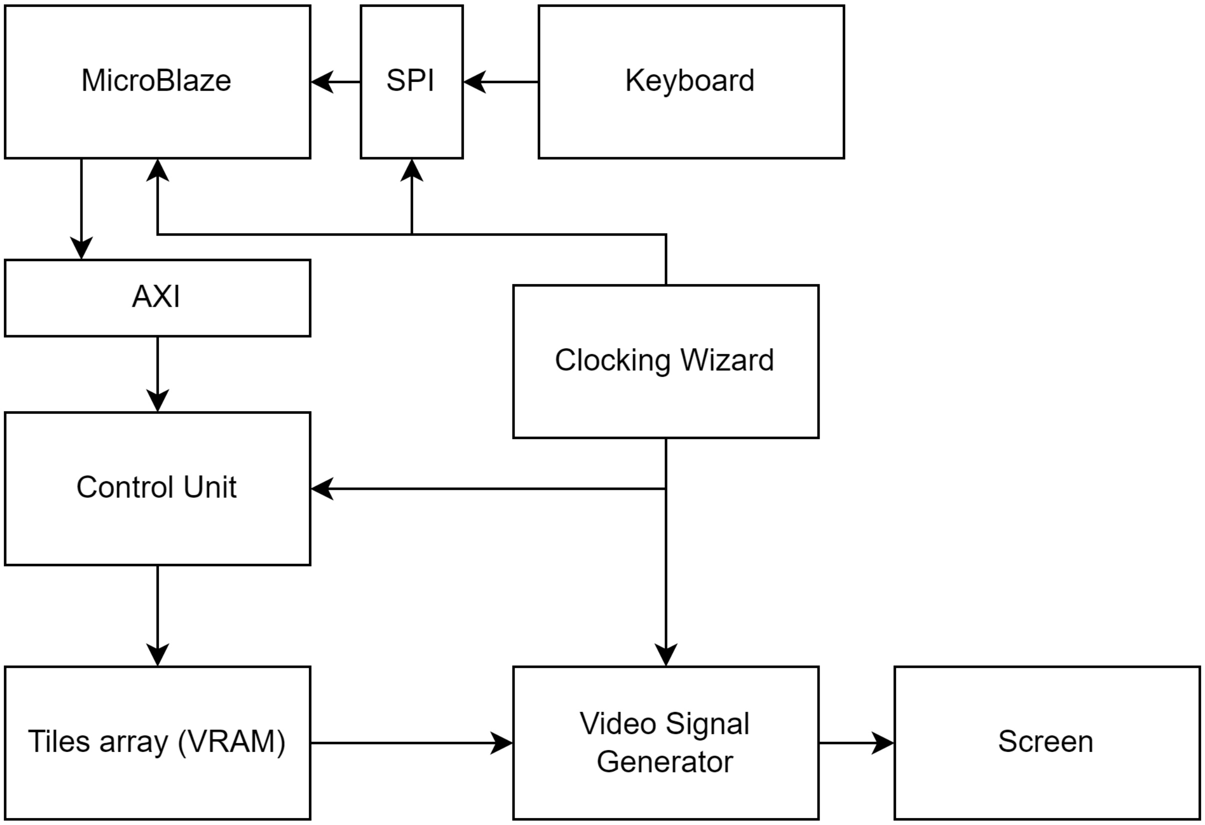
Final Project

Rachel Abraham (rachel20)
Albert Jiang (hongyi9)
TA: Henry Gillespie

Idea and Overview

We propose to design and implement a tile-matching game on the FPGA. The basic gameplay is to switch two adjacent tiles. Tiles will disappear if more than two of the same color are in the line, and tiles will fall down with new tiles refreshing at the top. Additional features include special items, score displays, animation, and sound effects. The game will be handling USB keyboard input using MicroBlaze as in lab 6. MicroBlaze will also check for valid moves and report that to the Control Unit. Then the tiles' positions, which are stored in RAM, will change based on the input. The color video display will be first generated as VGA signal and then converted to HDMI signal as in lab 7. The final goal is to play this game displayed on a monitor using a keyboard.

Block Diagram



List of Features

Baseline feature:

1. Tiles with different colors have been displayed.
2. The players can select a tile using a USB keyboard.
3. The player can switch the selected tiles with one of its adjacent tiles by pressing the arrow keys. The switch is only legal when it creates a new match of at least three tiles of the same color in a line.

4. Tiles disappear when there is a match of at least three tiles of the same color in a line.
5. The tile above the disappeared tiles falls down to fill up the space, and new tiles refresh from the top. If more matches appear after refreshing, those tiles will disappear too. Refresh continues until no match appears right after refresh.
6. The player can continue selecting tiles until there is no legal move possible.

Additional features:

1. Tiles have not only color but also drawings.
2. Tile animation when switching.
3. Special tile refresh when there is a match with more than three tiles of the same color. This tile will have the same color as the match, but it can eliminate a whole column and row when used for a match.
4. The score displayed at the top left corner shows the number of tiles that have been eliminated so far.
5. Special items are available to purchase using scores. Items include: bomb that eliminates all tiles with the chosen color, refresh that refreshes all the tiles, wild card that randomly changes some tiles to special tiles.
6. Various play fields with different sizes and shapes. Special block tiles that can't match any colors.
7. Falling animation when tiles disappear and fall from the top.
8. Background music and sound effects with each action.
9. Starting screen with tutorial

Expected Difficulty

We expect the baseline game doesn't include any animation or additional features would be a difficulty of 4. The challenge of the baseline version would be the logic of finding legal moves, eliminating tiles, and refreshing new tiles. Small additional features such as drawings for each tiles, special tile, items, scores, and starting screen should each raise difficulty by 1. Adding animation requires us to either use a frame buffer or object attribute memory to draw smooth falling on the screen. We would expect this to raise difficulty by at least 3. Having background music and sound effects should raise difficulty by 2.

Proposed Timeline

4/8 Waiting for proposal approval. This week we should try modifying lab 6.2 so that it uses frame buffer or object attribute memory. This should be a good start for implementing animation.

4/18 This week we should start working on the basic gameplay. Starting with lab 6.2 and 7.1 code, we should be able to implement the baseline function such as displaying tiles with different colors, selecting tiles using the keyboard, detecting legal moves, and game state machine. We might not have full gameplay, but we should be able to show some baseline gameplay for the mid-point check.

4/22 This week we should finish all baseline features and start adding additional features. We may start working on the falling animations and score display.

4/29 Final week, we should start wrapping up with every feature we can finish. Prepare the final version with all functional features.