Project Checkpoint 6 (+5 extra credit)

Interactive Game

Logistics

This is the last project checkpoint.

- Due: **Friday, December 13, 2024, at 6:00 PM** (Duke Time).
- Collaboration: you should form a team of two or three students. Feel free to form new teams for this PC.
- If you intend to work on this project, please complete this form by **December 8**. Afterward, we will release a time slot reservation form for you to book your project demo time.
- Project demos will take place in Hudson 214. **Students should be ready to run their project before their scheduled time.** The demo should last no longer than 5 minutes (showing all functionalities) while having up to 10 additional minutes for questions, for a total of 15 minutes per project. At least 2 evaluators will be present and ask questions, including TAs and the professor.

Requirements

You are to implement one of the below interactive games using your processor and IO components (a VGA screen and a PS2 keyboard). One major component of your design has to run in software on the processor (feel free to create additional custom ISA instructions). You are only expected to implement the most basic level of the game, but do not let this limit your creativity and innovation. Note that using behavioral Verilog is allowed for this PC.

Game options and requirements:

- Snake (https://en.wikipedia.org/wiki/Snake_(video_game_genre))
  - Snake's head continually moves forward without any buttons pressed
  - Keyboard buttons can steer the head left and right when they're pressed once (the rest of the snake's body should follow)
  - Snake's tail grows by 1 and the score increments with every item consumed by the snake's head
  - Game over when the snake's head hits a wall or the snake's body
- Tetris (https://en.wikipedia.org/wiki/Tetris)
  - Random blocks (among all 7 blocks) keep falling from top to bottom, one at a time
  - Blocks can be rotated and moved left and right using keyboard buttons
  - Score increases with every full line made (that line disappears and brings everything above it downward)
  - Game over when a block reached the top
- Pong (https://en.wikipedia.org/wiki/Pong)
  - After a player throws the ball and until a player loses the ball, the ball should keep moving appropriately
  - Each player can move their paddle vertically across the left or right side of the screen using keyboard buttons
  - The ball reflects off of the paddles and the top and bottom walls
  - Score of player increments whenever the ball goes past their opponent's paddle
- Flappy Bird (https://en.wikipedia.org/wiki/Flappy_Bird)

  - Bird should jump by a certain amount every time a keyboard button is pressed
  - Pipes/shapes with random vertical openings’ positions should keep moving left
  - Score is incremented every time the bird makes it in an opening
  - Game over when bird doesn’t make it
- Space Invaders (https://en.wikipedia.org/wiki/Space_Invaders)
  - Ship moved left/right at the bottom of the screen controlled by keyboard buttons
  - Ship can shoot at attackers falling from the sky (attacker disappears when hit and score increments)
  - Game over when attackers touch the ship or hit it
- Sokoban (https://en.wikipedia.org/wiki/Sokoban)
  - Person can move in all 4 directions using keyboard buttons
  - Boxes move appropriately (without going through walls)
  - Game over when all boxes are where they should be
- 1024 or 2048 (https://en.wikipedia.org/wiki/2048_(video_game))
  - Numbered tiles move appropriately (in 4 directions) with appropriate keyboard button press
  - New numbers are appropriately computed with every move
  - New random tiles appear appropriately with every move
  - Score increases with every move
  - Game over when out of legal moves

Deliverables and Grading

Teams who decide to go for the extra credit should demonstrate the functionality of their project during their scheduled time slot. Extra credit will be awarded after assessing the functionality and design of the project based on the following criteria:

● Ability to move components on the screen using a PS2 keyboard:	+1
● Ability to show a score on the screen that updates according to the game rules:	+1
● Running a significant part in software on the CPU:	+1
● Basic game rules are applied and functional:	+2