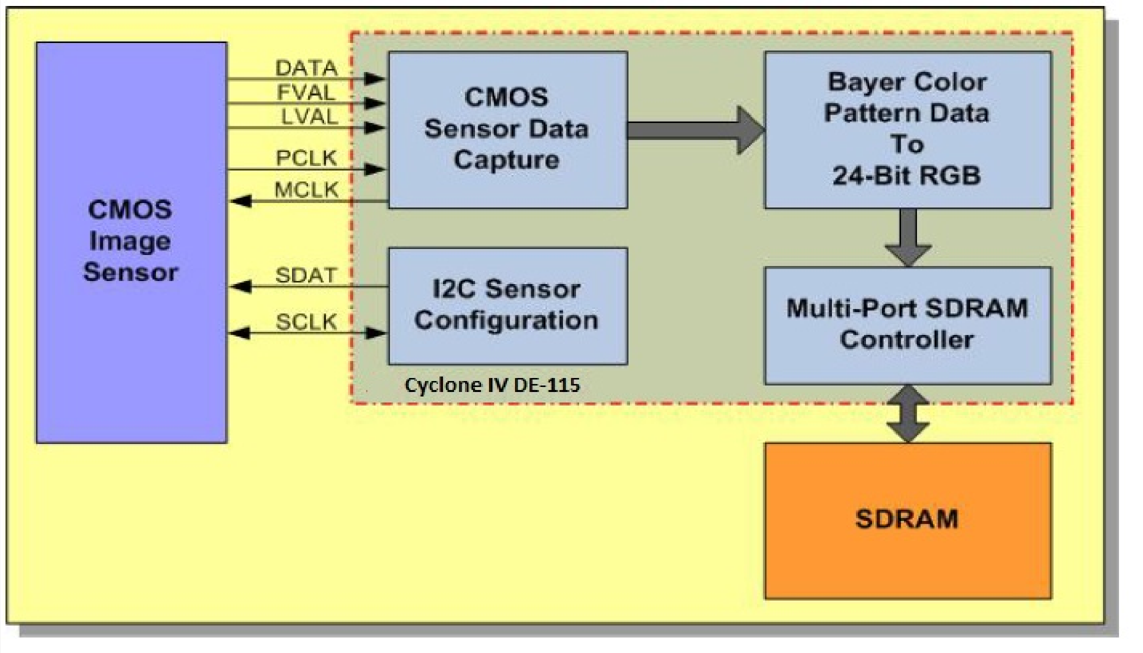


Image Sensor

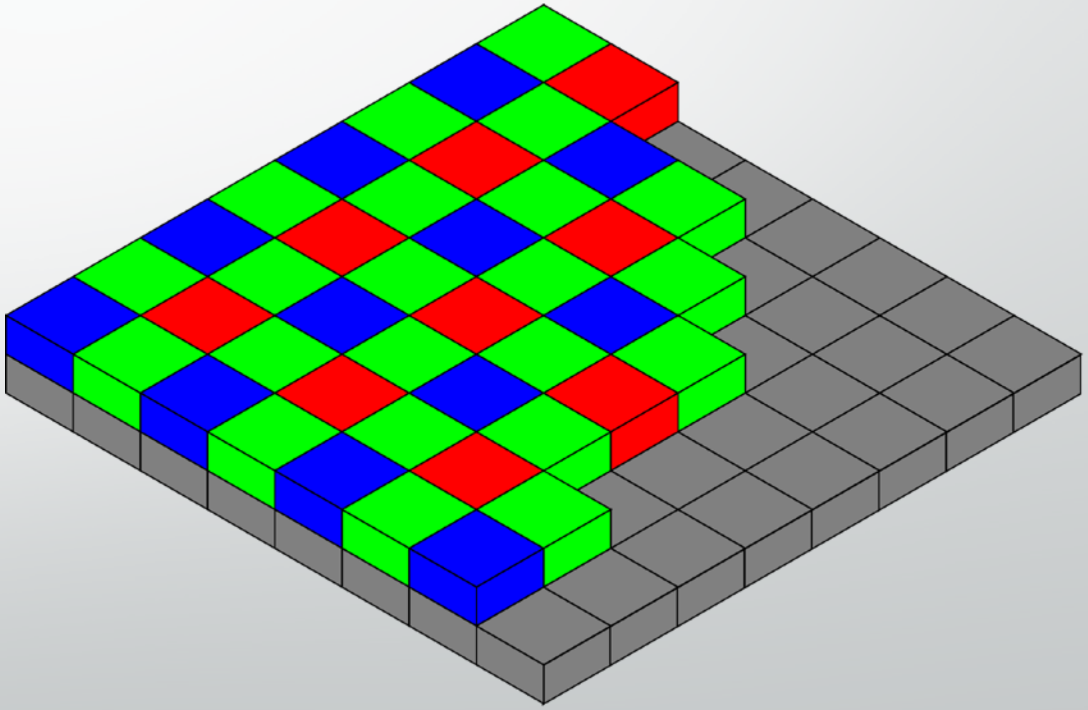
- Has good low light performance
- Improves image quality when resizing.
- The sensor requires 3.3V power supply.
- The maximum signal to noise ratio is 38.1dB.
- The sensor has 70.1dB pixel dynamic range.
- It has a pixel size of 2.2um by 2.2um. It uses RGB Bayer pattern color filter array.
- The sensor has a 12 bit analog digital conversion resolution.

Ambient Light Sensor

- It is used to estimate human-eye response.
- It allows accurate luminance measurement in various lighting conditions.

Bayer Color Pattern

- A Bayer filter mosaic is a color filter array (CFA) for arranging RGB color filters on a square grid of photo sensors. Its particular arrangement of color filters is used in most single-chip digital image sensors used in digital cameras, camcorders, and scanners to create a color image.



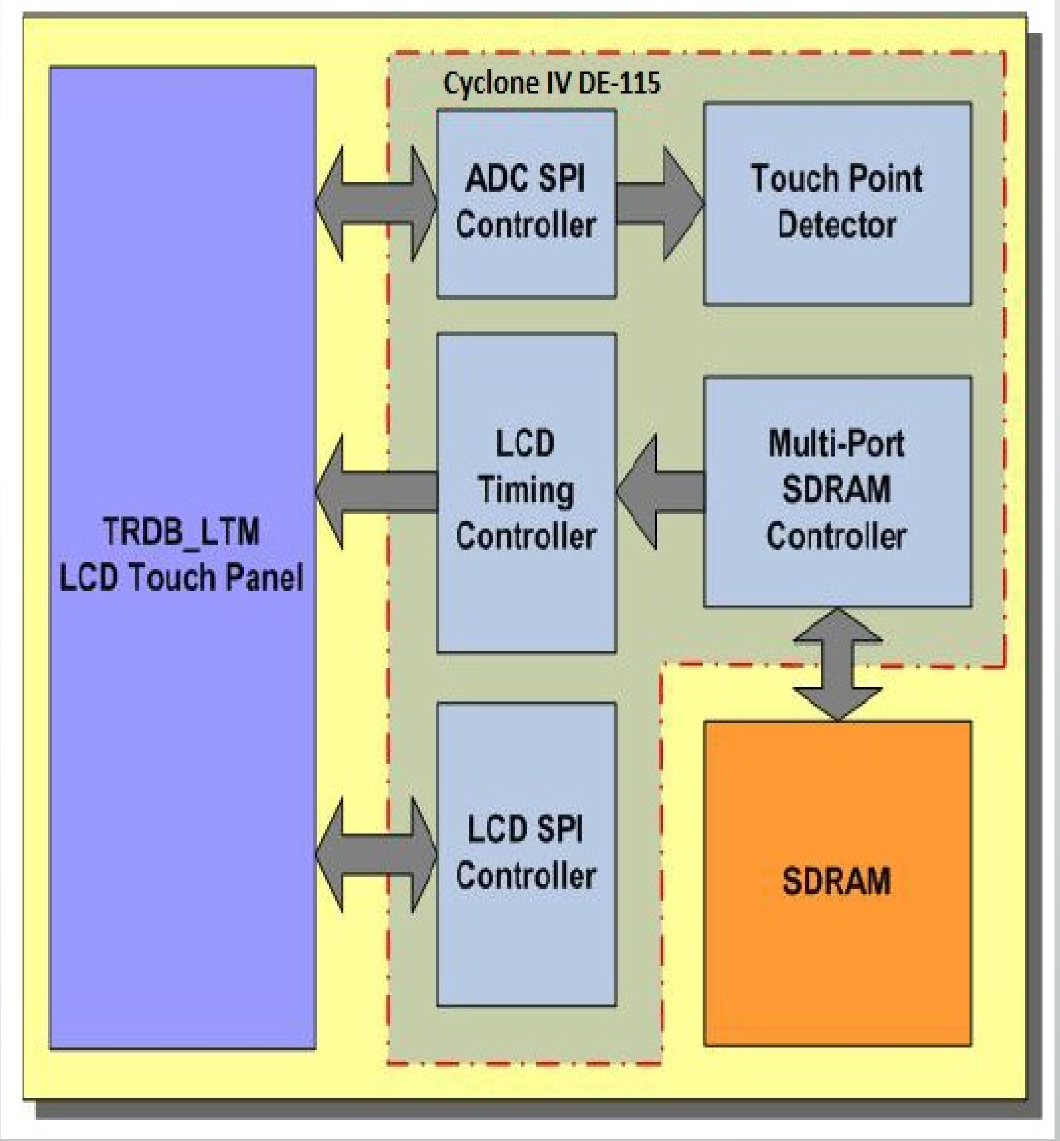
Color coding

- Bg_col=24888444
- The first two digits represent the alpha or the transparent bits.
- The remaining 6 bits represent the RGB color coding.
- Like 88='R', 84='B',44='G'.

		plum	#dda0dd	221,160,221
		powderblue	#b0e0e6	176,224,230
		purple	#800080	128,0,128
		red	#ff0000	255,0,0
		rosybrown	#bc8f8f	188,143,143
		royalblue	#4169e1	65,105,225
		saddlebrown	#8b4513	139,69,19
		salmon	#fa8072	250,128,114
		sandybrown	#f4a460	244,164,96
		seagreen	#2e8b57	46,139,87
		seashell	#fff5ee	255,245,238
		sienna	#a0522d	160,82,45
		silver	#c0c0c0	192,192,192
		skyblue	#87ceeb	135,206,235
		slateblue	#6a5acd	106,90,205
		slategray	#708090	112,128,144
		slategrey	#708090	112,128,144
		snow	#fffafa	255,250,250
		springgreen	#00ff7f	0,255,127
		steelblue	#4682b4	70,130,180
		tan	#d2b48c	210,180,140
		teal	#008080	0,128,128
		thistle	#d8bfd8	216,191,216
		tomato	#ff6347	255,99,71
		turquoise	#40e0d0	64,224,208
		violet	#ee82ee	238,130,238
		wheat	#f5deb3	245,222,179
		white	#ffffff	255,255,255
		whitesmoke	#f5f5f5	245,245,245
		yellow	#ffff00	255,255,0
		yellowgreen	#9acd32	154,205,50

LCD TOUCH PANEL SUB-SYSTEM

- Through the LCD Timing Controller the 24-bit data which are stored in the SDRAM are displayed on the LCD Touch Panel.
- The values of the control registers of the LCD Touch Panel which are related to its function are determined by the LCD SPI Controller.
- Every time touching is being detected at any spot of the LCD Touch Panel, the corresponding analog coordinates are created.



ADC In LCD Touch Screen

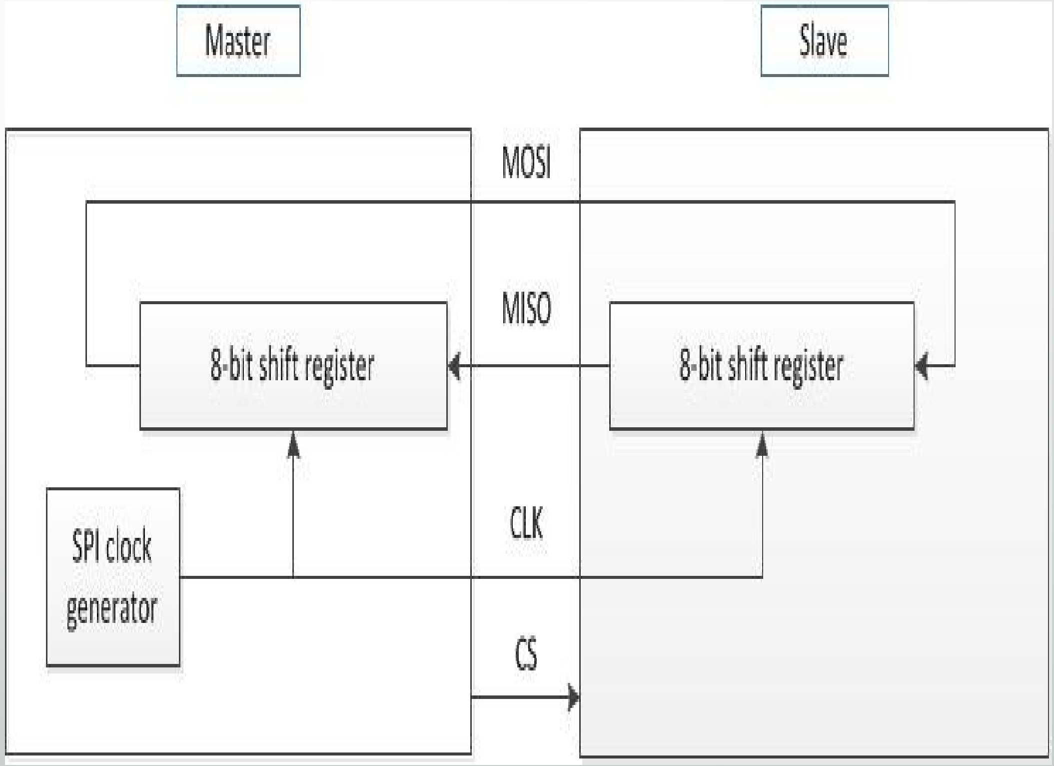
- The Analog Device, ADC transforms the analog coordinates into the corresponding digital data which are sent to the FPGA through the second 40-pin expansion header of DE2-115.
- the resolution of the LCD Touch Panel is 800Hx480V. Because the image that captured from the Camera Sub-system has resolution 640Hx480V.

SPI

- The ADC SPI controller receives the digital signals from the LCD Touch Panel's ADC every time an area on the Panel is activated through touching. Then, it exports two 12-bit numbers which represent the x and y coordinates of the area that has been activated.
- The Touch Point Detector Controller receives the coordinates of the activated areas and sends them to the 7Segment displays of the DE2 in order to be displayed. It also controls if the x and y coordinates reflect a point in one of the predefined active area.

Data Transfer SPI

- The SPI comprises four wires, clock (CLK), Master-Out Slave-In (MOSI), Master In Slave-Out (MISO) and chip select (CS).
- The clock signal CLK is generated by the master to synchronize the exchange of data.
- The MOSI line is used by the master to send commands and data to the slave, while the MISO line is used by the slave to respond to commands and send data back to the master.



I2C

- An Inter-IC bus is often used to communicate across circuit-board distances. Here's a primer on the protocol.
- The name I2C is shorthand for a standard Inter-IC (integrated circuit) bus.

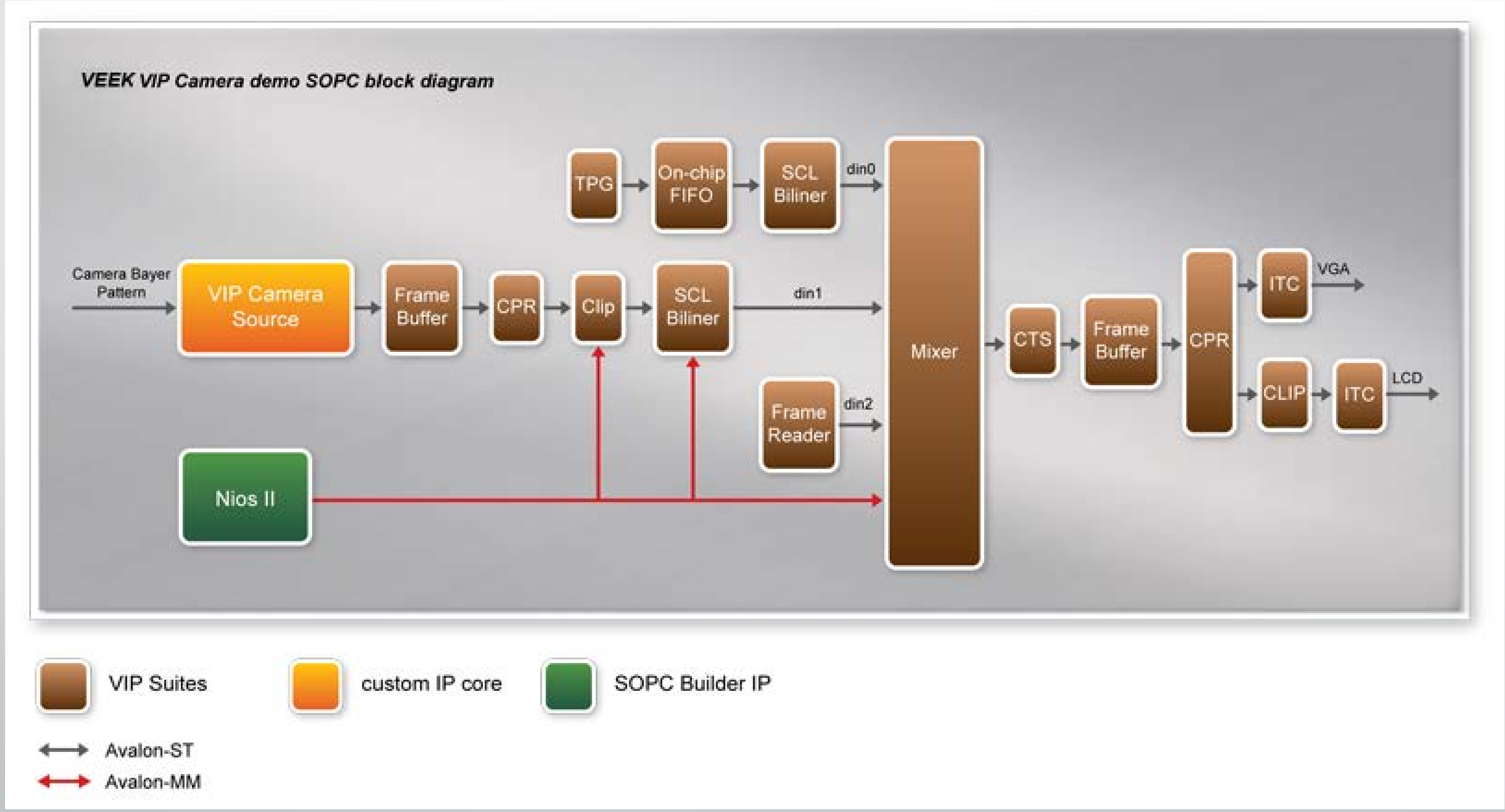
I2C

- I2C provides good support for communication with various slow, on-board peripheral devices that are accessed intermittently, while being extremely modest in its hardware resource needs.
- It is a simple, low-bandwidth, short-distance protocol. Most available I2C devices operate at speeds up to 400Kbps, with some venturing up into the low megahertz range.
- I2C is easy to use to link multiple devices together since it has a built-in addressing scheme.

Software

Verilog, Qsys, and C++ Oh My!

Software Systems Overview



Hardware Setup - Verilog

- Creates various connections in FPGA between camera, multi-touch screen and other DE2-115 board components
- Program uses:
  - Camera
  - Touchscreen
  - 7-segment displays – displays frame capture count in hex from camera
  - Key buttons – read by NIOS II via PIO
  - Switches – Adjust some camera settings

Additional Hardware Functions

- Switches can
  - Mirror part of the camera input
  - Adjust the exposure
- Keys
  - Start and end camera capture
  - Reset system
  - Adjust exposure

Qsys Configuration

Use	C...	Name	Description	Export	Clock	Base	End	IRQ	Ta
✓		clk_50	Clock Source		*exported*				
✓		sys_clk_timer	Interval Timer		clk_sys	0x0900_1500	0x0900_151f	2	
✓		jtag_uart	JTAG UART		clk_sys	0x0900_16d0	0x0900_16d7	1	
✓		button_pio	PIO (Parallel I/O)		clk_sys	0x0900_15c0	0x0900_15cf	0	
✓		led_pio	PIO (Parallel I/O)		clk_sys	0x0900_15e0	0x0900_15ef		
✓		touch_panel_pen_irq_n	PIO (Parallel I/O)		clk_sys	0x0900_1600	0x0900_160f	3	
✓		touch_panel_spi	SPI (3 Wire Serial)		clk_sys	0x0900_1540	0x0900_155f	4	
✓		alt_vip_itc_0	Clocked Video Output		clk_sys				
✓		alt_vip_clip_0	Clipper II		clk_sys	0x0800_0200	0x0800_021f		
✓		alt_vip_scl_0	Scaler II - Edge Adaptive		clk_sys	0x0800_0000	0x0800_01ff		
✓		alt_vip_vfb_0	Frame Buffer		clk_sys				
✓		av_i2c_data_pio	PIO (Parallel I/O)		clk_sys	0x0900_1640	0x0900_164f		
✓		av_i2c_clk_pio	PIO (Parallel I/O)		clk_sys	0x0900_1620	0x0900_162f		
✓		alt_vip_mix_0	Alpha Blending Mixer		clk_sys	0x0900_1000	0x0900_10ff		
✓		alt_vip_custom_tpg_0	alt_vip_custom_tpg		clk_sys	0x0900_16f0	0x0900_16f7		
✓		fifo_1	On-Chip FIFO Memory		clk_sys				
✓		alt_vip_itc_1	Clocked Video Output		clk_sys				
✓		alt_vip__clp_1	Clipper II		clk_sys				
✓		post_fifo_vip_empty_adapter_4	post_fifo_vip_empty_adapter		clk_sys				
✓		audio_avalon_controller	audio_avalon_controller		clk_50	0x0900_1660	0x0900_167f	5	
✓		td_reset_pio	PIO (Parallel I/O)		clk_sys	0x0900_16a0	0x0900_16af		
✓		alt_vip_vfr_0	Frame Reader		*multiple*	0x0900_1200	0x0900_127f	6	
✓		alt_vip_cts_0	Control Synchronizer		clk_sys	0x0900_1300	0x0900_137f	7	
✓		alt_vip_cpr_0	Color Plane Sequencer		clk_sys				
✓		sdram	SDRAM Controller		clk_sys	0x0000_0000	0x07ff_ffff		
✓		sram	TERASIC_SRAM		clk_sys	0x0a20_0000	0x0a2f_ffff		
✓		altpll_0	Avalon ALTPLL		clk_50	0x0900_16e0	0x0900_16ef		
✓		alt_vip_vfb_2	Frame Buffer		clk_sys				
✓		alt_vip_cti_0	Clocked Video Input		clk_sys				
✓		alt_vip_cpr_2	Color Plane Sequencer		clk_sys				
✓		multi_touch	TERASIC_MULTI_TOUCH		clk_50	0x0900_1400	0x0900_147f	8	
✓		cpu	Nios II Processor		clk_sys	0x0900_0800	0x0900_0fff		
✓		sysid	System ID Peripheral		clk_sys	0x0900_16c0	0x0900_16c7		
✓		tri_state_bridge_flash_bridge_0	Tri-State Conduit Bridge		clk_sys				
✓		tri_state_bridge_flash_pinSharer_0	Tri-State Conduit Pin Sharer		clk_sys				
✓		cfi_flash	Generic Tri-State Controller		clk_sys	0x0880_0000	0x08ff_ffff		
✓		c1	Clock Bridge		clk_sys				

Qsys Configuration

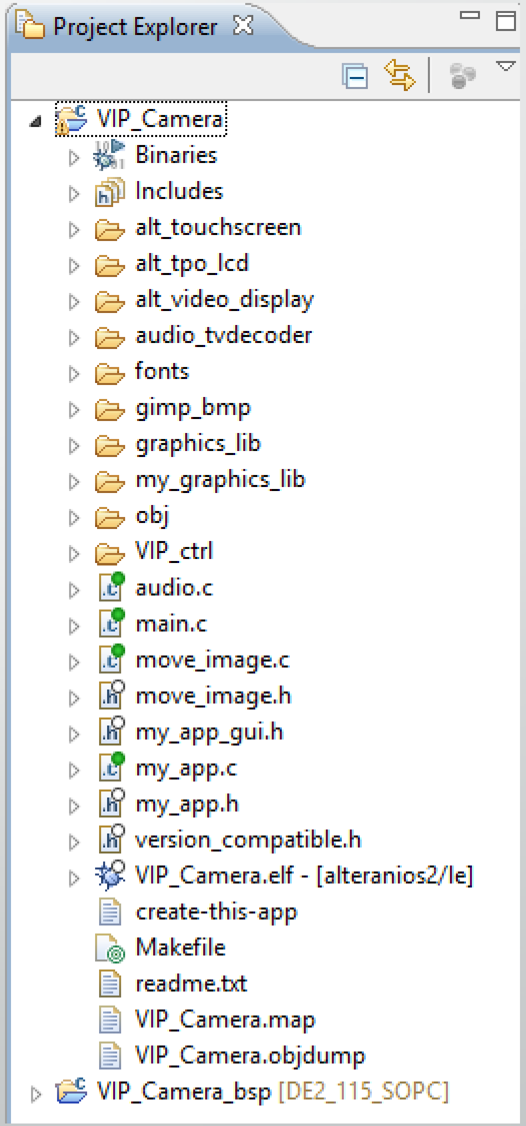
- Contains bulk of configurations for system
- Includes configurations for
  - Frame buffers
  - Video clippers
  - Video mixer
  - SDRAM and SRAM
  - Multi-touch touchscreen
  - Parallel I/O ports for DE2-115 hardware
  - NIOS II

Interrupt Based Controls

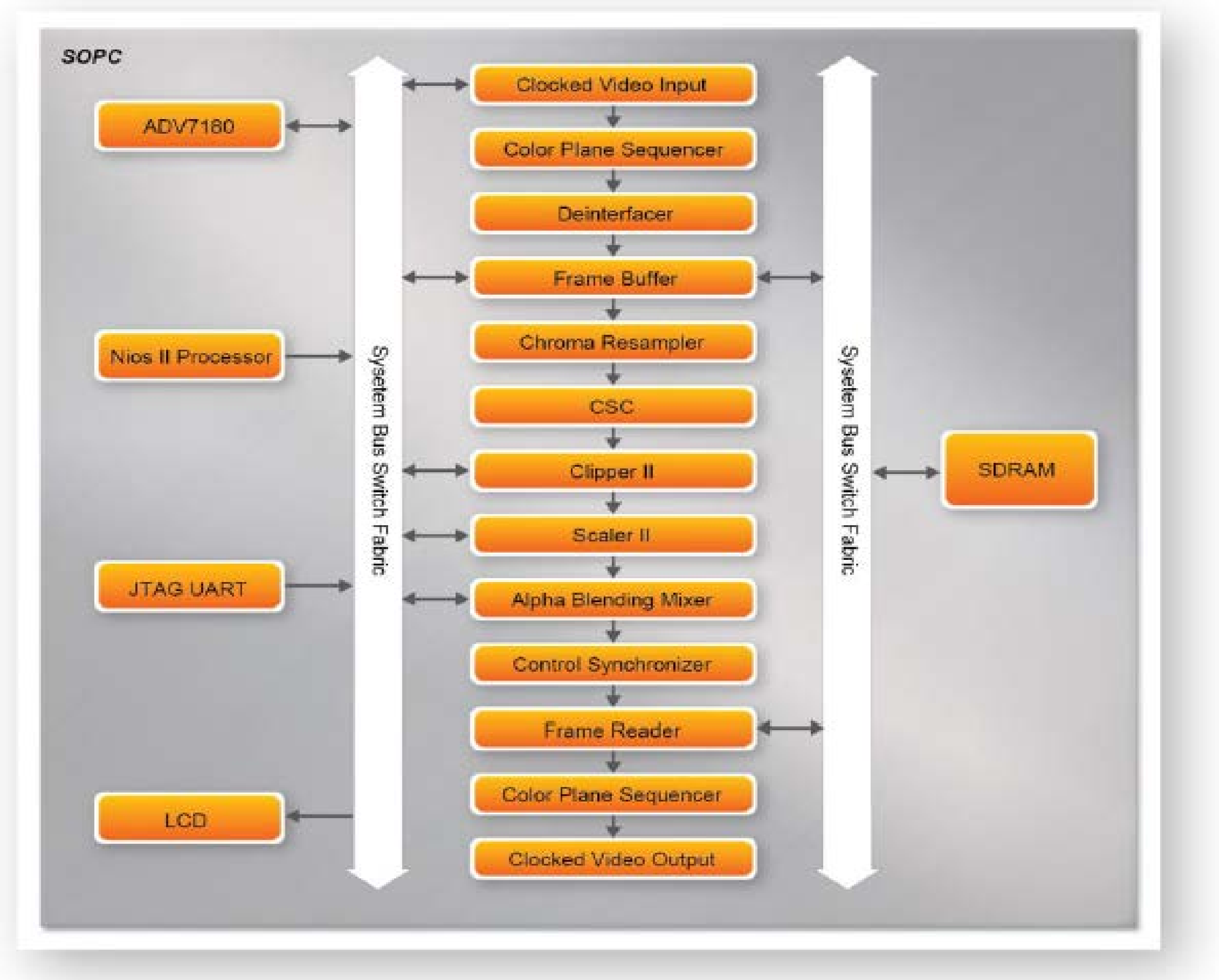
- Interrupts for the NIOS II are generated by
  - DE2-115 Buttons
  - JTAG UART
  - System clock timer
  - Touch panel input and SPI
  - Audio controller
  - Frame Reader
  - Control Synchronizer
  - Multi-touch touchscreen

NIOS II C Code Components

- Control of system is written predominately in C++
- Code contains may IP core functions from Altera
- Most is part of Video and Image Processing (VIP) cores



VIP Core



SOPC
ADV7180
Nios II Processor
JTAG UART
LCD
Sysetem Bus Switch Fabric
Clocked Video Input
Color Plane Sequencer
Deinterlacer
Frame Buffer
Chroma Resampler
CSC
Clipper II
Scaler II
Alpha Blending Mixer
Control Synchronizer
Frame Reader
Color Plane Sequencer
Clocked Video Output
Sysetem Bus Switch Fabric
SDRAM

Video Processing Components

- Frame Reader
  - Read video from external memory and outputs it as a stream
- Control synchronizer
  - Synchronized the changes made to the video stream in real time between two functions
- Scaler
  - Allows custom scaling and real-time updates to image sizes and scaling
- Clipper
  - Clips video streams. Can be set to be configured at compile or run-time

Video Processing Components

- Mixer
  - Mixes and blends multiple video streams. Used for overlays and picture-in-picture
- Frame Buffer
  - Buffers video frames into external RAM. Includes options for frame dropping and repeating
- Gamma Corrector
  - Adjusts video properties for the display

What Does the VIP Camera Program Do?

- Continually captures images from camera
- Creates a video feed with 3 layers
  - Background
  - Camera feed
  - Title bar text overlay

Camera Feed Video Manipulation

- Camera feed has multiple modes and features
  - Touching screen will allow user to:
    - Move feed
    - Resized feed
    - “Throw” feed so it bounces around screen
  - Without user input it will
    - Rescale itself
    - Move around the screen

Code

```
volatile int go_time_update_flag=0;
volatile int go_pen_update_flag=0;
volatile int ticks=0;
/*******************************************************/
volatile int time_sec=00;
volatile int time_min=30;
volatile int time_hour=5;
alt_u32 time_alarm_callback(void * context){

    go_time_update_flag = 1;

  ticks+=ALARM_CYCLE_TIME_UPDATE;
  if (ticks >= alt_ticks_per_second()){

    ticks -= alt_ticks_per_second();
    time_sec++;
    if (time_sec>=60) {
        time_sec = 0;
        time_min++;
        if (time_min>=60){
            time_min=0;
            time_hour++;
            if (time_hour>=24)
                time_hour=0;
        }
    }


  }
  return ALARM_CYCLE_TIME_UPDATE;
}
```

```
extern void init_i2c();
extern void move_image(int xini, int yini, int wini, int hini, int direcini);
```

```
snprintf(strbuff,256,"%02d:%02d:%02d", time_hour, time_min, time_sec);
sw = 144;
```

```
void update_grapics(int write_all){

static int w2 = FRAME_BUF_W;
static int h2 = FRAME_BUF_H;


static int col_var=0;


char strbuff[256];
int sw;

if (write_all){
  set_frame_color(display, GRAPH_BG_COL);

        snprintf(strbuff,256,"%s","ECE 576 Project");
        sw = vid_string_pixel_length_alpha( tahomabold_32, strbuff );
        vid_print_string_alpha(10, h2/2 -32/2 -1 +0, INDIANRED_24, GRAPH_BG_COL,
                    tahomabold_32, display, strbuff);
```

```c
int touchscreen_event_handling(MTC_INFO *pTouch){

  int pen_x;
  int pen_y;
  int pen_is_down;
  static int pre_pen_x;
  static int pre_pen_y;
  static int pre_pen_x1;
  static int pre_pen_y1;
  static int pre_pen_is_down;

  if (MTC_GetStatus(pTouch, &Event, &TouchNum, &pen_x, &pen_y, &pre_pen_x1, &pre_pen_y1))
  {
    pen_is_down = 1;
  }
  else
  {
    pen_is_down = 0;
  }

  if ( pen_is_down ) {
    if (pre_pen_is_down)
      pen_move(pen_x, pen_y);
    else
      pen_down(pen_x, pen_y);
  }
  else if (pre_pen_is_down) {
      pen_up(pen_x, pen_y);
  }

  pre_pen_x = pen_x;
  pre_pen_y = pen_y;
  pre_pen_is_down = pen_is_down;

  usleep(15*1000); //add for mtlc

  return pen_is_down;

}
```

```c
//bg_col = 0x24888444; //First two digits are transparency, last 6 are color
  bg_col = 0x24000000;
IOWR(ALT_VIP_CUSTOM_TPG_0_BASE,0,bg_col);

// LCD display area size
int w_max=LCD_DISPLAY_W;
int h_max=LCD_DISPLAY_H;

// Live image  default size/pos
int w1 = LIVE_IMAGE_W;
int h1 = LIVE_IMAGE_H;
int x1 = (w_max - w1)/2;
int y1 = (h_max - h1)/2;

// NiosII default size/pos
title_bar_w = FRAME_BUF_W;
title_bar_h = FRAME_BUF_H;
title_bar_x = (w_max - title_bar_w)/2;
title_bar_y = 0;

// Set up Nios II frame buffer background color
set_frame_color(display, GRAPH_BG_COL);
```

```
// update frame size/pos variables
if (! touchscreen_event){
    if (auto_timer<=0) {  // move by inertia
    {  bg_col &= 0x0;     //
      IOWR(ALT_VIP_CUSTOM_TPG_0_BASE,0,bg_col ) ;
     }
      if (motion_count==5) {
        free_fall_image();
        }


      else if (motion_count==4) {
        move_image_top_center(-1,-1,-1,-1,-1);
        }

      else if (motion_count==3) {
        pan_and_scroll_image(-1,-1,-1,-1,-1);

        }

      else if (motion_count==2) {
        zoom_and_scroll_image(-1,-1,-1,-1,-1);

        }

      else if (motion_count==1) {
        move_image_center(-1,-1,-1,-1,-1);

       }
      else{
        motion_count=0;
        move_image(-1,-1,-1,-1,-1);

      }
      bounce_image_reset();
    }
    else {
      bounce_image();
    }
  }
```

```
    // push button handling (KEY3->KEY0 match bits 3->0)
    button = IORD(BUTTON_PIO_BASE,0) & 0xf;
    if (button != 0xf) {
      switch (button) {
      case 0xd : //0x1101
        title_bar_y+=4;
        if( title_bar_y > (h_max - FRAME_BUF_H) ) title_bar_y = (h_max - FRAME_BUF_H);
          Mixer_set_layer_position(2, title_bar_x, title_bar_y);
        usleep(1000*5);
        break;
      case 0xe : //0x1110
        title_bar_y-=4;
        if (title_bar_y<0) title_bar_y=0;
          Mixer_set_layer_position(2, title_bar_x, title_bar_y);
        usleep(1000*5);
        break;
      case 0x7 : //0x0111
        time_min++;
        if (time_min>59) time_min = 0;
        update_grapics(0);
        usleep(1000*200);
        break;
      case 0xb ://0x1011
        time_hour++;
        if (time_hour>23) time_hour = 0;
        update_grapics(0);
        usleep(1000*200);
        break;
      }
    }

  } // end of while(1) loop

  return ( 0 );
}
```

Video Demonstration

Conclusion

- FPGAs’ flexibility, is mainly targeting to be used as an open and low cost platform for implementing and testing real-time image processing algorithms.
- In addition the exploitation of LCD Touch Panel can effectively assist in the control of more camera’s parameters.
- Image processing algorithms can take place before or after the data storing and because of the FPGA’s presence, system has the ability to be easily modified.
- In addition we intend to create an extended menu for the LCD touch panel. Developing such a menu the user can fully and in a friendly manner control Camera’s functionality.

References

- www. Terasic.com
- www. Altera.com
- www.Wikipedia.com

Thank You