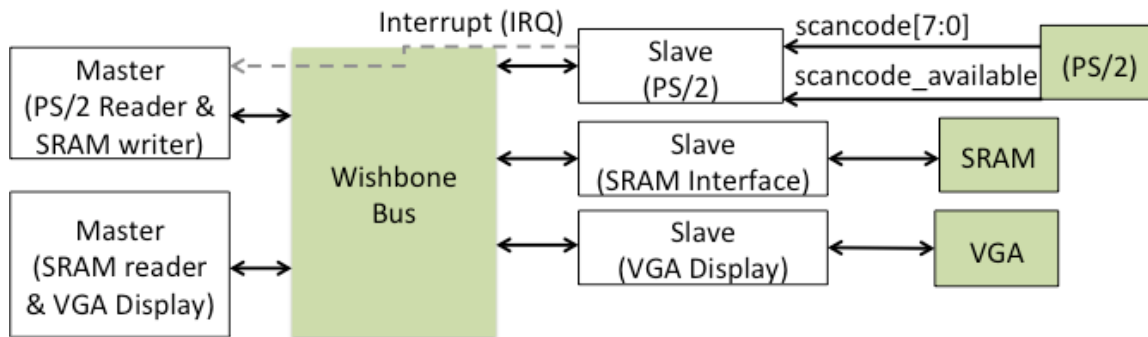


ECE 4401 (Fall 2014)

Lab 8 – VGA Interface & Text VGA Display

The goal of this lab is to create a system that can write text to the screen. You will combine your writer module from lab5 with the master module from lab6 to create a new master module that will take input from the keyboard and then write the appropriate bits to memory. The text will be in a 80x40 grid, so each character is 8 pixels wide and 12 pixels high. A `char8x12_lookup_table` module is provided to you. This module takes an 8-bit ASCII character encoding and a line number ranging from 0 to 11. The line number identifies the 8 pixels in the corresponding row. The module outputs 8 bits corresponding to the line in the character. The 8 bits are monochrome, so you need to AND this with a color field so that you can write a full 32-bits to memory. Note that the 8x12 character will need 12 lookups to the `char8x12_lookup_table`.



The VGA controller with a Wishbone interface is provided to you. The SRAM interface is the same as from lab5 and the PS/2 Keyboard interface is the same as from lab6.

Some things to be aware of:

- Every time you press a character, you will need to write 12 words to memory – one word for each line in the character.
- You can calculate the memory address for the lines of a character as follows:

$$\text{Addr} \leq (80 * \text{line}) * 4;$$

where, 80 is the offset to index the 80x40 grid for a single character. The line ranges from 0 to 11

- You will need to initialize the SRAM to background color when you reset the board. Otherwise, you will end up displaying random data from the SRAM on reset.

Modules you have to design:

- **wb_kb_read_sram_write**: This module does the following:
 - Reads the scan code from the keyboard slave and converts it to its ASCII value
 - Pass this ASCII value to the char8x12_lookup_table module and gets the output (the lookup table takes 1 cycle to generate the result, so, you need to handle this properly in your state machine)
 - Writes this output to the SRAM at the Addr calculated by the formula provided previously
 - If the input character is a backspace, fill the memory locations with the background color (which equates to erasing the previous character)
- **wb_vga640x480**: This module builds on the lab 8 demo VGA module. It performs the following actions:
 - At reset, cyc_o should de-asserted to give a chance to the other module to take the bus
 - Waits for the bus to be available and reads the 12 words representing the character from the memory and stores them locally in a register.
 - Releases the bus for 1 cycle at the end of capturing the new character. This allows the other module to take the bus and wait for an interrupt from the keyboard
 - In parallel, the VGA is displayed at 60Hz. The output color (red, green and blue) for the character is driven by the register. The rest of the screen is always configured to display the background color.

Bonus (50%!!!):

Implement a word processor. You basically have to extend the lab to be able to display 3200 characters. You will have to keep track the current location in the 80x40 character grid. You will start filling the screen from top left and fill the row (80 characters). Once you are at the end of the row, the next character input will move to the first character of the next line. At reaching the end of screen, you can start over-writing the 1st character on the screen (top left).

A backspace should move back the current character counter and clear the location. If we are at the 1st character of a line, a backspace will move the current location to the last character of the previous line and clear it.

Note that your design must manage 3200 characters being displayed continuously without glitches.