Lab3 Graphic Processor Microprogramming

Kent Palmkvist, ES, ISY, LiU

v0.5

120912

1 Introduction

The VGA lab design can be expanded to not only present a picture, but also modify the existing picture in the memory. The extension consists of a datapath used to read pixels, modify them, and write them back to memory. This should be done at the same time as the VGA screen image is presented.

The picture modification will consist of inverting the color of the image in a rectangle on the screen. The upper left coordinate and the size of the rectangle is entered through the switches and pushbuttons of the DE2-115 board.

The SRAM control unit is extended to support the VGA to continue presenting the image from the SRAM, and at the same time allow the rectangle generator access to the memory. The VGA have priority to the SRAM, and only allows the rectangle unit access during the blanking periods.

The read pixels from SRAM can be inverted by the rectangle generator, and written back to SRAM. The address to the SRAM is generated by the rectangle generator through combination of a loadable horizontal position counter and a vertical position counter. These counters can be loaded or incremented individually. There are also two counters available that can be loaded and decremented. When the counter value reaches zero is a flag set, which can be tested by the microprogram. The address is calculated as verticalpos* linelength + horizontalpos. Both pixels stored in a SRAM word is read at the same time, and both are written at the same time.

The rectangle generator is also able to read switches 17 to 10. and store that value in one of 4 registers. These registers can be loaded and read from. The value of the registers are shown on the 7-segment display.

The control of these registers and counters are handled by a microprogrammed controller structure. It consists of a microcode memory, which is addressed from a loadable counter. The counter can be loaded by bits extracted from the microcode memory, and may be controlled by a external input, creating conditional branching.

The current microprogram address is displayed on the green LEDs on the DE2-115 board.

2 Task definition

The rectangle generator shall read SW17-10 when key3 is pressed, and displayed on the 7-segment display. Switches SW0 and SW1 determines which two 7-segment display the value should be presented on. The four possible combinations of SW1 SW0 should correspond to "00": X coordinate of upper left corner, "01": Y coordinate of upper left corner, "10": horizontal size, "11": vertical size. Remember that horizontal coordinates will work in steps of 2 pixels.

When key2 is pressed, the picture should change, where the color of each pixel should be inverted in the rectangle defined by the coordinates shown on the7-segment displays. The design should then wait for key2 to be released before checking key2 and key3 again.

3 Copy of template design

The programming should be done on an existing design. The design contains everything except the correct memory contents of the microprogram ROM. Follow the steps below to create your own editable copy:

1. Load the TSTE12 module and start mentorskal. In the mentorskal window, start the lab tool using tste12lab
2. Open the library containing the design to modify:
   File->new->library.
   Select protected library, next
   root directory /site/edu/es/TSTE12/material, Library name TSTE12_lab3_DE2_115,
   next, finish
3. Create a new library to place yor copy in:
   File->new->library
   regular, next,
   root directory set to directory in the lab directory, name lab3_microprog, next, finish
4. Copy design units from TSTE12_lab3_DE2_115 into lab3_microprog:
   In TSTE12_lab3_DE2_115, select all design units, Edit->copy,
   rightclick on lab3_microprog->paste

The files have now been copied to the lab directory. Open the lab3_microprog library. It should now contain 9 blocks and 2 components.

4 Design structure

The available design is based on the DE2-115 board, and uses the default picture stored in SRAM.

The top module is called VGA_GPU_DE2_115, and contains the complete design as shown in Figure 1. The SRAM control block read data from SRAM and feeds it to the VGA Controller, that outputs an image on a VGA screen. The SRAM control block also supplies the Rectangle Generator with SRAM access whenever the VGA Controller is not needing SRAM access. The Rectangle generator will respond to key presses on the DE2-115 board, and modify the contents of the SRAM image.

4.1 Inputs and outputs

The VGA connects to the DE2-115 onboard Video DAC. See Lab 2 VGA controller notes for more details on this.

The design uses a 50 MHz system clock. This clock is used by all clocked units in the design.

There is a reset input activated by the KEY0 key on the DE2-115. Resetting the design will clear the rectangle settings and microprogrammed controller. It will not restore the image. The simplest way to restore the image is to turn the power of the DE2-115 board off and on.

The switches SW17 – SW10 describes the individual values that are entered as position or size information. The 8 switches allows any value between 0 and 255 to be entered.

The SW1 switch should select between the posistion and size registers. The SW0 switch should select between the horizontal and vertical settings.

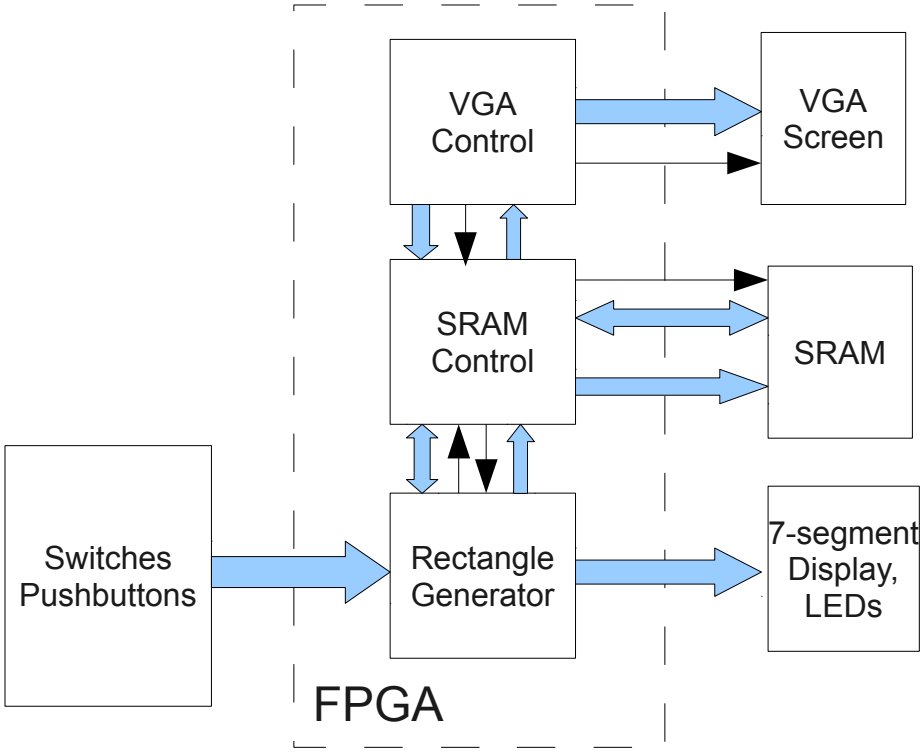

Figure 1: Complete design overview

The 7-segment display should be used to present the selected rectangle position and size. The number stored in the corresponding register is presented in hexadecimal form (00 – FF).

The key KEY3 should trigger the controller to update one of the registers storing position and size. This key returns a '0' when pressed.

The key KEY2 should trigger the change of the image according to the rectangle definition stored in the corresponding registers.

The green LEDs (LED4 to LED0) indicates the current Microprogram ROM address, and is thus useful to help debug your design.

4.2 General Structure

The Rectangle Generator uses RECT_RD and RECT_WR signals to tell the SRAM controller that it needs SRAM access. The SRAM Controller tells the Rectangle Generator that the SRAM accesses have completed by use of a RD_ACK signal.

The Rectangle Generator contains a datapath for coordinate and size information, as shown in Figure 2. It can read the value from the switches SW17 – SW10 or values stored in the display registers. It can write values into the display registers and into the loadable counters.

The SRAM address is generated by combining the values from the vertical position counter and the horizontal position counter. These counters can be loaded with a start value, and incremented.

The size counters are down counters, that when loaded with a value then will decrement. There is a flag from the two down counters that indicates when the counter reaches zero.

The Rectangle Generator only supports one type of operation on the pixel values read from SRAM, inverting the sample bit pattern. This datapath is shown in Figure 3 below.

Figure 2: Pixel position data path

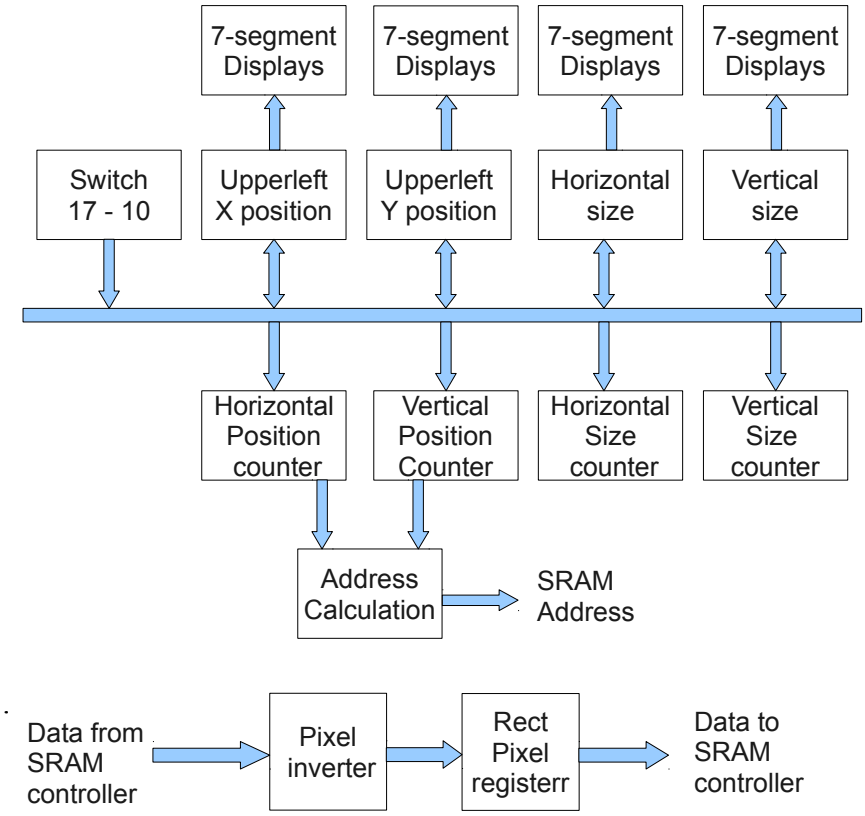

Figure 3: Pixel value modification data path

4.3 Microprogrammed control structure

The microprogram control structure consists of a microprogram address generator, a microprogram ROM, and a microcode decoder, as shown in figure 4. The address generator consists of a condition selection multiplexer, and a loadable counter. If the condition is true, the counter is loaded by a value stored in the microcode instead of being incremented by 1. The microcode decoder simply renames the ROM outputs to suitable signal names to make the design more readable.

Th Flags and Input signals consists both of switches and keys on the board, as well as constants and status signals from counters. See table below for more detail.

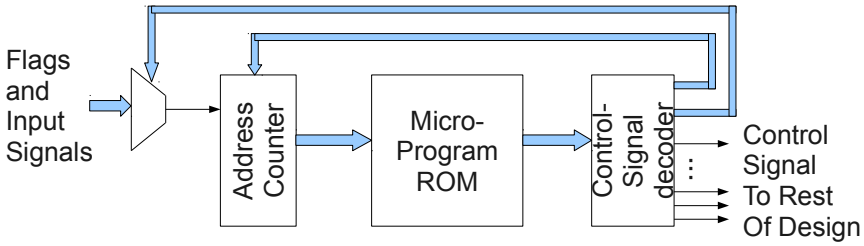

Figure 4: Microprogrammed control structure

The microcode ROM consists of 32 words of 29 bits, and its contents and behavior is described in the microprogram block. The bits are defined as follows

28	StoreRectData	Store the inverted value of the latest data read from SRAM
27	ReadSW17_10	Read the settings of the switch 17 downto 10
26	Rect-RD	Start a read from the SRAM

25	Rect-WR	Start a write of the Rectangle data register into SRAM
24	NextHorPos	Increment horizontal position counter
23	NextVertPos	Increment vertical position counter
22	DecHorCnt	Decrement horizontal counter
21	DecVertCnt	Decrement vertical counter
20	SetHorPos	Set the value of the horizontal position counter
19	SetVertPos	Set the value of the vertical position counter
18	SetHorCnt	Set the value of the horizontal counter
17	SetVertCnt	Set the value of the vertical counter
16	LoadUpperLeftX	Set the display showing upper left X value
15	LoadUpperLeftY	Set the display showing upper left Y value
14	LoadSizeX	Set the display showing the horizontal size
13	LoadSizeY	Set the display showing the vertical size
12	ReadUpperLeftX	Read the value from the upper left X display register
11	ReadUpperLeftY	Read the value from the upper left Y display register
10	ReadSizeX	Read the value from the horizontal size display register
9	ReadSizeY	Read the value from the vertical size display register
8-5	condition select.	
		0000 Always false => no jump
		0001 Always true => always jump
		0010 horizont size count zero
		0011 vertical_count size zero
		0100 SW0
		0101 SW1
		0110 Memory access not complete (complete => '0' for 1 clock cycle)
		0111 KEY2 (pressed => '0')
		1000 KEY3 (pressed => '0')
4-0	jump address	loaded into the microprogram address counter if the condition is true (='1')

4.4 Memory access protocol

The read and write of the SRAM may take long time, as the VGA generation has priority over the rectangle generator. Special care needs tobe taken to avoid problems with this.

The Rectangle block should indicate the request to access the memory by setting the Rect_WR or Rect_RD to '1' during one control step. That signal should then be returned to '0', and the Rectangle generator controller must then wait for the RD_ACK signal (condition number 0110), that will be activated during one clock cycle once the SRAM controller hav completed the SRAM access. The check for the memory acknowledgement must be ched in every clock cycle, that is, the check must jump back to itself (the current microprogram address) if the memory access has not completed.