SketchMaster

A DRAWING TOOL
EMBEDDED SYSTEMS DESIGN

AJAY VANAMALI
MANISH SHANKAR
RAHUL SHANBHAG
BHOOMI SHAH

Introduction

We bring you Sketch Master, a drawing tool that allows users to learn how to draw digitally. We have interfaced a Wacom DTF 510 with the DE1-SOC. Custom VGA peripheral and libusb based drivers were used to capture the user input and display the artwork on screen.

Goal

- Create a closed system that allows a user to draw on a screen in real time.
- Provide a simple UI to learn how to use a tablet

HPS

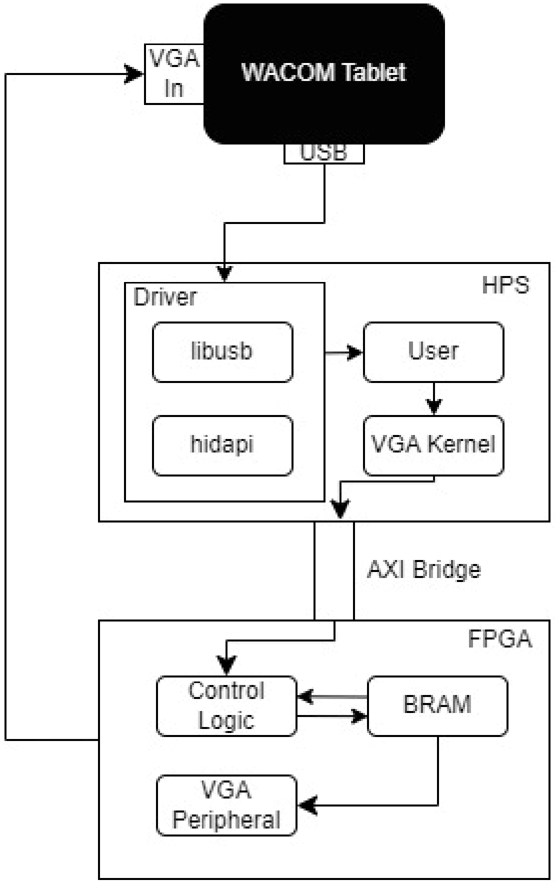
- Reads input from USB output of the the Wacom DTF 150
- Converts input to corresponding image pixels.
- Writes into frame buffer block ram

FPGA

- Takes the image frame buffer from the RAM
- Outputs it to the the custom VGA peripheral

System Architecture

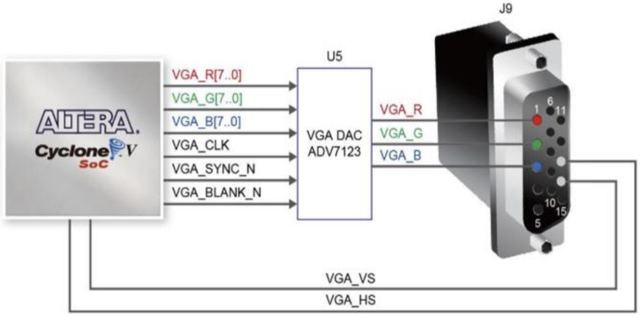
- Input
- Input Processing
- User Application
- Peripheral Drivers
- Block RAM
- VGA Peripheral
- VGA Output



Hardware Design

There Are Two Major Hardware Components :

- **The Custom VGA Peripheral**
  - The VGA module handles pixel selection for different areas of the disp determined by the stylus, including user-drawn images,the color palette the stylus buttons.
  - It communicates with the HPS via the Avalon bus to obtain pixel input stores it in the BRAM for future use, ensuring the persistence of the displayed image.
  - The module takes various inputs, such as the pen tap location, to determ state transitions.
  - For example, if the pen tap is on the color red, the module records this information, updates the internal pen color, stores it in the BRAM, and uses the appropriate color for subsequent drawing commands.
  - The VGA module facilitates the coordination of pixel sources, communication with the HPS, and state transitions to maintain and update the displayed image accurately.

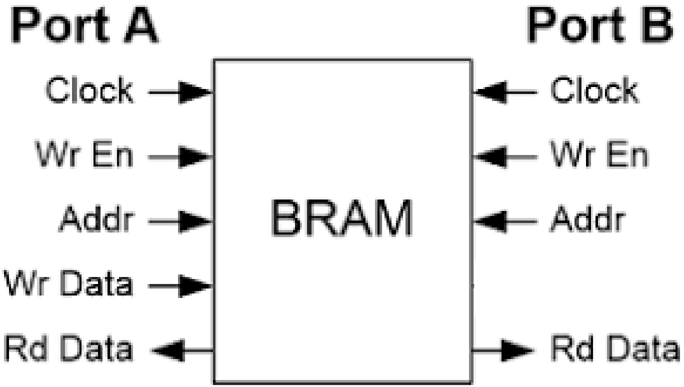


The peripheral produces the above lines to send out the image in a raster format.

Hardware Design

There Are Two Major Hardware Components :

- **The Dual Ported BRAM**
  - BRAM has 3-bit data width, suitable for small data applications.
  - 307,200 memory locations, allowing pixel-to-location mapping for VGA display.
  - 19-bit address width, addressing up to 524,288 memory locations.
  - Dual-port read enables simultaneous high-speed data access.



Software Design

Interfacing with the Wacom DTF 510

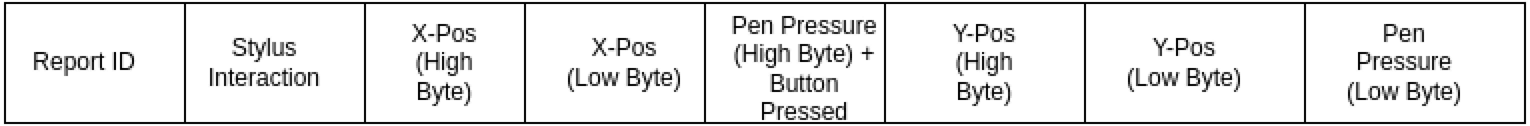
- The DTF 510 is a pen-based graphics tablet with a built in VGA input and USB output. The tablet acts as a Human Interface Device (HID) that transmits data via USB packets accessible via the USB-HID protocol.
- The output packets contain primarily “pen data” that talks about the various modes and positions of the included pen with respect to the tablet surface.
- We used hid-api a multi-platform library which allows an application to interface with USB and Bluetooth HID-Class devices on Linux. In our case, we use the libusb backend to communicate directly with the tablet. We first communicate with the tablet and configure the mode we want it to operate under and after that we receive packet data about the pen use.

Wacom Driver

- We use the HID API, an open source multi-platform library that allows systems to interface with a wide range of HID devices.
- While custom linux Wacom drivers that allow users to interface with such graphics tablets exist, they do not have support for the Wacom DTF 510 without the X-Driver.
- HID-API allows us to interface with Wacom DTF 510.
  - The Linux device tree - treats it as a mouse
  - To resolve this issue, we write a specific set of values to configure the device as a tablet.
  - We configure the device as a Generic Bamboo Pen Touch Tablet by writing [0x02 0x02 0x02] to the device.
- Now that the device has been configured as a tablet, we receive 8 byte packets from the device which have a specific format, which is explained in the next slide.
- With the received packets, we calculate the absolute position of the stylus and map it to our 640*480 VGA monitor.

Wacom Tablet Packet Format

- The Wacom DTF 510 8 Byte Packet Format using HID-API



Report ID	Stylus Interaction	X-Pos (High Byte)	X-Pos (Low Byte)	Pen Pressure (High Byte) + Button Pressed	Y-Pos (High Byte)	Y-Pos (Low Byte)	Pen Pressure (Low Byte)

User Interface

Writing Area: 81:630 pixels horizontally. 81:470 pixels vertically

Features:

Color Selection:

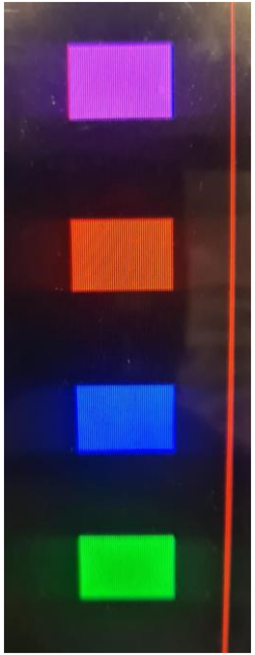
- Users can choose from up to 7 different colors for their drawings
- The finite state machine (FSM) changes its state to update the pixel value to be written into the memory location when the stylus receives packets within the area of the color options

Erase:

- The stylus switches to erase mode when users tap on the ERASE option displayed on the screen
- The size of the eraser can be adjusted using buttons on the stylus, and the thickness can be increased or decreased based on the user's preference

Clear Screen:

- Users can clear the entire sketching area quickly by tapping on the CLEAR option located in the top-right corner of the screen
- The corresponding pixel locations within the sketch area are written with black color, effectively clearing the drawing area, providing users with a fresh start for new artwork.



Challenges

- Interfacing with the Wacom DTF 150
  - Proprietary device without supporting Linux Drivers.
    - Modifications to drivers for existing devices worked on personal Linux but not on the DE1-SOC image.
    - LIBUSB alone was incompatible - incorporated hidapi to communicate with the device
  - Packet Interpretation
    - Lack of existing documentation meant we had to backtrace the packet data to interpret what it meant
- Creation of a frame buffer
  - Due to the nature of a drawing tablet, a frame buffer had to be utilized such that we could specify in real time the entire frame to be displayed
  - Memory constraints came into the picture
  - The VGA peripheral had to be interfaced with the block ram
- Software Setup
  - Storage of the drawings created back to the HPS

Contributions

- Ajay Vanamali
  - Control Logic
  - VGA Peripheral
  - BRAM Design and Read/Write
- Bhoomi Shah
  - Driver Research
  - Wacom Device Research
- Manish Shankar
  - User Interface Design
  - BRAM Design
  - USB Packet Analysis
- Rahul Shanbhag
  - Wacom Device Driver using HID-API
  - VGA Kernel Module
  - SketchMaster User Space Logic