Tile Matching Memory Game

Ashlee Stone and Ketevan Gallagher

Project Description

Our project is a tile matching game!

To play the game:

1. Select a tile, and flip it!
2. Select another tile and flip!
3. The two tiles get compared
    a. They stay flipped for only half a second if they don't match!
    b. If they do match they stay on
4. Repeat until all are matched!

The lower your score, the better!

Project Description

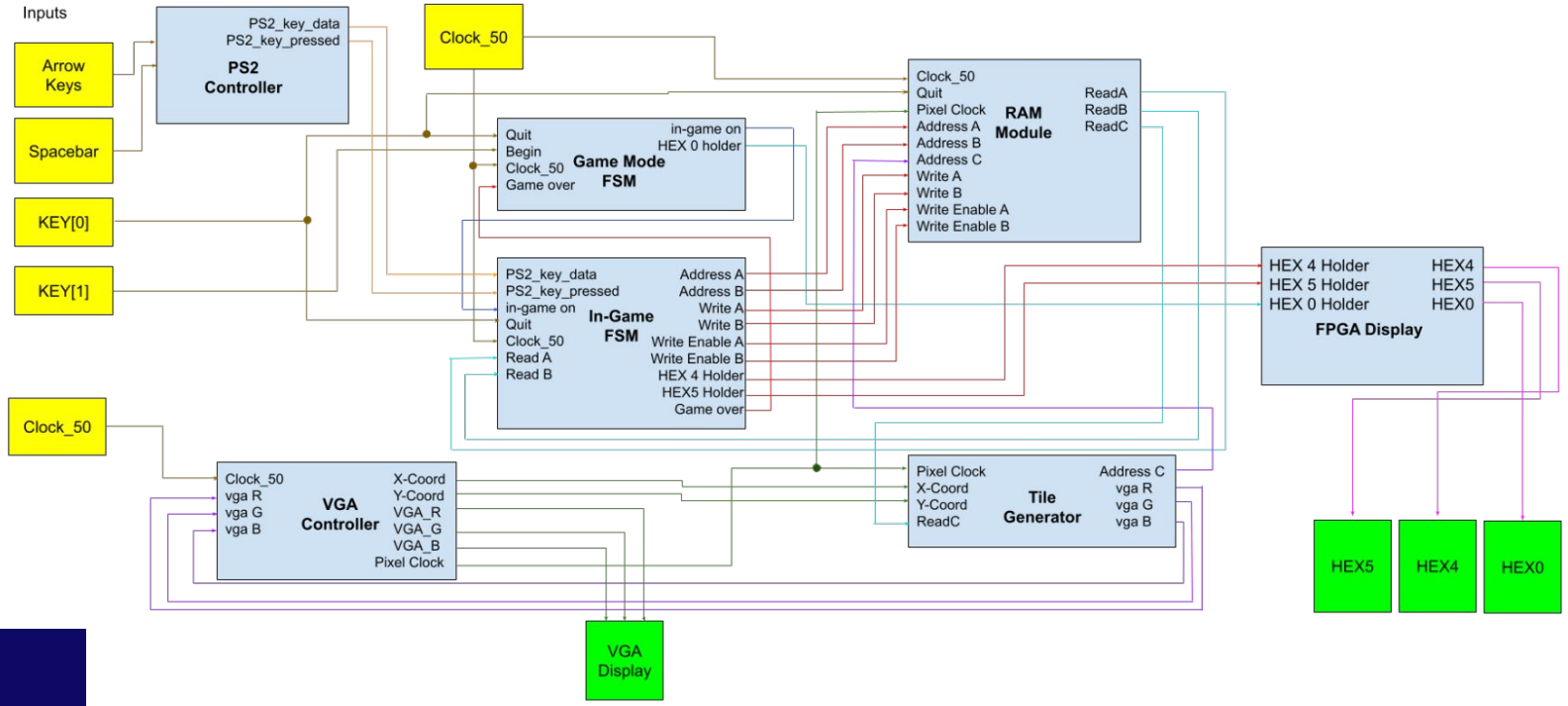
Game controls:

- KEY[0] to reset the game and VGA
- KEY[1] to start the game from the menu state
- Arrow keys to move between tiles
  - They wrap around the 4×4 grid
- Spacebar to flip over a tile

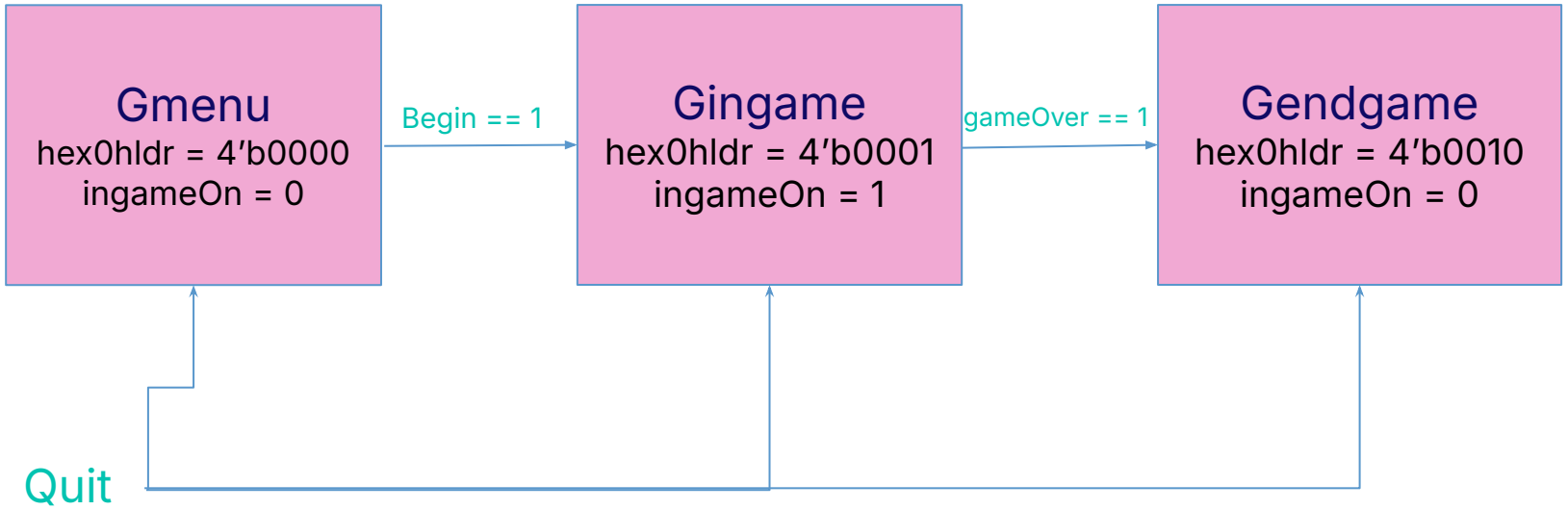
Display:

- Tiles shown on the VGA in a 4×4 grid
- HEX0 shows the game mode
- HEX4 and HEX5 show the player’s score in decimal

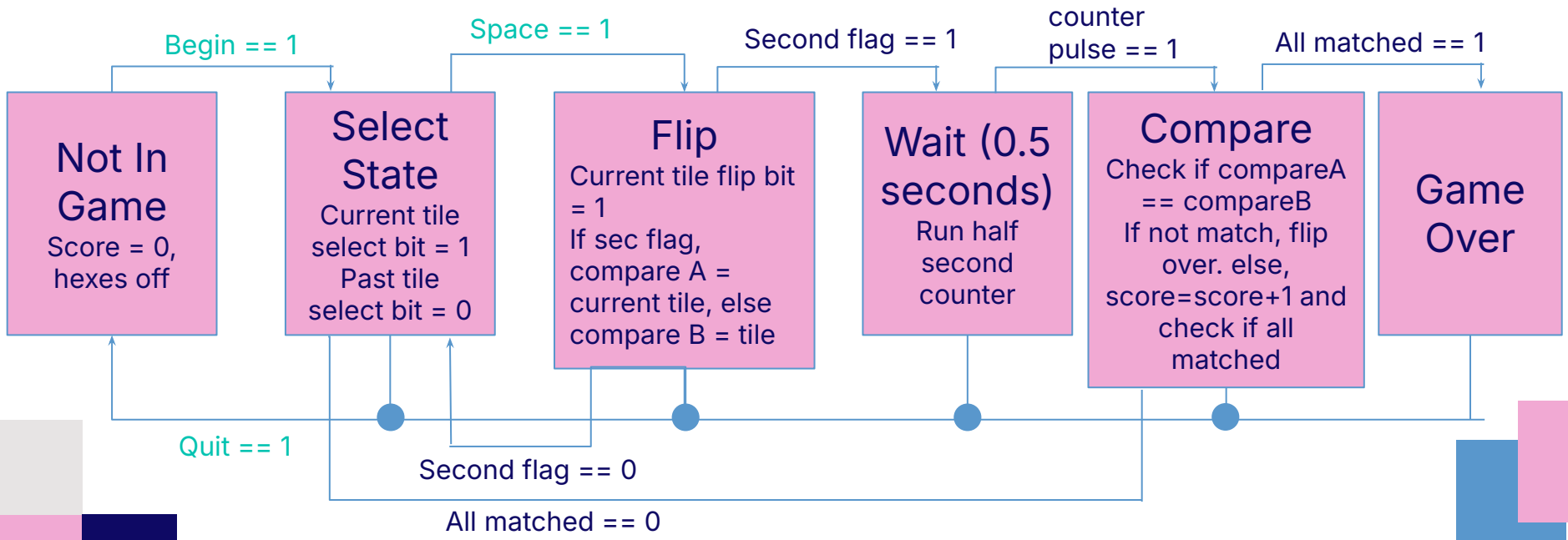
High Level Block Diagram



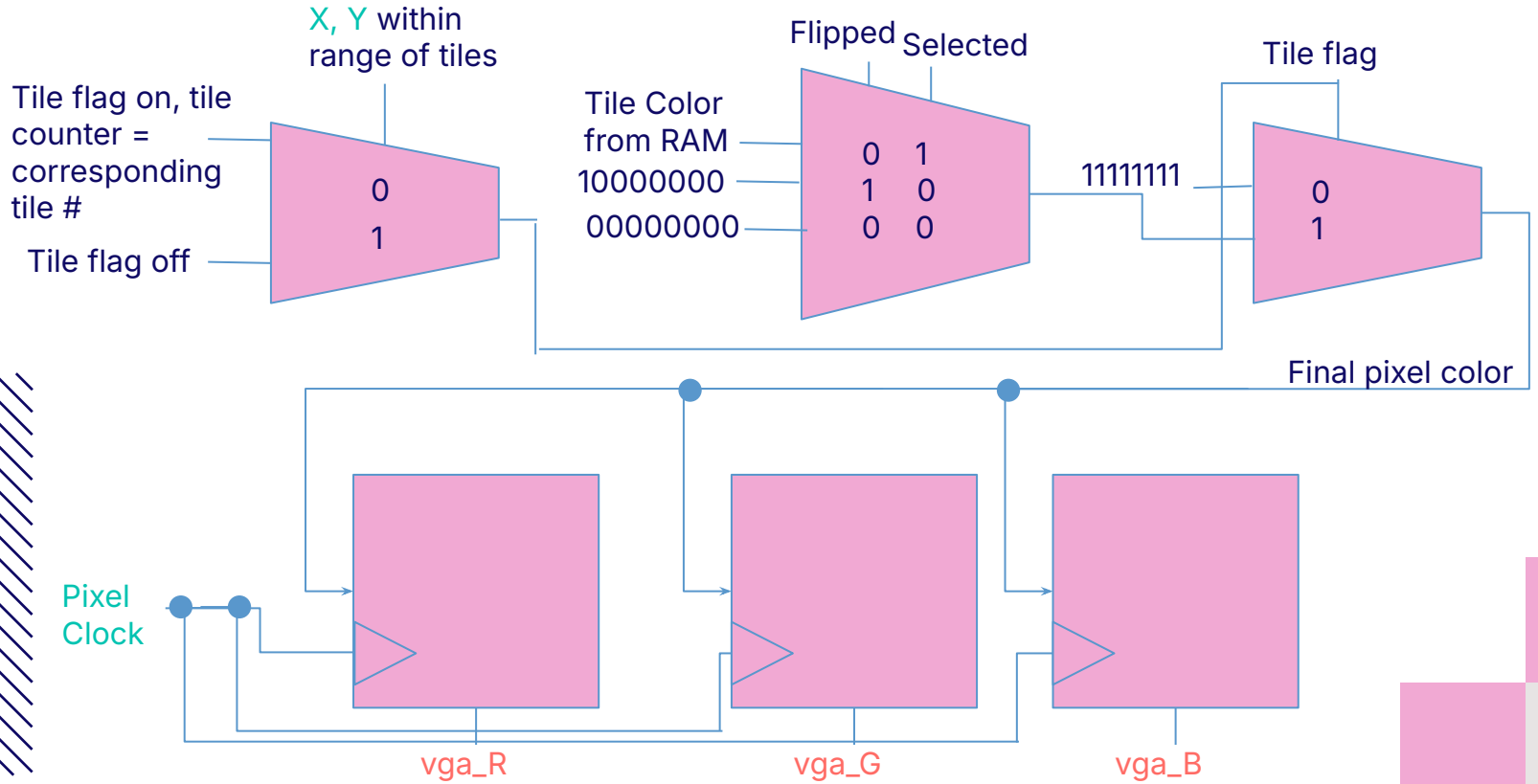
Game Mode FSM



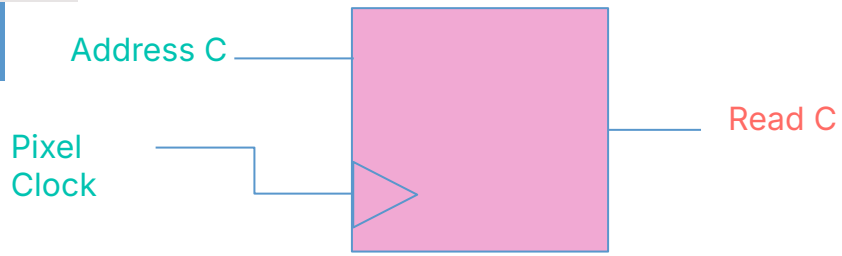
In Game Mode FSM

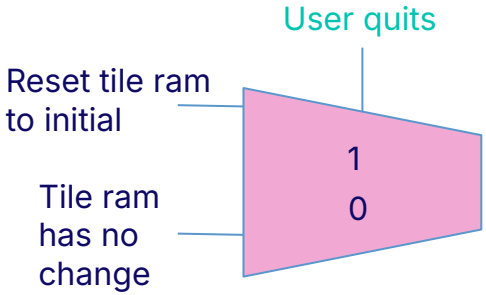


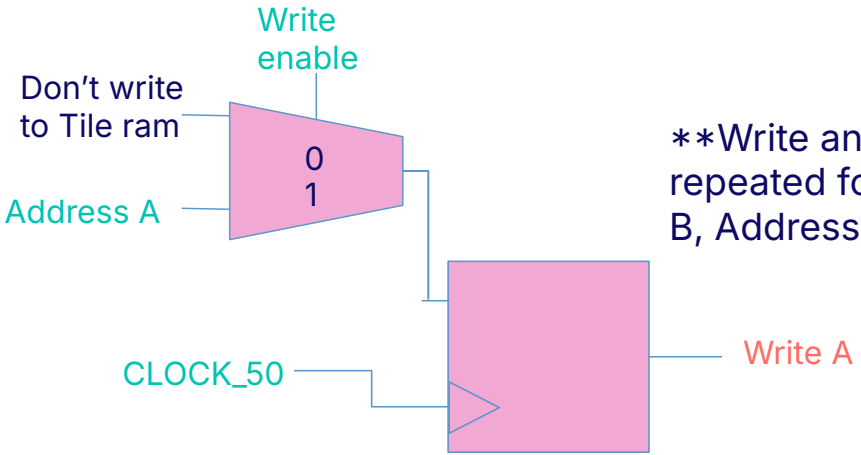
Tile Generator



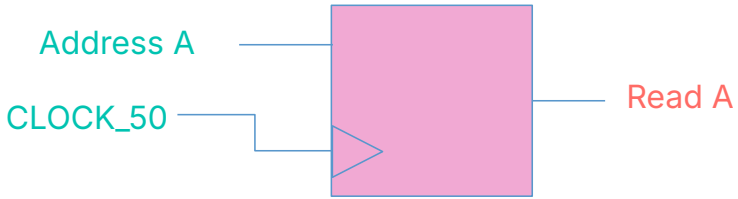
RAM Module



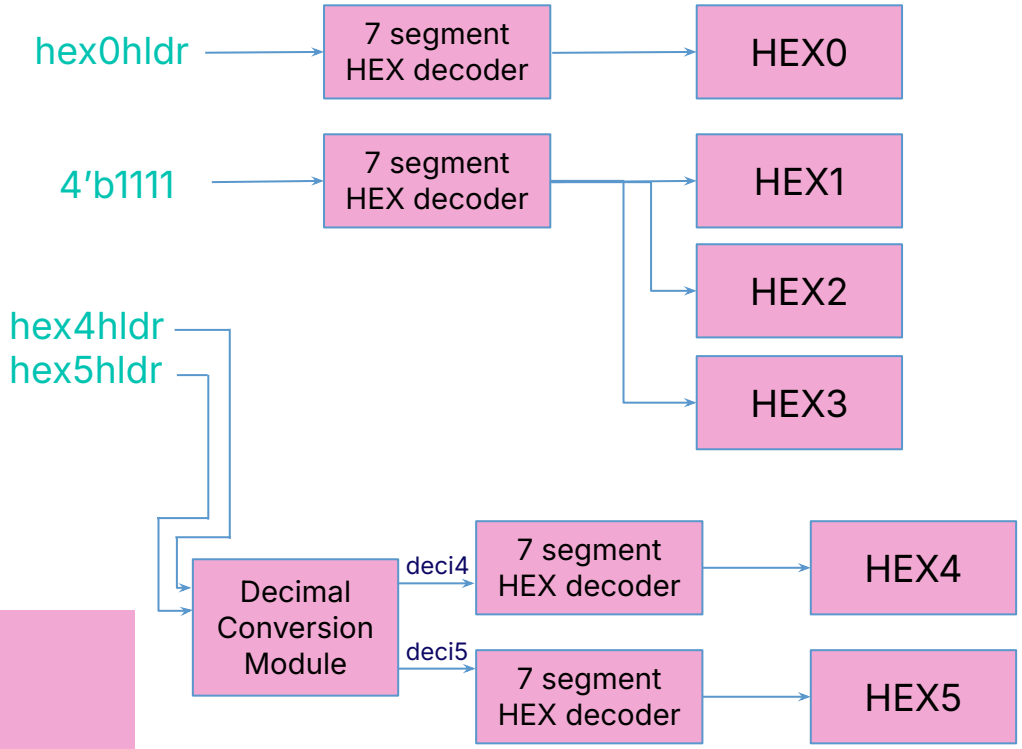




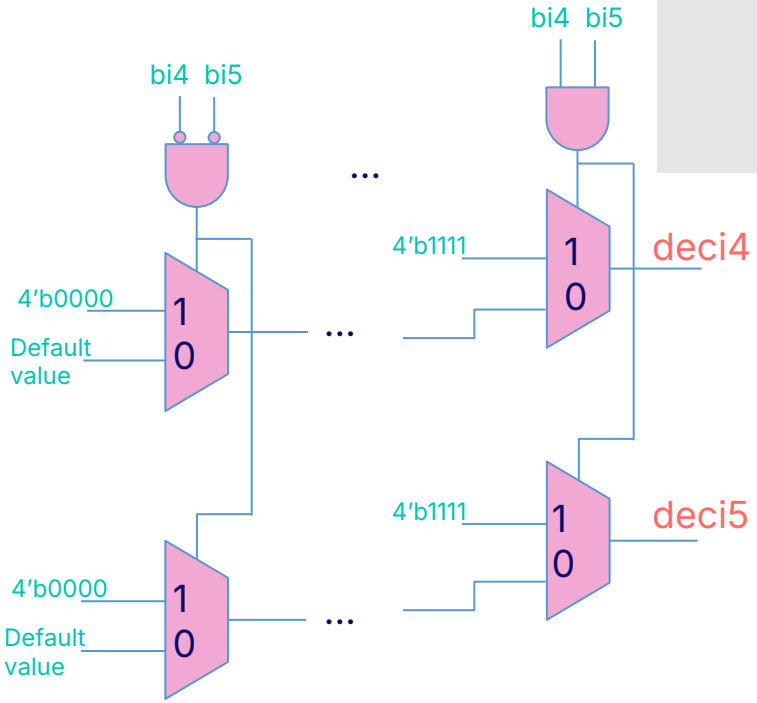
Write and Read blocks repeated for Read B, Write B, Address B



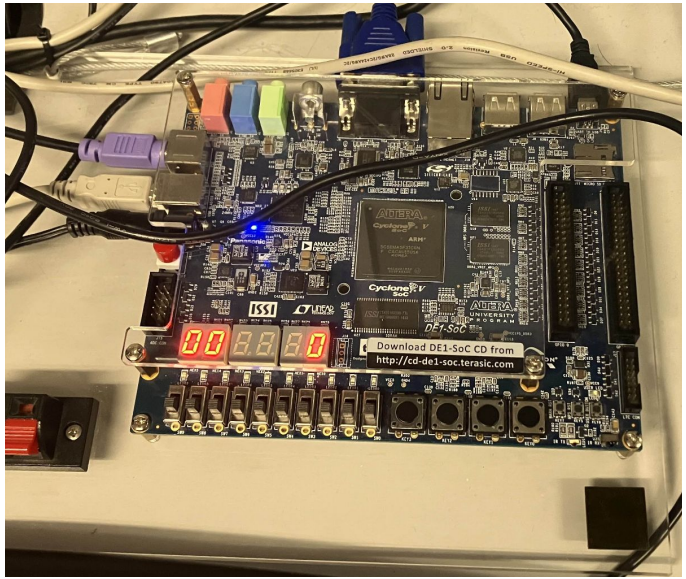
FPGA Display

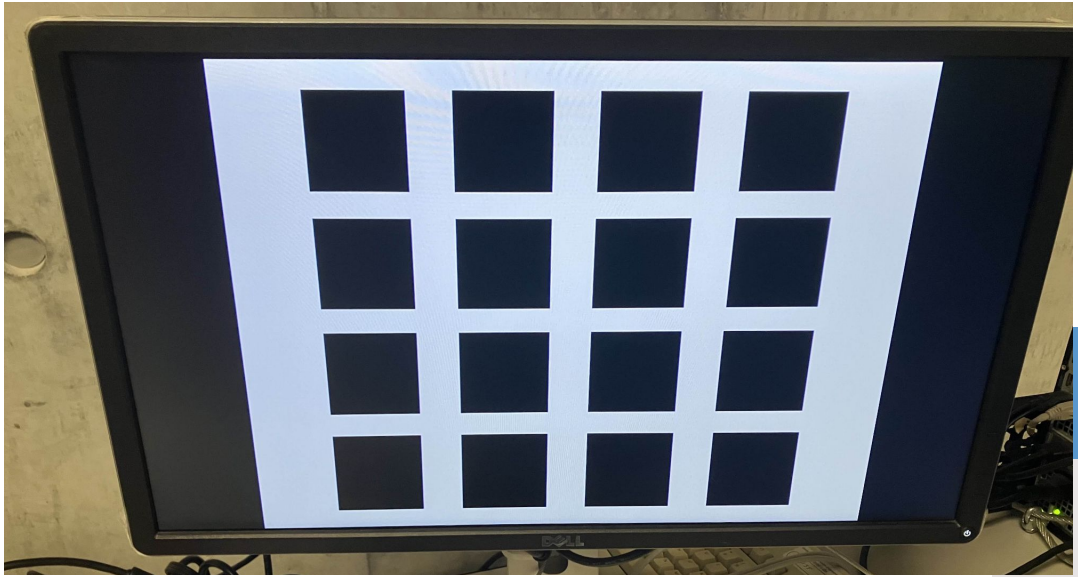


Decimal Conversion Module:

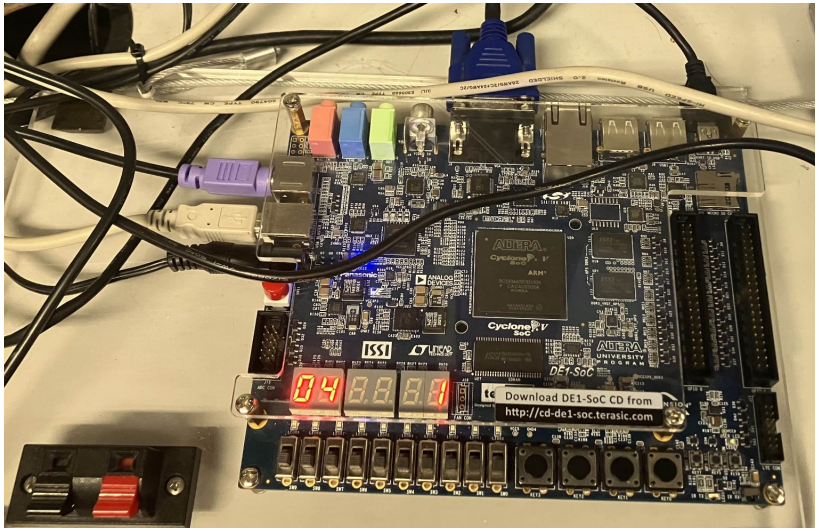


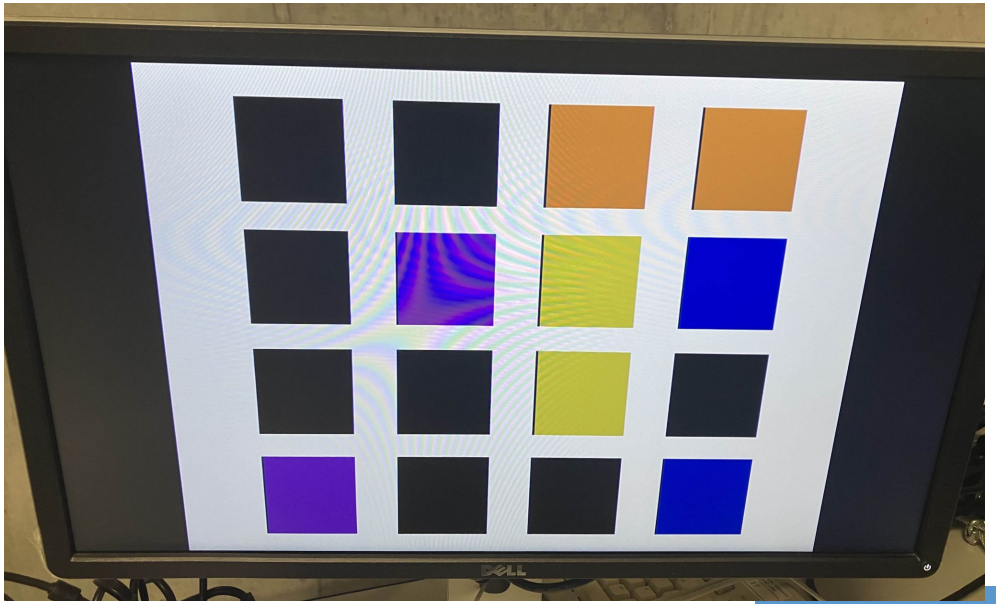
Project Pictures



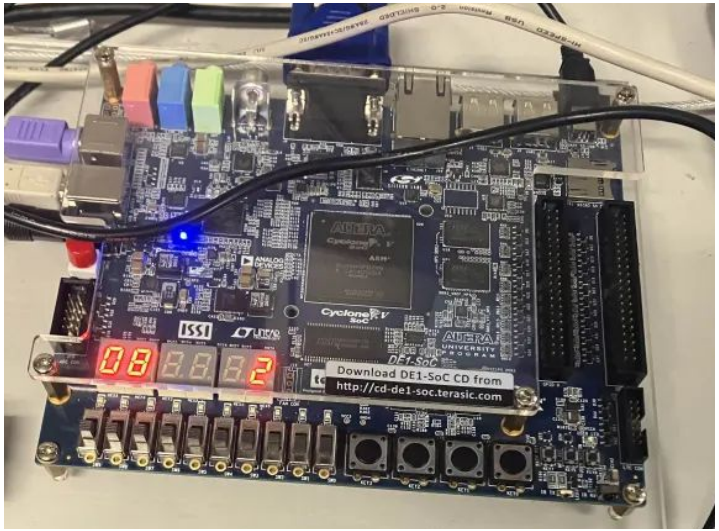


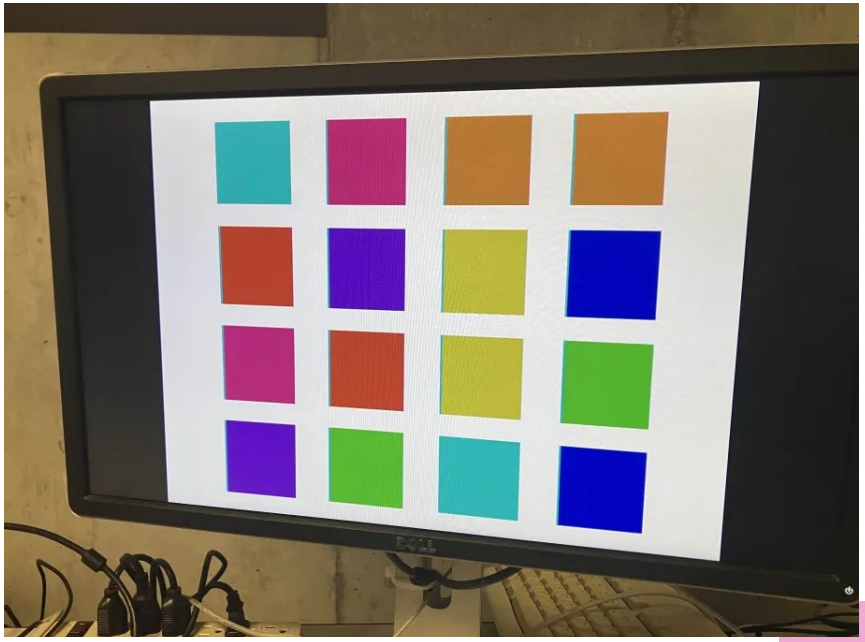
Project Pictures





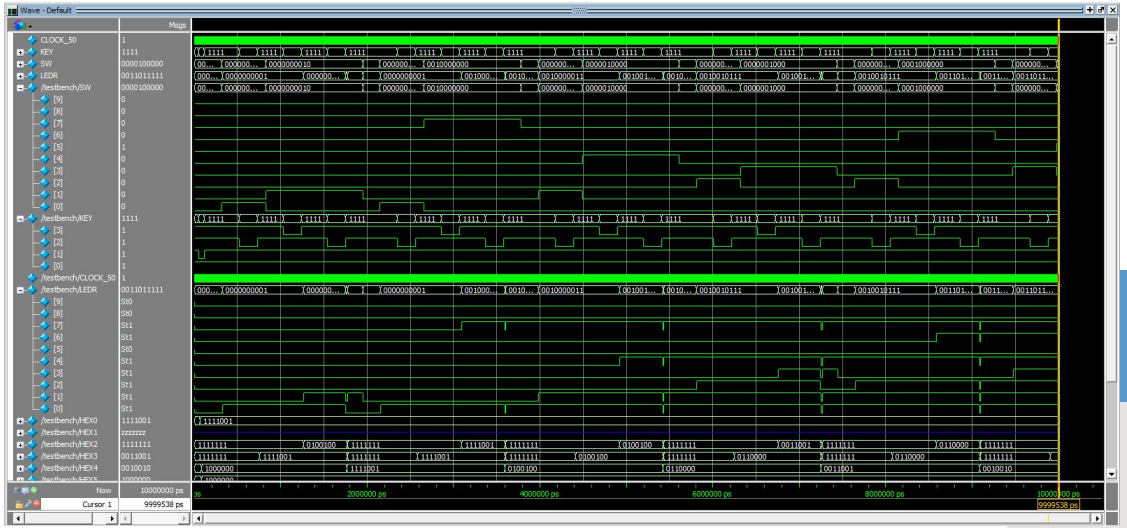
Project Pictures





Bugs and Fixes

- Game would not correctly change states when using inputs directly from the FPGA board
  - → The switches and keys could not be pressed fast enough
  - → Inputs would trigger twice
  - → Worked in ModelSim due to non-manual toggling
  - → Switched to PS2 input where the signal only triggers once then goes off

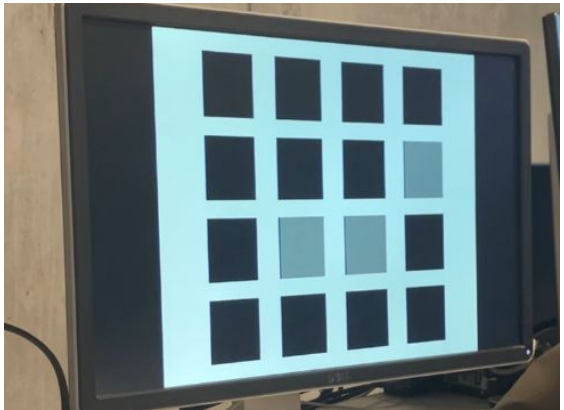


Bugs and Fixes

- Grey highlight stayed when tiles unflipped
  → Change select bit to zero when flip bit changes to zero

```
else begin
    addrA <= compareLocA;
    writeA <= compareA & 8'b11111110;
    addrB <= compareLocB;
    writeB <= compareB & 8'b11111110;
    weA <= 1'b1;
    weB <= 1'b1;
end
```

```
else begin
    addrA <= compareLocA;
    writeA <= compareA & 8'b11111100;
    addrB <= compareLocB;
    writeB <= compareB & 8'b11111100;
    weA <= 1'b1;
    weB <= 1'b1;
end
```



Bugs and Fixes

- Score wasn't showing up in Game Over state
  - → Jumped to the NotInGame state from SelectState after end of game
  - → Tested for error by adding values on the HEX displays in all in-game states
  - → Added an extra check in SelectState

```
SelectState:
    begin
      if (userquit == 1 || !inGameOn)
          nextInGame = NotInGame;
            else if (selectWait || selectWait2 == 2'b01 || selectWait2 == 2'b10)
                  nextInGame = SelectState;
        else if (select_pulse)
            nextInGame = Flip;
        else
            nextInGame = SelectState;
    end
```

```
SelectState:
    begin
          if (gameOver == 1)
                  nextInGame = OffGameOver;
        else if (userquit == 1 || !inGameOn)
            nextInGame = NotInGame;
              else if (selectWait || selectWait2 == 2'b01 || selectWait2 == 2'b10)
                    nextInGame = SelectState;
        else if (select_pulse)
            nextInGame = Flip;
        else
            nextInGame = SelectState;
    end
```

Bugs and Fixes

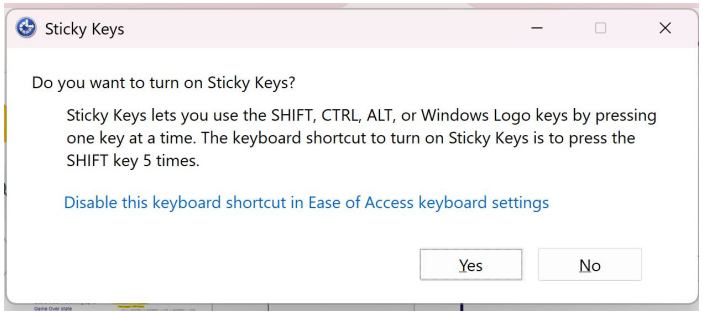
- First pair of tiles matched didn’t show color of second selected tile
  → Set counter value and pulse from the counter in initial block

```
initial
begin
   currentInGame <= NotInGame;
   nextInGame <= NotInGame;
   gameOver <= 1'b0;
   weA <= 1'b0;
   weB <= 1'b0;
   currentMatched <= 4'b0000;
   compareA <= 8'b00000000000;
   compareB <= 8'b00000000000;
   dementiaScore <= 8'b00000000;
   currentTile <= 4'b0000;
   waitCycle <= 1'b0;
   waitCycle2 <= 1'b0;
   firstInFlip <= 1'b0;
   selectWait <= 1'b0;
   selectWait2 <= 2'b0;
   compareWait <= 1'b0;


end
```

```
initial
begin
   currentInGame <= NotInGame;
   nextInGame <= NotInGame;
   gameOver <= 1'b0;
   weA <= 1'b0;
   weB <= 1'b0;
   currentMatched <= 4'b0000;
   compareA <= 8'b00000000000;
   compareB <= 8'b00000000000;
   dementiaScore <= 8'b00000000;
   currentTile <= 4'b0000;
   waitCycle <= 1'b0;
   waitCycle2 <= 1'b0;
   firstInFlip <= 1'b0;
   selectWait <= 1'b0;
   selectWait2 <= 2'b0;
   compareWait <= 1'b0;
   counter <= 27'd100000000;
   counterPulse <= 1'b0;

end
```

Future Work

- Fixing sticky keys
- Not allowing the user to select previously selected tiles
- Adding a randomizer to the tile positions



Demo Time!

Project Distribution

Ashlee

- Game Mode FSM
- PS2 Controls
- FPGA Display
- In-Game (FPGA version)
- In-Game debugging
- Top level module
- Presentation slides
- Final block diagrams (game mode FSM, FPGA display)

Ketevan

- Design for In-Game FSM
- Tile generation and VGA display
- In-Game debugging
- RAM module
- Top level module
- Presentation Slides
- Final block diagrams (tile generator, RAM, In-Game FSM)

Thank you!

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**