Pong

Progress Report
Cole Adams, Ozwin Cordes, and Jonny Sylvain

We struggled with the clock being too fast and not visibly updating the location of the ball on the screen. The clock was so fast that the make_box component drawing the ball reset the ball to essentially the same location. To fix this we include a register that waits clock cycles to make the change easily visible.
Attached in the ZIP is the VGA_video folder which is our top level for the project.
In components/make_box.v the test bench for the make box component exists.
We have the Stage 2 outcome currently where the ball will move and bounce on the X-axis.
We have decided to leave our anticipated outcomes unchanged except the ambitious outcome:

Link to github with Verilog files: [alegomonkey/ECE275_pong: Pong on the FPGA]()

Outcomes:
(Minimal): A moving ball demo
(Stage 2): A bouncing ball demo (bounces with edges of screen)
(Stage 3): A ball that bounces off a paddle
(Stage 4): Control the paddle with button inputs
(Ambitious): The ball and paddles can loop through the boundaries of the screen

- Minimal
    - Inputs: Display Module, Ball Module
    - Outputs: Display Module outputs to VGA board based on Logic Module
    - Testing: Visually verify the ball moves on the VGA
- Stage 2:
    - Inputs: Display Module, Ball Module
    - Outputs: Ball Module outputs ball state to the Display Module which outputs to VGA screen.
    - Testing: Verify each possible state of the ball on the VGA screen
- Stage 3:
    - Inputs: Display Module, Ball Module, Paddle Module

    - Outputs: Paddle Module outputs state of the paddle to the display and ball modules. Ball module outputs state of the ball to the display module. Display module outputs to the VGA screen.
    - Testing: Verify each possible state of the ball module and paddle module on the VGA screen.
- Stage 4:
    - Inputs: Display Module, Ball Module, Paddle Module, Button[2:0]
    - Outputs: Buttons output their states to the paddle module. Paddle Module outputs state of the paddle to the display and ball modules. Ball module outputs state of the ball to the display module. Display module outputs to the VGA screen.
    - Testing: Verify each possible state of the buttons, ball module, and paddle module on the VGA screen.

- Stage 5 (Ambitious):
    - Inputs: Display Module, Ball Module, Paddle Module, Button[2:0]
    - Outputs: Buttons output their states to the paddle module. Paddle Module outputs state of the paddle to the display and ball modules. Ball module outputs state of the ball to the display module. Display module outputs to the VGA screen. The paddles and ball can move through the Y-axis freely.
    - Testing: Visually verify the logic of the Breakout module functions correctly. Visually verify each possible state of the buttons, ball module, and paddle module on the VGA screen.

Inputs

- Button[2:0]:
    - Used to control the paddle

Outputs

- VGA screen:
    - Used to display game state

Modules

- Display module (output): this module will inherit from the VGA driver code we first started using for the VGA demo in today’s lab. We will leverage this functionality to display rudimentary shapes corresponding in size and motion to the logical submodules implemented in the core.
- Logic Module (core): This module will consist of submodules for the ball, paddle, and hopefully brick submodules (if we reach ambitious level), and will handle the logical loop of gameplay for our Pong implementation.
- Controller module (input): This module will consist of code that handles two inputs (left and right) and outputs the desired functionality to the core module.

Testing

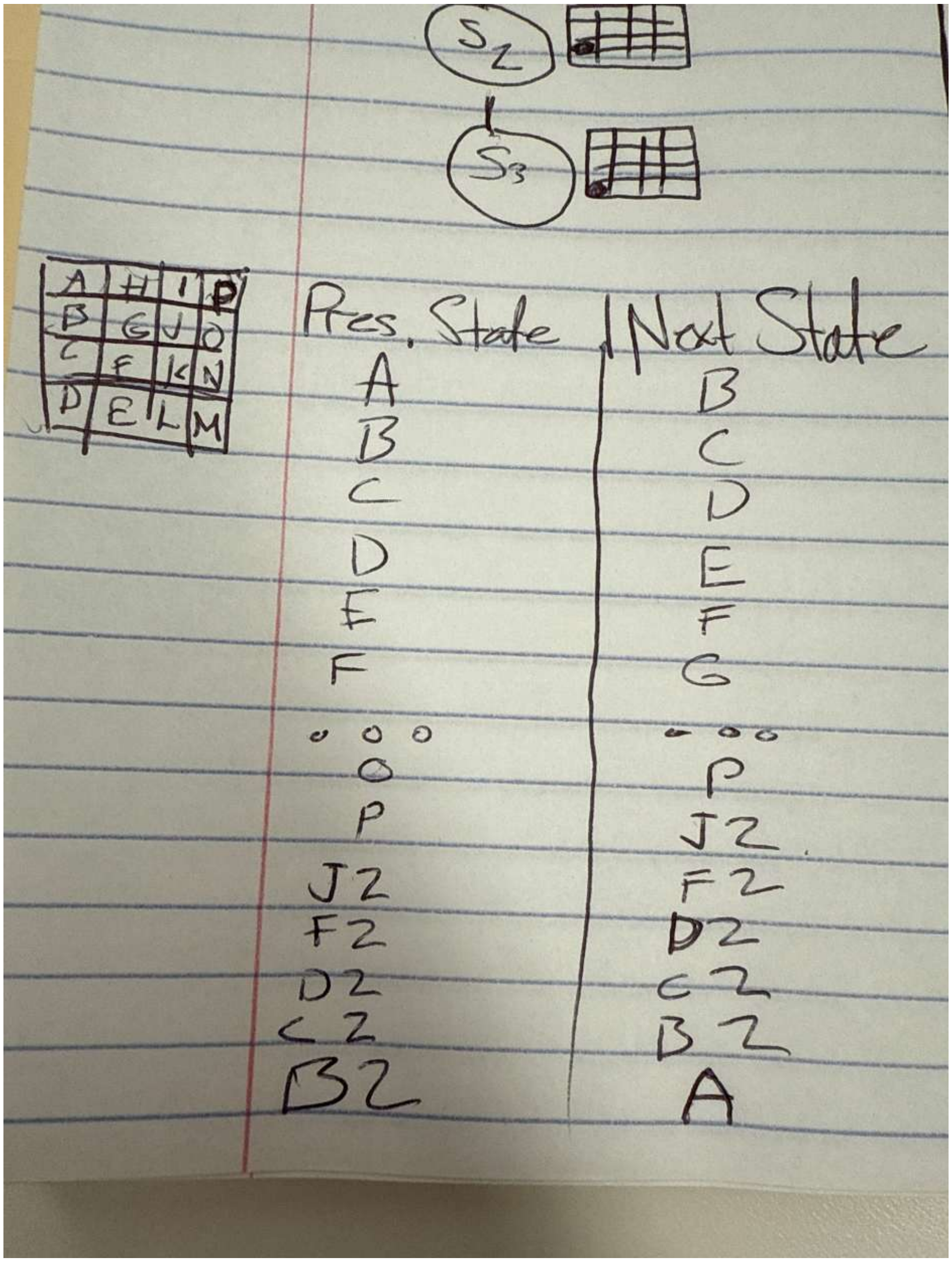
Test bench module for the box module has been created. Future test benches will be constructed as we construct each module.

Anticipated Timeline:

1. Research modules for game functionality (VGA driver etc), Begin basic implementation - 4/20
2. Complete rough draft (ideally first and second level of implementation) - 4/27
3. Complete as much functionality as we can (aiming for L4 or ambitious) - 5/1

State Transition Tables / Diagrams:

ST for ball bouncing around 4x4
pixel screen:

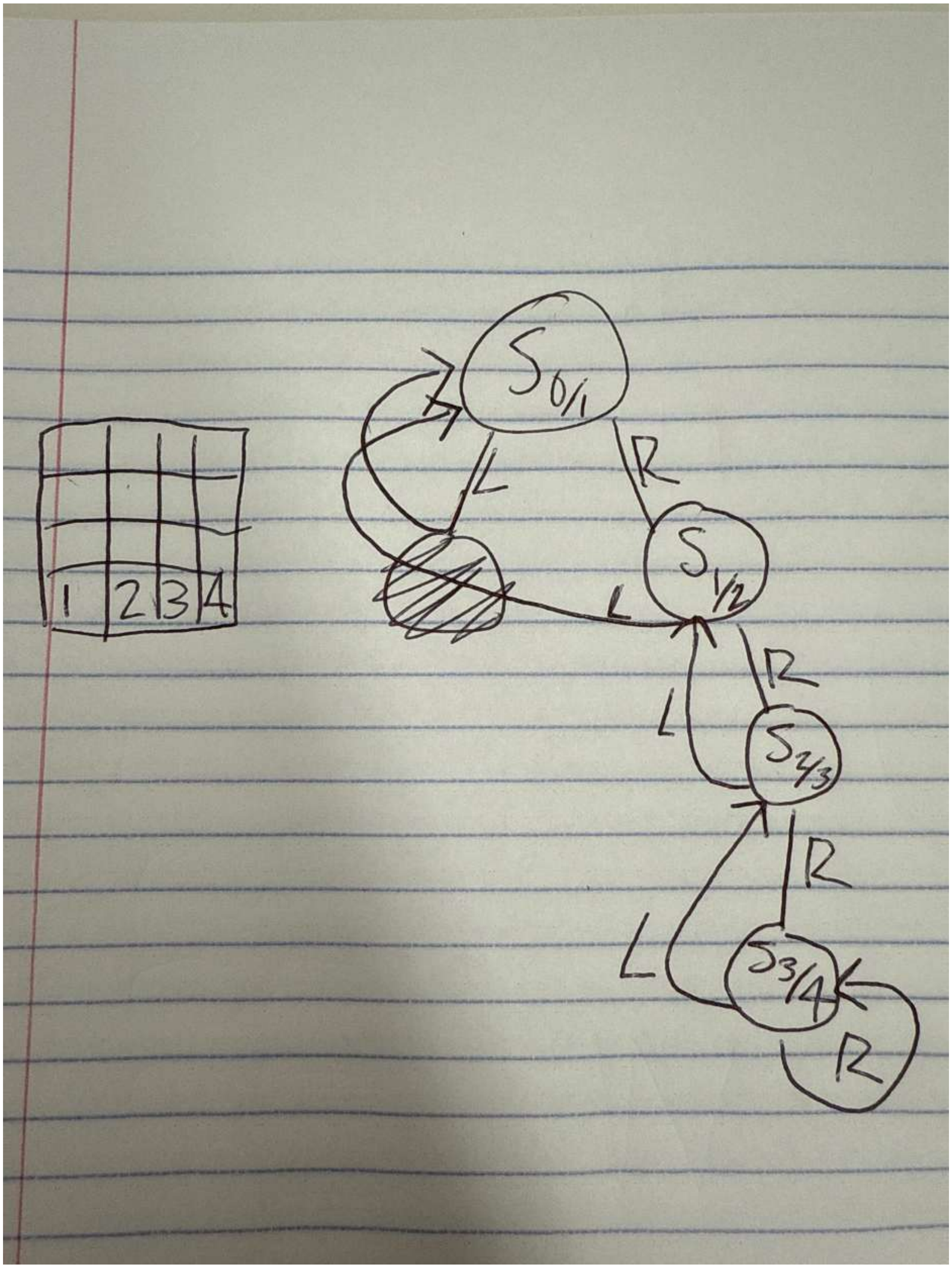


A	H	I	P
B	G	J	O
C	F	K	N
D	E	L	M

Pres. State	Next State
A	B
B	C
C	D
D	E
E	F
F	G
∘ ∘ ∘	∘ ∘ ∘
O	P
P	J2
J2	F2
F2	D2
D2	C2
C2	B2
B2	A

In this state diagram, the ball traverses every pixel of the screen, eventually returning to start(A).

SD for 1-pixel paddle on 4x4 pixel screen:



In this state diagram, the paddle can maneuver between the 4 available positions.