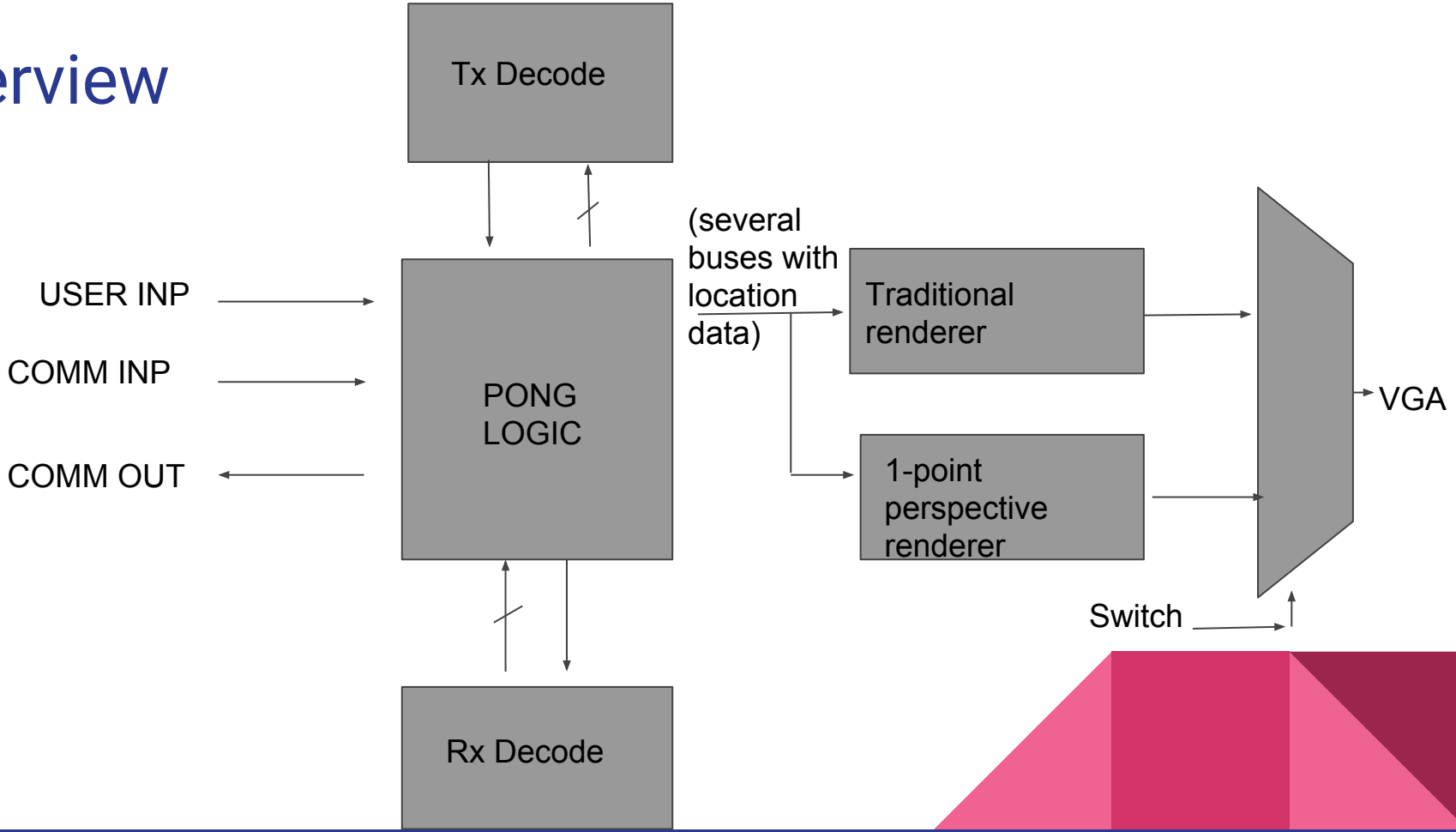
Pong: 2-player, 1st person.

William Navarre, Emmanuel Akinbo

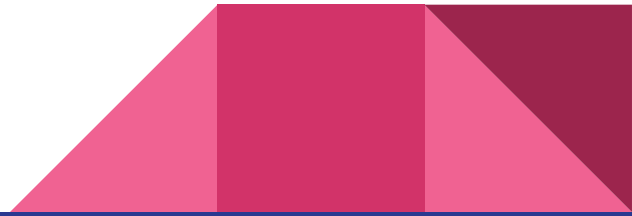
Overview


Tx Decode
USER INP
COMM INP
COMM OUT
PONG LOGIC
Rx Decode
(several buses with location data)
Traditional renderer
1-point perspective renderer
VGA
Switch

Pong Logic: Basics

- Update local paddle position each frame.
- Update position of (any) *incoming* ball each frame.
- Flip direction of the ball when appropriate, incorporating some notion of momentum (the ball will receive momentum from the paddle).

INPUTS:
- Reset.
- Up button.
- Down button.

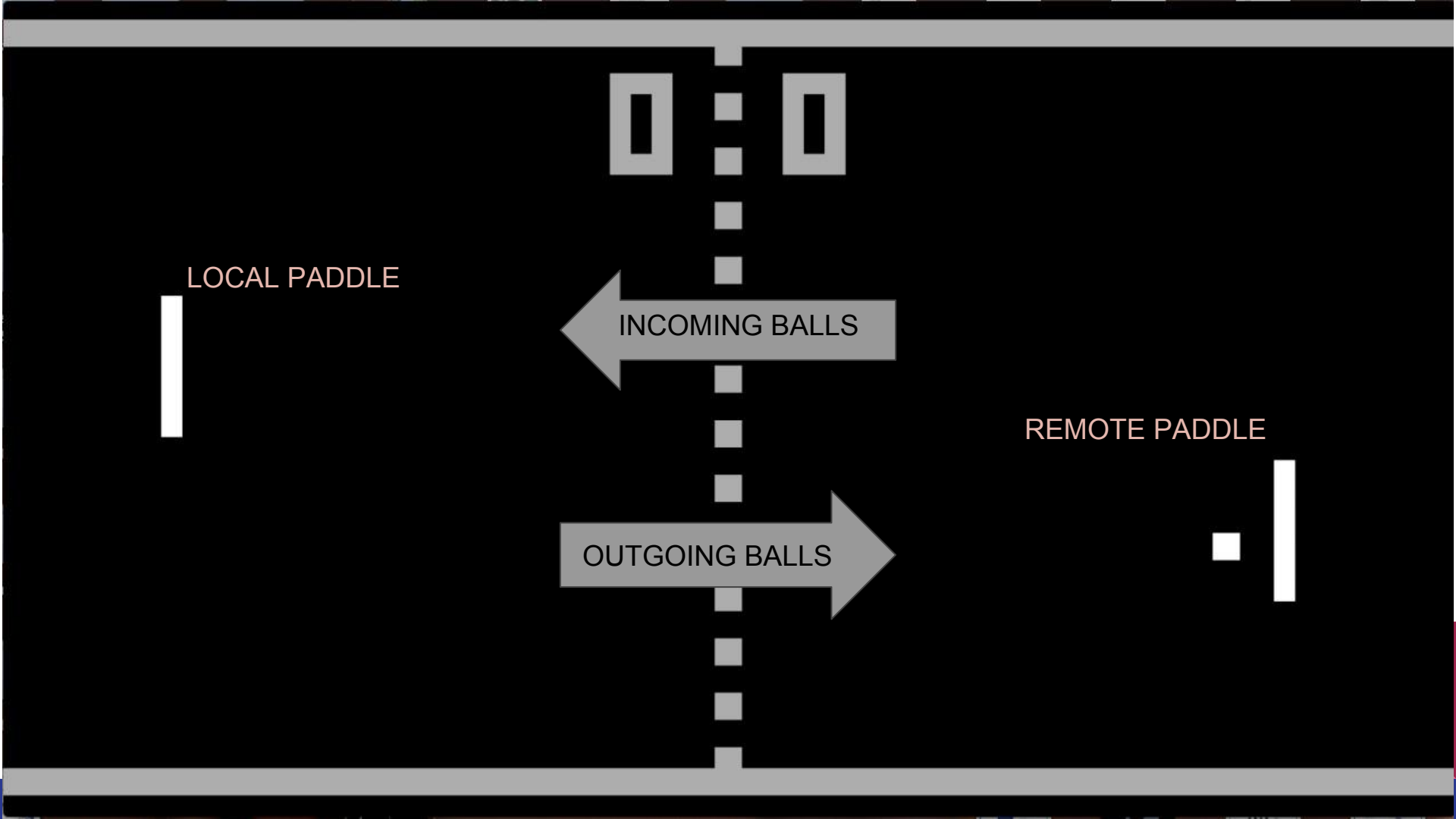


Pong Logic: 2-player system

At every frame, the FPGA will send, over the communication system:
-the position of the local paddle
-its belief on the position of the ball
-its belief on the direction the ball is moving (horizontal, vertical velocities)

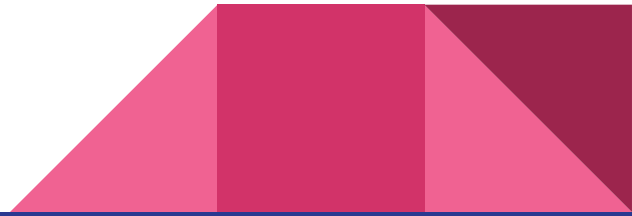
The position of the remote paddle is updated on each frame so it can be re-drawn in the correct place.

When the ball is going away from the local paddle, the other two parameters are updated in internal state to be displayed on the screen.


LOCAL PADDLE
INCOMING BALLS
REMOTE PADDLE
OUTGOING BALLS

Pong logic outputs

- Location of puck (x, y).
- Location of local paddle (y).
- Location of remote paddle (y).



Communication Modules (wired prototype)

- Shift registers detect preamble.
- Bits in word are transmitted in sequence (serialized voltages):

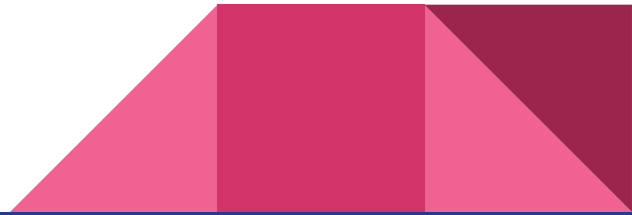
- The different messages numbers transmitted are transmitted with an identifying tag at its start.

Traditional renderer

The function is equivalent to lab 3.

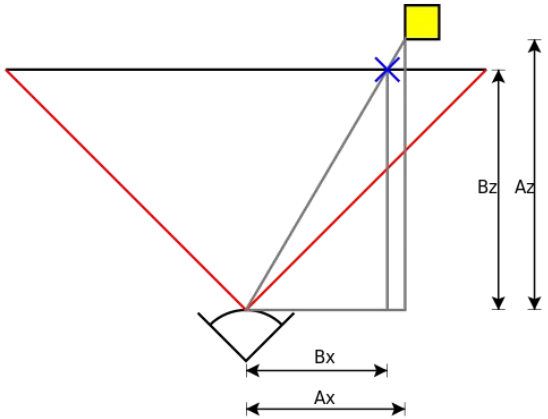
To make design easier, the 2D size of various objects are parameterized. The wired inputs are:

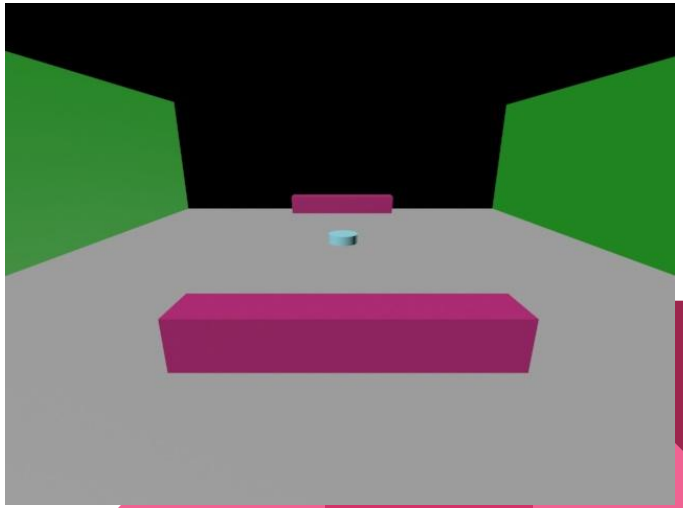
- Puck position.
- Local paddle position.
- Remote paddle position.

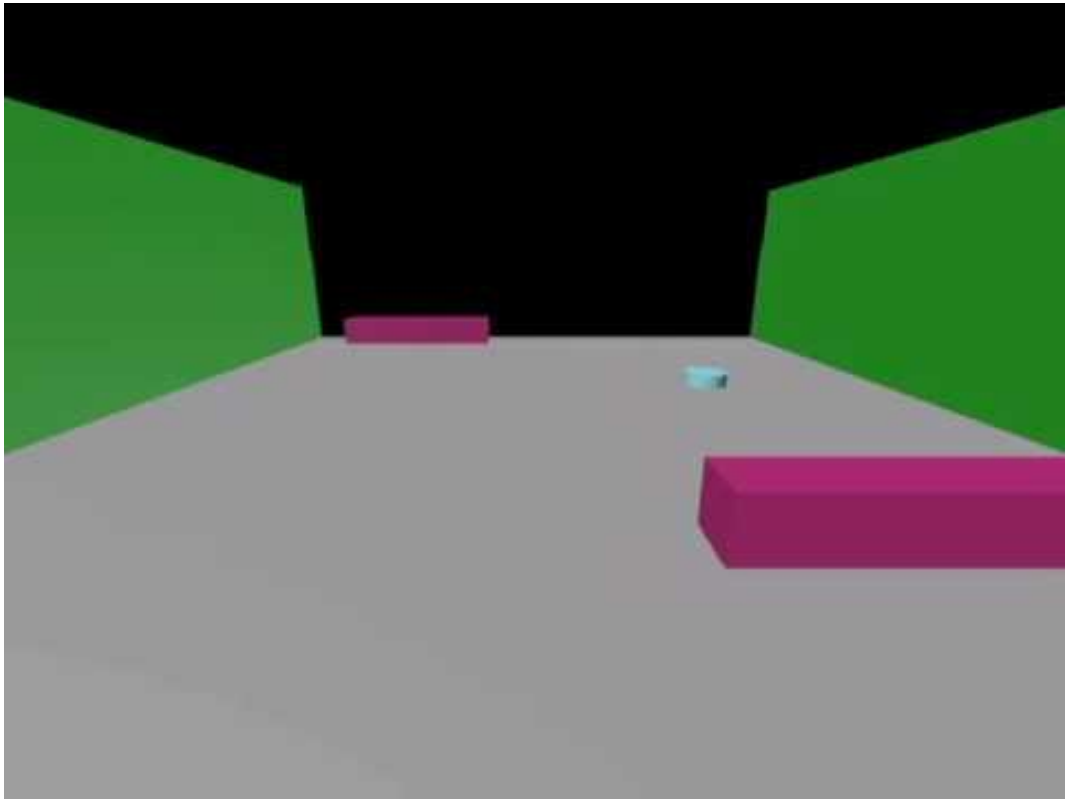


1-point perspective renderer

- 3D Projection mapping
  - Matrix transformations
- Base game in 3D map (use x and y from game logic, arbitrary z height)
- Pre-compute walls
- 1st person view





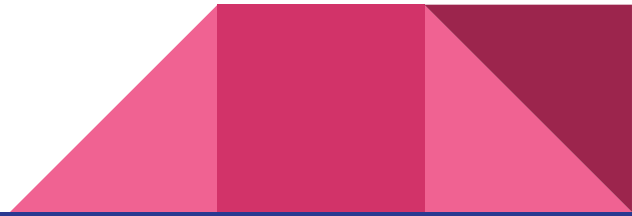


Data transmission and memory requirements

- ~65 bits per transferred per frame for location and other state information. TAGX|DATAXXXXXXX
- Game uses no significant rewritable memory.
  - Read-only lookup table for one-point perspective
- When we implement analog transmission, we will need some memory to buffer samples.

Stretch Goals

- Implement the communication channel with sound
  - Ultrasonic transmitter and receiver
- Add the option to have 2 pucks on screen
- Shifting perspective in 1st person mode



Timeline

Week of:

11/7 - Communication Prototype

11/14 - 3D module

11/21 - IR Communication

11/28 - Sound Communication module, Second puck

12/5 - Debugging

12/12 - Debugging