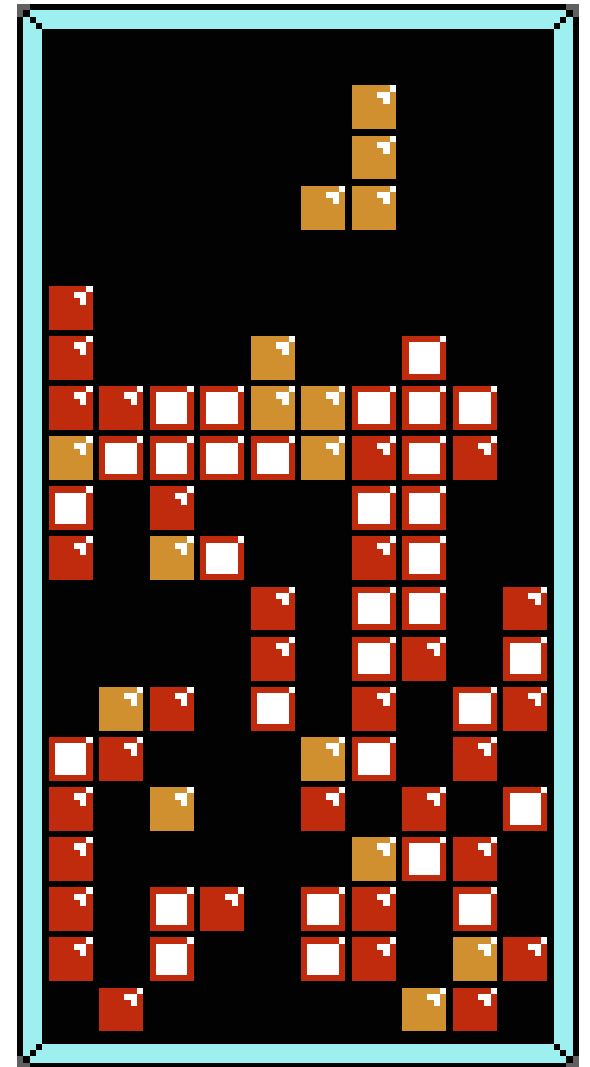
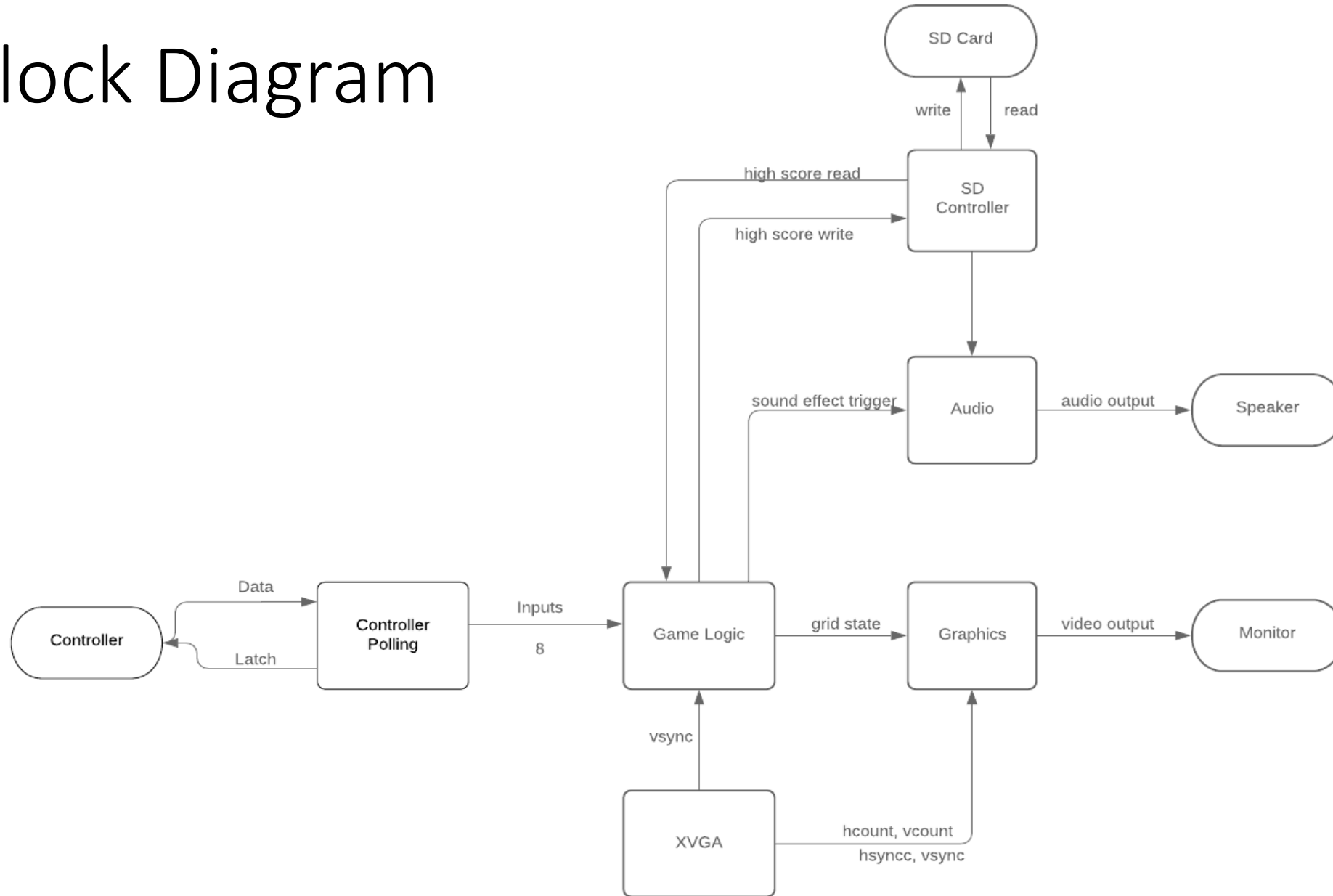


FPGA Tetris

Roberto Garcia



Block Diagram



Game Logic – Game State

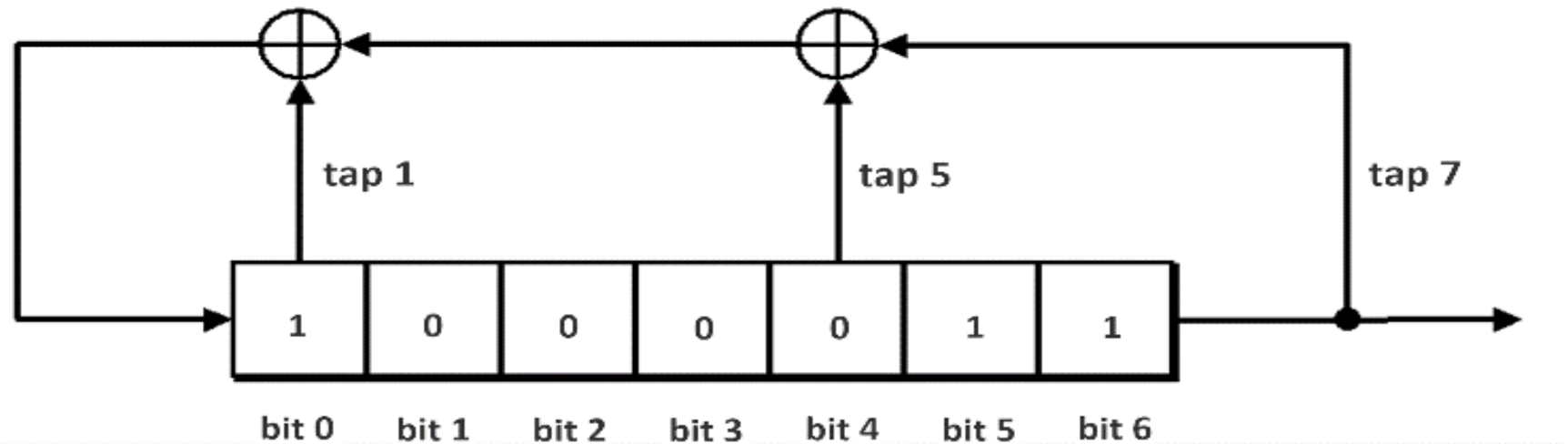
- A game state can be represented by the state of all currently placed squares, as well as the currently falling Tetromino
- The playing grid is comprised of 200 squares, each of which will be in a colored state or empty state
- Grid will transition states upon a falling Tetromino landing
- A falling Tetromino can be in 1 of 4 rotational orientation states, and any coordinate
 - Next state will be determined by rotation input and left/right input

Graphics

- Character Generation:
 - Identically sized characters A-Z, 0-9 stored in memory
 - Can be parsed together to display text and scores
- Playing grid generation:
 - Draw 200 unique squares
 - Each square will be colored by feeding in the corresponding color state and mapping it to an RGB value

Game Logic – Tetromino Generator

- There are 7 unique Tetrominoes
- Generate using a linear-feedback shift register, a pseudorandom number generator

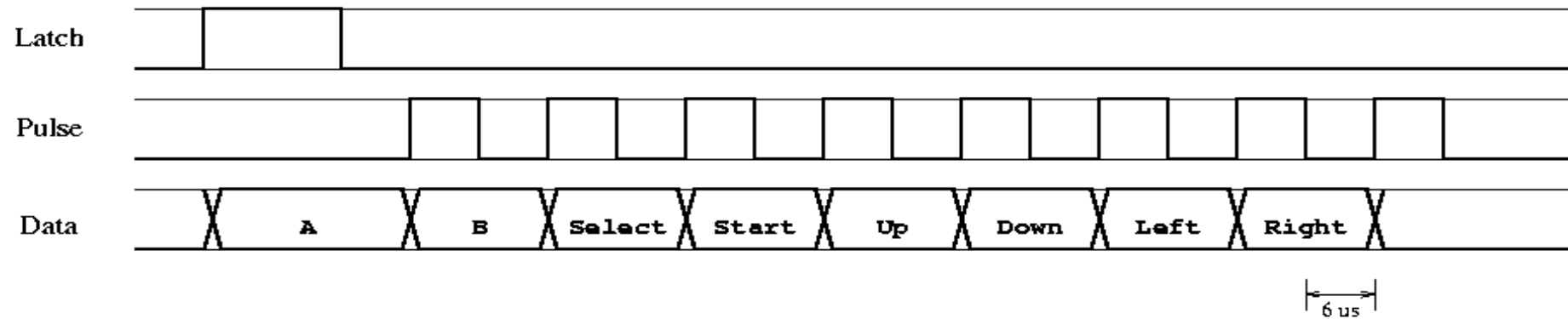


Game Logic – Tetromino Generator

- Problem
 - Initial seed results in same first game every play session
- Solution
 - Cycle through LFSR every clock cycle in background
 - Next Tetromino is a function of PRNG sequence and how much time has elapsed since last Tetromino generation

NES Controller Interface

- A 12 us Latch pulse commands the controller capture the current state of all buttons
- Button states are sent serially via 6us pulses



- Repeating every 120 us gives 138 samples/frame to debounce with

Audio

- A single theme song will be continuously playing on a loop
 - Stored in SD card
- Simple short-tone sound effects will be included
 - Higher priority, will momentarily pause music playback

SD card

- Interfaced with using the provided SD controller
- Music:
 - preload music data bits directly using software to avoid SD filesystem
 - Read only
- High Scores:
 - Can be stored in predetermined location anywhere on SD card that does not overlap with music
 - Reads and writes

Timeline

- 11/4: Display matrix of Tetrominoes, begin implementing core gameplay logic via FPGA button inputs
- 11/11: Finish implementing core gameplay logic, add text generation
- 11/18: Add NES controller interface, add sound effects
- 11/25: Implement music reading from SD card and high score saving/loading
- 12/2: Debugging