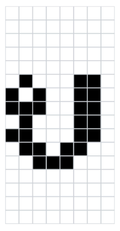
Hardware Synthesis, Term Project

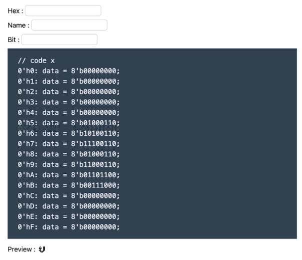
Our “Simple billboard/Terminal” Functionality

In this project, we are asked to design a simple billboard system in FPGA. The interface is our VGA, and a USART input, which will function as both receiver/display and sender at the same time.

- As a receiver, we accept input from UART from another board.and as a transmitter, The VGA display with at 6 lines (with 40 characters per line)
- The display is capable of displaying alphanumeric characters (ASCII). At minimum, the characters are A-Z, a-z, 0-9 and new lines. Other non-supported characters may show as “ ”.
- As a sender Use 8 switches as an input with one push button as a send command to send an 8-bit data through serial communication.
- We support laptop keyboard input 8-bit data through serial communication.
- We use 640x480 display resolution with a 12-bit color system.
- We have a “**reset**” switch for resetting the cursor position at **btnC**.
- We can use 8-bit input (using 8 switches) and one send button (**btnU** use as push/send button)
- We support the PS2 External keyboard displaying input through its VGA.

We support the Thai language by receiving 3-byte data from the Keyboard I/O in UTF-8 format. We use 16-bit data to store 2 bytes, as the first byte is not necessary when handling only one language.

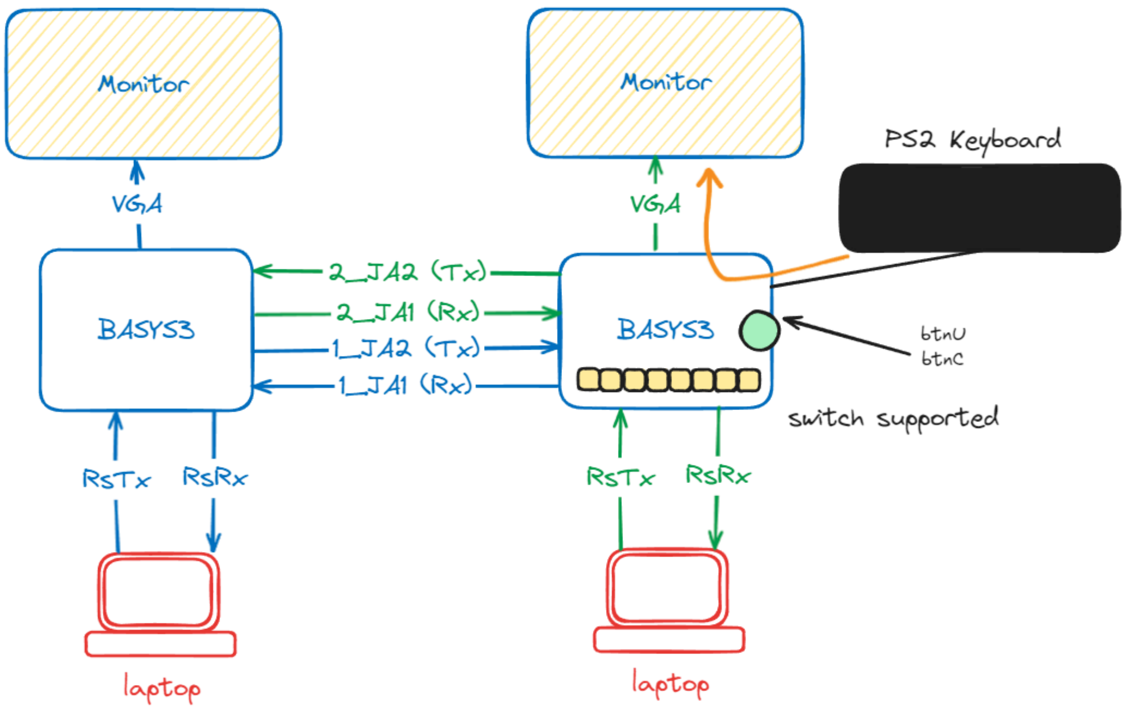


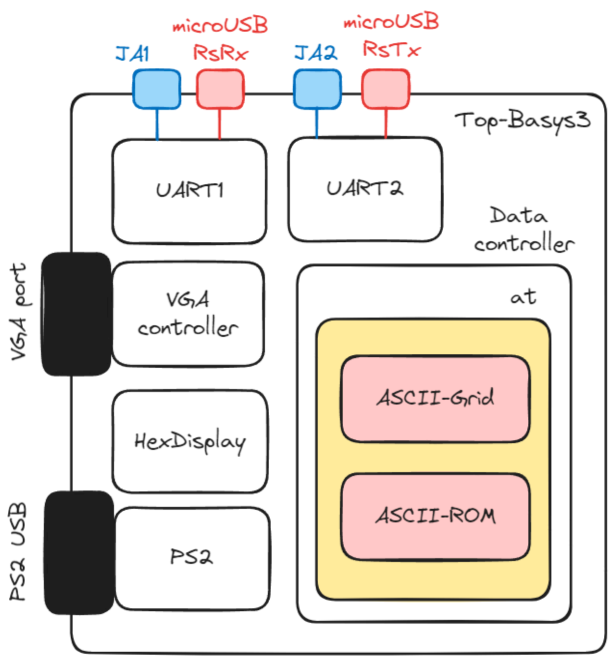

Hex :
Name :
Bit :

```
// code x
0'h0: data = 8'b00000000;
0'h1: data = 8'b00000000;
0'h2: data = 8'b00000000;
0'h3: data = 8'b00000000;
0'h4: data = 8'b00000000;
0'h5: data = 8'b01000110;
0'h6: data = 8'b10100110;
0'h7: data = 8'b11100110;
0'h8: data = 8'b01000110;
0'h9: data = 8'b11000110;
0'hA: data = 8'b01101100;
0'hB: data = 8'b00111000;
0'hC: data = 8'b00000000;
0'hD: data = 8'b00000000;
0'hE: data = 8'b00000000;
0'hF: data = 8'b00000000;
```

Preview : ป

We also use our own tools to create fonts, such as Thai fonts, because it is faster than manually guessing binary values. I chose this method instead of auto-generating fonts because we dislike machine learning, as it cannot produce beautiful fonts that align with our conventions.





In the following diagram, our Basys3 board has two UART modules for receiving and transmitting data to the corresponding target.

To display the input received by the receiver, data is transferred from the UART module to the ASCII_Grid module, which maps the data to variables that store the 16-bit code for each character. Then, the ASCII_Test module renders the text by mapping each line of data from the ASCII_ROM. Finally, the VGA Controller paints the screen using RGB signals generated by the ASCII_Test module from the ASCII_ROM.

Github source code