Michael Rodriguez
Ahmet Musabeyoglu

Interactive Checkers

Overview-

This project seeks to create a new interactive way of playing checkers by combining FPGA driven image processing and an Artificially Intelligent checkers player. This hybrid game will be a combination of a board game and a computer game. The game is set up by placing a flat computer screen horizontally on the table. An 8x8 checkers board will be displayed on this screen. The human player will place their 16 real red pieces on the checkers board and the computer's 16 black pieces will be displayed on the screen. When it is human player's turn to play, the player will move one of their own real pieces to the desired location and will press a "ready" button to inform the AI player that human's move is completed.

After human's move is completed, a webcam located above the screen will get an image of the screen and analyze what move has been made. When one of computer's pieces is removed from the game, it will vanish from the screen with some visual effects (for example breaking up and disappearing of that piece). When it is computer's turn to play, our computer AI player will calculate an optimal movement and will execute it on the screen. When one of human's pieces is removed from the game, the square that piece is located in will start blinking colorfully signaling the human player to remove their own piece from the game board. Combining all this functionality will create a completely new interactive way to play Checkers.

Design Description-

This checkers game is split into four main parts: the Image Capture Module with a camera that captures the current state of the board and sends this information to the FPGA, the Image Processing Module which takes in the captured images of the game board and determines which move has been made by the human player, the Computer AI Module that takes in the human player's move and outputs the new state of the board, and finally the Display Module that draws all these changes to the computer screen. These four parts, as seen in the Figure 1 below, represent the flow of information that occurs in every round of game play.

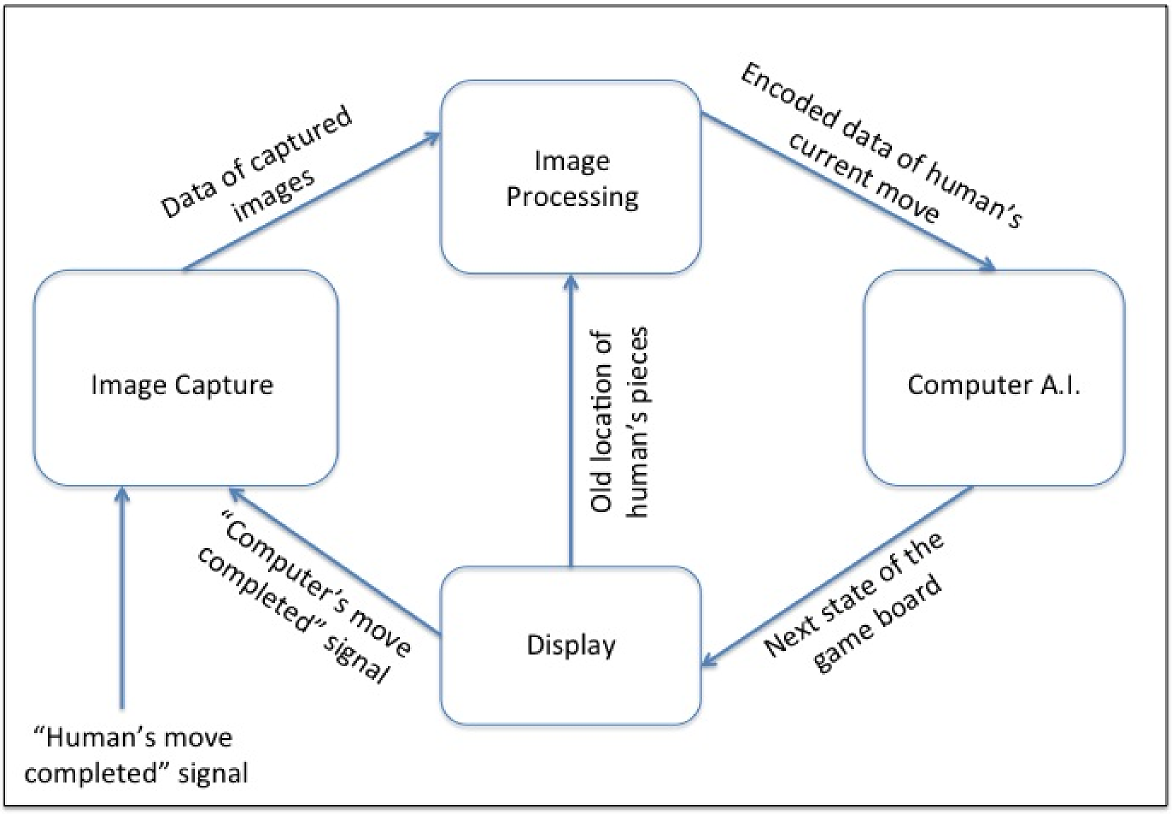
Image Processing
Data of captured images
Encoded data of human's current move
Image Capture
Computer A.I.
Old location of human's pieces
"Computer's move completed" signal
Next state of the game board
Display
"Human's move completed" signal

Figure 1: Block Diagram of Interactive Checkers Game

Figure 1 illustrates the transfer of data amongst the four modules. Each cycle represents a turn of game play.

Image Capture Module: The Image Capture Module uses an NTSC camera interface to transfer live video of the game board to ZBT memory. This interface will be waiting on two signals before it makes the data transfer. The first is a debounced button that is pressed by the human player after their move is completed. The second is a signal from the Display Module which signals that the computer's previous counter move has already been drawn to the screen.

Image Processing Module: The Image Processing Module pull video data from ZBT memory and determines which move was made by the human player. This is done by comparing the two sets of data. The first piece of data (from the Display Module) is a list of the location of every human piece at the end of the previous round. The second set of data comes from the Camera which is analyzed to determine the current location of all the human's pieces. The different between these two data sets is the human player's current move. This move is then encoded to be sent off to the Computer AI through a serial port.

Computer AI Module: The Computer AI Module takes in the Human's current move and calculates a counter move. Once this module receives the encoded data of human's current move from Image Processing Module, it decodes this data to determine which move has been made. Then, AI code running in the computer executes this move and determines the optimal counter move that should be made by the computer. After analyzing the counter move's implications on all other pieces, the Computer AI Module encodes the state of each of the 32 squares (ie. Square 4 has a red piece) and sends that data to the Display Module.

Display Module: The Display Module is responsible for drawing the state of the game board onto the screen. Once this module receives the encoded data of the next state from Computer AI Module, first it decodes this data, then reflects the new state of each block on the checker's board. Once the new state of the game board has been drawn, the Display Module sends a "Computer's move completed" signal to the Image Capture Module. The Display Module also sends the information about previous locations of human's pieces to Image Processing Module.

Conclusion-

Our "Interactive Checkers" game will create the feeling of playing checkers on an actual game board, which differs from the regular checkers games played through mouse inputs on a computer. The unique user interface and playing style of our game will make it more enjoyable and easier to play. Thus, our game will be a good alternative to other computer checkers games on the market.

Goals-

- Set up an interactive checkers environment by placing a flat computer screen horizontally, displaying the checkers board on the screen and placing a camera above the screen.
- Determine the human player's moves efficiently, through advanced image processing techniques of the camera feed.
- Create the basic playing functionalities of a checkers game.

Stretch Goals-

- Implement logic that determines if an illegal move was made by the human player. This logic will request a new move from the human player.
- Currently a button pressed by the Player tells our system if their move has been completed. We hope to be able to use Image Processing Techniques to determine, from the camera feed, when the move has been made.
- Convert the checkers AI algorithm into a Verilog code, in order to run the artificial intelligence directly on the FPGA. This will eliminate the need for the extra computer.

Potential Problems-

- The complexity in the image processing will be the biggest issue. We need to keep track of up to 16 small objects at the same time. Also our image processor needs to be able to correctly determine where each square is on the board in order to analyze the state of each square.
- Problems may arise when interfacing between our FPGA and the computer that runs the AI algorithm.

Timeline-

November 11, 2013: Image Capture block will constructed and we will start working on the Image Processing block.

November 16, 2013: Interface the FPGA with the Computer AI block.

November 22, 2013: Complete the interface between Computer AI and Display blocks.

November 30, 2013: Put different parts together and start debugging the whole system.