Lab 7: Digital Photo Frame

Submission Due Dates:

Source Code: 2020/12/08 18:30

Report: 2020/12/13 23:59

Objective

1 Getting familiar with finite state machines (FSMs) in Verilog.

2 Getting familiar with the control of VGA Display and other I/Os on the FPGA demo-board.

Action Items

In this lab, you will implement a digital photo frame with fancy transition effects on the VGA display. Pick an image you like. Use **PicTrans.exe** to generate your own .coe file**. (Make sure it is an appropriate image)**.

1 VGA controller

A. lab7_1.v (60%)

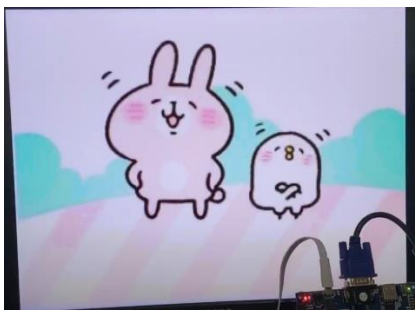
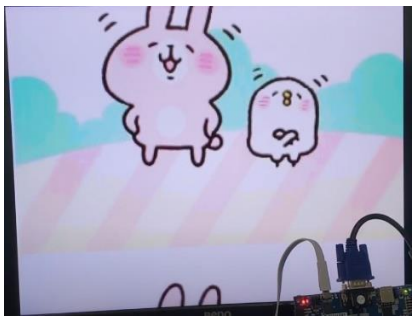
Design a VGA controller that makes your image scrolls up or down.

a. IO list:
   - ✓ Inputs: clk, rst, en, dir
   - ✓ Output: vgaRed, vgaGreen, vgaBlue, hsync, vsync

b. **rst**: the positive-edge-triggered reset, the VGA display will show the image at the origin position after triggered.

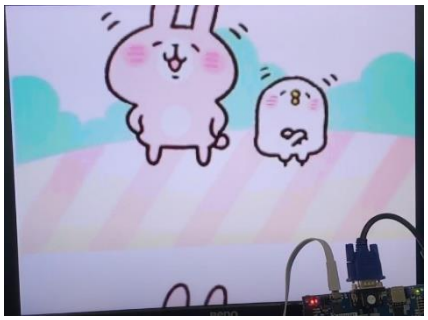
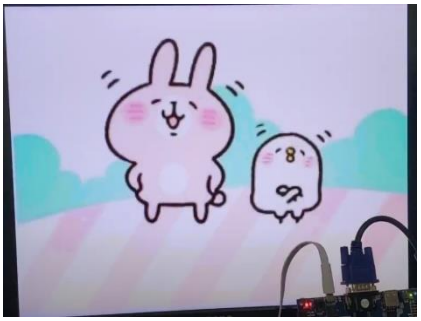
c. If **en** == 1'b0: hold the image on the screen unchanged.

d. If **en** == 1'b1:
   - ✓ If **dir**== 1'b0:
     - the image scrolls up at the frequency of 100MHz divided by 2^{22}.


   - ✓ I If **dir**== 1'b1:

- the image scrolls down at the frequency of 100MHz divided by 2^{22}.

✓ Refer to the demo video: **https://youtu.be/chPyVra7fyY**

e. You have to use the following template for your design:

```
module lab7_1 (
  input clk,
  input rst,
  input en,
  input dir,
  output [3:0] vgaRed,
  output [3:0] vgaGreen,
  output [3:0] vgaBlue,
  output hsync,
  output vsync
  );
  // add your design here
endmodule
```

IO Connection:

clk	connected to W5
rst	connected to U18 (btnC)
dir	connected to V16
en	connected to V17
vgaRed	connected to pin N19, J19, H19, G19
vgaGreen	connected to pin J18, K18, L18, N18
vgaBlue	connected to pin D17, G17, H17, J17
hsync	connected to pin P19
vsync	connected to pin R19

B. lab7_2.v (40%)

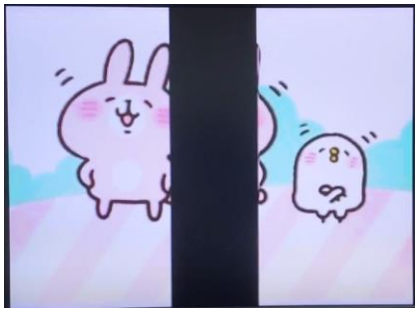
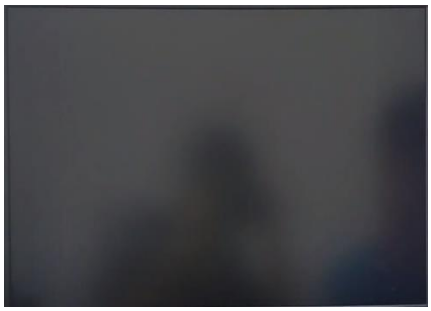
Design a VGA controller to operate as follows.

a. IO list:
   - ✓ Inputs: clk, rst, split
   - ✓ Output: vgaRed, vgaGreen, vgaBlue, hsync, vsync

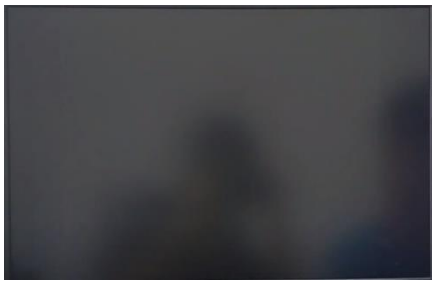
b. **rst**: the positive-edge-triggered reset, the VGA display will show the image at the origin position after being reset.

c. If **split** == 1'b0: hold the image on the screen unchanged.

d. If **split** == 1'b1:
   - ✓ When the button **split** is pressed, the image is split into the left half and right half.
   - ✓ The left part slides toward the left direction. The right part slides toward the right.

   - ✓ After the two parts disappear entirely, the top half of the original image will slide from the top of the screen toward the center of the screen after one cycle; The bottom half will slide from the bottom of the screen toward the center simultaneously; in the end, they merge into the origin image.
   - ✓ After they merge into the original image, the image will start split again, keep looping these two operations (split and merge) until pressing the **rst**.



Refer to the demo video: **https://youtu.be/q_32RRG_PPk**

e. When the VGA controller operates in the split mode, only the **rst** button is effective.

f. You have to use the following template for your design:

```
module lab7_2 (
  input clk,
  input rst,
  input split,
  output [3:0] vgaRed,
  output [3:0] vgaGreen,
  output [3:0] vgaBlue,
  output hsync,
  output vsync
  );
  // add your design here
endmodule
```

IO Connection:

clk	connected to W5
rst	connected to U18 (btnC)
split	connected to W19 (btnL)
vgaRed	connected to pin N19, J19, H19, G19
vgaGreen	connected to pin J18, K18, L18, N18
vgaBlue	connected to pin D17, G17, H17, J17
hsync	connected to pin P19
vsync	connected to pin R19

Attention

- ✓ DO NOT copy-and-paste code segments from the PDF materials. Occasionally, it will also paste invisible non-ASCII characters and lead to hard-to-debug syntax errors.
- ✓ You should hand in the file named **lab7_1.v** and **lab7_2.v**. If you create several modules for your design, merge them all into one Verilog file. Upload each source file directly! DO NOT hand in a compressed ZIP file!
- ✓ You should also hand in your report as **lab7_report_StudentID.pdf** (i.e., lab7_report_108456789.pdf).
- ✓ You should be able to answer questions of this lab from TA during the demo.
- ✓ You need to generate bitstream before the demo.