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
• Hardware description
• Software description
• Code and demonstration
• Conclusion
• References
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!!!
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.
AL TERA 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 $800 \times 3 \text{ RGB} \times 480$. 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’. 

<table>
<thead>
<tr>
<th>Color</th>
<th>Hex Code</th>
<th>RGB</th>
</tr>
</thead>
<tbody>
<tr>
<td>plum</td>
<td>#da0dd</td>
<td>221,160,221</td>
</tr>
<tr>
<td>powderblue</td>
<td>#b0e66</td>
<td>176,224,230</td>
</tr>
<tr>
<td>purple</td>
<td>#800080</td>
<td>128,0,128</td>
</tr>
<tr>
<td>red</td>
<td>#ff0000</td>
<td>255,0,0</td>
</tr>
<tr>
<td>rosybrown</td>
<td>#bc8f8f</td>
<td>188,143,143</td>
</tr>
<tr>
<td>royalblue</td>
<td>#4169e1</td>
<td>65,105,225</td>
</tr>
<tr>
<td>saddlebrown</td>
<td>#8b4513</td>
<td>139,69,19</td>
</tr>
<tr>
<td>salmon</td>
<td>#fa8072</td>
<td>250,128,114</td>
</tr>
<tr>
<td>sandybrown</td>
<td>#fa4600</td>
<td>244,164,96</td>
</tr>
<tr>
<td>seagreen</td>
<td>#2e8b57</td>
<td>46,139,87</td>
</tr>
<tr>
<td>seashell</td>
<td>#ffffff</td>
<td>255,255,255</td>
</tr>
<tr>
<td>sienna</td>
<td>#a0522d</td>
<td>160,82,45</td>
</tr>
<tr>
<td>silver</td>
<td>#c0c0c0</td>
<td>192,192,192</td>
</tr>
<tr>
<td>skyblue</td>
<td>#87ceeb</td>
<td>135,206,235</td>
</tr>
<tr>
<td>slateblue</td>
<td>#6a5acd</td>
<td>106,90,205</td>
</tr>
<tr>
<td>slategray</td>
<td>#708090</td>
<td>112,128,144</td>
</tr>
<tr>
<td>snow</td>
<td>#708090</td>
<td>112,128,144</td>
</tr>
<tr>
<td>springgreen</td>
<td>#00ff7f</td>
<td>0,255,127</td>
</tr>
<tr>
<td>steelblue</td>
<td>#4682b4</td>
<td>70,130,180</td>
</tr>
<tr>
<td>tan</td>
<td>#d2b48c</td>
<td>210,180,140</td>
</tr>
<tr>
<td>teal</td>
<td>#008000</td>
<td>0,128,128</td>
</tr>
<tr>
<td>thistle</td>
<td>#d8bfd8</td>
<td>216,191,216</td>
</tr>
<tr>
<td>turquoise</td>
<td>#ff6347</td>
<td>255,99,71</td>
</tr>
<tr>
<td>tomato</td>
<td>#40e0d0</td>
<td>64,224,208</td>
</tr>
<tr>
<td>violet</td>
<td>#ee82ee</td>
<td>238,130,238</td>
</tr>
<tr>
<td>wheat</td>
<td>#ffdeb3</td>
<td>245,222,179</td>
</tr>
<tr>
<td>white</td>
<td>#ffffff</td>
<td>255,255,255</td>
</tr>
<tr>
<td>whitesmoke</td>
<td>#f5f5f5</td>
<td>245,245,245</td>
</tr>
<tr>
<td>yellow</td>
<td>#ffff00</td>
<td>255,255,0</td>
</tr>
<tr>
<td>yellowgreen</td>
<td>#9acd32</td>
<td>154,205,50</td>
</tr>
</tbody>
</table>
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
<table>
<thead>
<tr>
<th>Use</th>
<th>C...</th>
<th>Name</th>
<th>Description</th>
<th>Export</th>
<th>Clock</th>
<th>Base</th>
<th>End</th>
<th>I...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>clk_50</td>
<td>Clock Source</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>sys_clk_timer</td>
<td>System Timer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>jtag_uart</td>
<td>JTAG UART</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>button_pio</td>
<td>PIO (Parallel I/O)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>led_pio</td>
<td>PIO (Parallel I/O)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>touch_panel_pen_irq_n</td>
<td>PIU (Parallel I/O)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>touch_panel_spi</td>
<td>SPI (3 Wire Serial)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_itc_0</td>
<td>Clocked Video Output</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_dic_0</td>
<td>Clipper II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_scl_0</td>
<td>Slicer II - Edge Adaptive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_vfb_0</td>
<td>Frame Buffer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>av_i2c_data_pio</td>
<td>PIU (Parallel I/O)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>av_i2c_clk_pio</td>
<td>PIU (Parallel I/O)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_mix_0</td>
<td>Alpha Blending Mixer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_custom_top_0</td>
<td>alt_vip_custom_top</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>fifo_1</td>
<td>On-Chip FIFO Memory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_itc_1</td>
<td>Clocked Video Output</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_vfb_1</td>
<td>Clipper II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>post_fifo_vip_empty_adapter_4</td>
<td>post_fifo_vip_empty_adapter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>audio_avlon_controller</td>
<td>audio avlon_controller</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>id_reset_pio</td>
<td>PIO (Parallel I/O)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_vfr_0</td>
<td>Frame Reader</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_cts_0</td>
<td>Control Synchronizer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_cpr_0</td>
<td>Color Plane Sequencer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>sdram</td>
<td>SDRAM Controller</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>sram</td>
<td>TERAASIC SRAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>aipll_0</td>
<td>Avlon ALTPLL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_vfb_2</td>
<td>Frame Buffer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_vfi_0</td>
<td>Clocked Video Input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>alt_vip_cpr_2</td>
<td>Color Plane Sequencer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>multi_touch</td>
<td>TERAASIC_MULTI_TOUCH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cpu</td>
<td>Nios II Processor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>syslog</td>
<td>System ID Peripheral</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>tri_state_bridge_flash_bridge_0</td>
<td>Tri-State Conduct Bridge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>tri_state_bridge_flash_pinsharer_0</td>
<td>Tri-State Conduct Pin Sharer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cli</td>
<td>Generic Tri-State Controller</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cl</td>
<td>Clock Bridge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 many IP core functions from Altera
- Most is part of Video and Image Processing (VIP) cores
VIP Core
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
volatile int go_time_update_flag=0;
volatile int go_pen_update_flag=0;
volatile int ticks=0;

volatility int time_sec=00;
volatile int time_mli=38;
volatile int time_hour=0;

alt_uX 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_12c();
extern void move_image(int xini, int yini, int wini, int hni, int direcini);

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

void update_graphics(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_BK_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 +8, INDIAHRED_24, GRAPH_BK_COL, tahomabold_32, display, strbuff);
    }
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_SetStatus(pTouch, &Event, &TouchNum, &pen_x, &pen_y, &pre_pen_x, &pre_pen_y, &pre_pen_is_down))
    {
        pen_is_down = 1;
    }
    else
    {
        pen_is_down = 0;
    }
    
    if (pen_is_down) {
        if (pre_pen_is_down)
            pen_mov(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(1510000); // Add for MILC
    return pen_is_down;
}

// bg_col = 0x24885444; // First two digits are transparency, last 6 are color
bg_col = 0x24000000;
DONE(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;

// MiosII default size/pos

// Set up Mios II frame buffer background color
set_frame_color(display, GRAPH_BG_COL);
// update frame size/pgx variables
if (!touchscreen_event)
  if (auto_timer==0) { // move by inertia
      bg_col &= 0x8;  // IONR(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 (KEYS->KEY0 match bits 3->0)
button = IOND(BUTTON_PIO_BASE,0) & 0xf;
if (button != 0xf) {
  switch (button) { 
    case 0x0 : //0x1001
      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 0x1 : //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 0x2 : //0x1111
      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_graphics(0);
      usleep(1000*200);
      break;
    case 0x8 : //0x1011
      time_hour++;
      if (time_hour>23) time_hour = 0;
      update_graphics(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