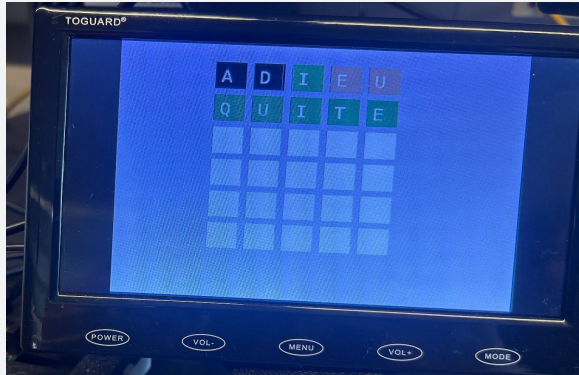


WORD GAME ENGINE!

By: Ben Li, Alice Hu

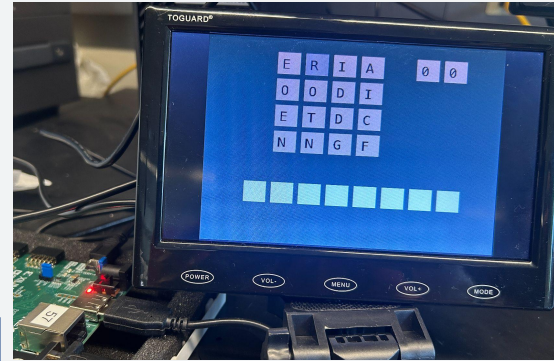
PROJECT OVERVIEW

What we built: a word game engine running on our 5-stage pipelined processor.
Input: PS/2 Keyboard. Output: VGA Display.



WORDLE

Guess a 5-letter word in 6 tries



WORD HUNT

Find connected words in a 4x4 grid

Wordle Implementation

1. Data Storage / Word Selection:

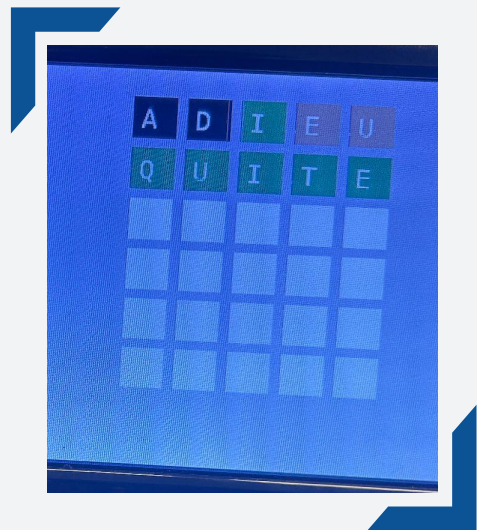
- Word list converted from .txt → .mem, loaded into RAM
- LFSR used to generate random index

2. I/O:

- Keyboard input via MMIO
- VGA reads from memory to display letters + colors

3. Features

- Capture keyboard input
- Supports backspace
- Compare input to answer (exact match → green, partial → yellow)
- Displays correct answer at the end



Word Hunt Implementation

1. Data Layout:

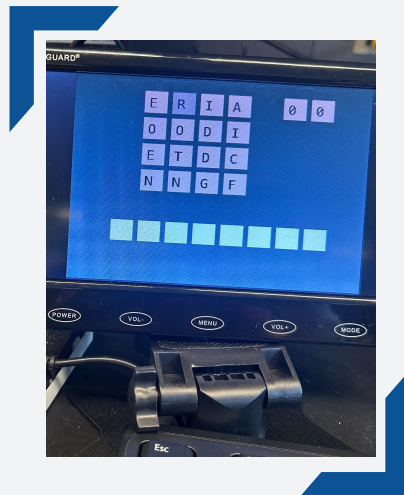
- Boards + Answer Keys stored in RAM, loaded to MMIO location
- User input MMIO

2. I/O:

- WASD: move cursor
- Spacebar: select letters
- Enter: submit word

3. Features:

- Only allow adjacent letters (check row/col difference)
- Prevent repeating cells
- Search answer list



Additional Features

1. **Menu**
2. **Keyboard navigation between 2 games**
3. **Intuitive UI**
4. **Longer words (up to 8 letters) spanning multiple 32-bit words in RAM**



Challenges

1. Word Encoding in Memory:

- Packing/unpacking English words into 32-bit words

2. Word Comparison Logic:

- Wordle: Exact matches (green) vs. partial (yellow)
- Word Hunt: Adjacency checks and answer lookup

3. I/O:

- Keyboard: Detecting valid keypress without glitches
- VGA: Mapping many RAM addresses to screen

Thanks!