**ECE324 Lab 9: PacMan**

Name(s):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| **Exercise** | **Course outcome** | **Grade** |
| Lab9 Demo | 2.a, 2.d, 5.c, 7.b | /15 |
| Lab9 Report | 2.a, 5.c, 7.b | /25 |
|  | **TOTAL:** | /40 |

2.a. Define engineering problems from specified needs for digital systems including implementation on FPGAs using HDL programming.

2.d. Produce FPGA designs that meet specified needs.

5.c. Collaborate with individuals with diverse backgrounds, skills and perspectives.

7.b. Employ appropriate learning strategies such as communicating with an expert, using external resources, experimentation, simulation, etc.

# Learning objectives

1. Understand video graphics array (VGA) concepts, architecture, implementation, and timing.
2. Understand how tiles and sprites work.
3. Using SystemVerilog HDL, make modifications to a moderately complex design of a VGA driver circuit, use Xilinx Vivado to implement the circuit in an FPGA chip, download the implemented design to a Nexys4DDR board, and display the results on a video monitor.

# Procedure Part A

In Vivado, add design sources free\_run\_shift\_reg.sv and mod\_m\_counter.sv, which came from Pong Chu’s SystemVerilog book and have been used in previous labs. Also add new design sources Lab9\_PacMan.sv, spriteMotion.sv. Also add videoClk108MHz.v (you’ll learn in lab10 how to generate this Verilog file to produce the 108 MHz video clock, but for now you can just use it without understanding it). Add constraint file Lab9\_PacMan.xdc. While you don’t need to add the memory initialization files PacManTileMapRom.txt, PacManTileSet.txt, PacManSpriteSet.txt, and GhostSpriteSet.txt, make sure they’re in the same directory as your Xilinx project file.

Synthesize without any changes. While waiting, bring down WSU’s sxga (1280x1024) monitor from the top lab shelf (please ask for help if you’re uncomfortable bringing it down or putting it back up). Connect the monitor’s VGA cable to the VGA connector near the top right of the Nexys4DDR board.

This lab emulates a subset of the PacMan game originally developed by Namco. Pushing the left red button on the Nexys4DDR board labeled “CPU Reset” initializes the board, and the four directional buttons control the movement of PacMan (pushing the left or right button starts the motion). You do not need to demonstrate this.

# Procedure Part B

It’s not unusual in industry to be given a large amount of code, and be asked to modify a small part of it, like in this lab. It’s typical that you won’t have enough time to understand all of the code you are leveraging (even though with enough time you could), so concentrate first on learning what is needed to make the following changes:

1. By modifying PacManTileMapRom.txt, make the tile map bilaterally symmetrical by changing from a small pellet to a large pellet at the appropriate location near the upper right corner.
2. In the center of the image is the ghost den, including two tiles at the center top forming a door. After studying Lab9\_PacMan.sv (specifically section Video Pixel Generation, subsection Draw Tiles), modify PacManTileSet.txt to make both door tiles the same color as the pellets (peach) instead of the same color as the walls (blue).
3. The tunnel that extends outside of the outermost walls of the left and right sides of the image was in the original PacMan game, but it wasn’t displayed. By modifying the last section of Lab9\_PacMan.sv, overlay black on the first two and last two tile columns. Note that the PacManTileSet columns and rows are each replicated four times to become the video columns and rows.
4. When Blinky (the red ghost) starts moving, it goes to the right, but in the original Pacman game it starts going to the left. Fix this in spriteMotion.sv.

Demonstrate all four changes to the professor or lab assistant.

# Deliverables

1. Demonstrate operation of your completed FPGA design to the instructor or lab assistant.
2. Write a brief (1-2 pages) lab report including the following items:
   1. Cover sheet with names, course number, assignment number, grade box and ABET outcomes.
   2. A written description of your SystemVerilog code: describe the overall function of your design and the operation of each module, including any salient details.
   3. Find and report the number of FPGA device logic resources used: how many BRAMs, LUTs, muxes, flip-flops, latches, clock buffers, and IO buffers. What is the utilization percentage of the device?
3. Document any SystemVerilog code you modified. Your SystemVerilog code ***must include header comments stating your names, date and class number.*** Any changes to the SystemVerilog code ***must also include comments explaining the operation of the code***.
4. Upload your written lab report ***in .pdf format***, upload all of the .sv and .xdf text files used for your solution (zipped together if you like), and then hit the submit button just once.

# References

Chu, Pong, *FPGA Prototyping by SystemVerilog Examples*, Wiley 2018, ISBN 9781119282662