i. Inputs

1. SW[6:0]: Indicated with column to make a move. If SW[0] is 1 when KEY[3] is pressed, this will count as the player making a move in column 0. Similarly, SW[1] indicates column 1, SW[2] is column 2 and so on. If multiple switches in SW[6:0] are set when KEY[3] is pushed, this will count as an invalid move.
2. SW[9]: This must be set on ONLY one board. If this switch is set to 1 after reset, then we are player 1 and will move first. Otherwise, we are player 2 and we will move second. Both board are using the clock from the board whichever has SW[9] on.
3. KEY[0]: Reset the board
4. KEY[3]: Submit a move. When it is our board's turn to make a move, this will indicate that we have decided what move to make. To choose a column to make a move, use SW[6:0], as described above.
5. GPIO[35:31]: used to send move data between our board and the other board. When it is the other board's turn to make a move, we will wait until they send a move over the GPIO wires. We send data and ready signal using GPIO[34] and GPIO[33]. We receive data and ready signal using GPIO[32] and GPIO[31].
6. GPIO[23:0]: used to drive the LED matrix.

ii. Outputs

1. LED matrix: A 7x7 matrix is displayed on the lower right of the LED matrix, representing the board. Red indicates player 1, Green indicates player 2
2. Console of Eclipse:
   a) Tell users whose turn to make the move.
   b) Print out grind each time users make a move.
   c) Tell users when someone makes false move
   d) Tell users when one player wins and also display the times of wins, loses and ties for the user.
3. FPGA board:
   a) LEDR[8:6]: indicate data transferred from the other player to our board. LEDR[8] is for the transferred clock and will light up when we are using the other player's board. LEDR[7] and LEDR[6] are for ready signal. LEDR[7] will light up when ready signal is sent and LEDR[6] will light up when ready signal is received.
   b) LEDR[5]: indicate if it's our turn to move. It will light up when it is our turn

c) HEX[5:0]: HEX[5:0] are used for displaying times of wins, losses and ties. HEX0 and HEX 1 are for wins, HEX2, HEX3 are for losses and HEX4 , HEX5 are for ties.

iii. Major function

1. For hardware/verilog
   a) Program NIOS II microprocessor and use it to connect to software system.
   b) Communicate with the other board by inputting their clock, sending signal and data an also outputting our clock, sending signal and data.
   c) Convert the signal received from the other board bit by bit to a four-bit signal showing 0-6 representing all columns.
2. For software/c code
   a) Ask the user to input when it's their turn
   b) Print out error message when a false move is made
   c) Print out the current grid each time a user made a move
   d) Print out the winner of the game

iv. Function decomposition

1. For hardware/verilog
   a) first_nios2_system: used to program microprocessor NIOS II
   b) io_handler: used to handle communication with the other board
   c) led_matrix_driver: used to drive matrix
   d) Long_binary_to_short: convert inputs from switches to a 4-bit value and represent our next move
   e) HEXdisplay: drive the seven-segment HEX to display the times of wins, losses and ties
2. For software/c code
   a) get_input_from_other_board: get input from other board, pass it into the grid and also print to the console
   b) get_input_from_board: get input from our board, pass it into the grid and also print to the console
   c) print_val: convert the value that will be printed to the console into decimal
   d) main: handle most functions for the game. It inputs each move from both players and print out updates grid each time. It detects false move from the players and give warnings. It also detects wins, loses or ties and print them out to the console and also outputs to the microprocessor

v. Processing procedure

1. When initially starting, the console will print out “Hello” to indicate that it’s working.
2. Use SW[9] to decide who will play first. if our switch is true, then we are player 1 and will play first, LEDR[5] and LEDR[8] will light up since it’s our turn and we offer clock. On the console, our move will also be printed out as number “1”. Otherwise we are player 2 and will play second. Press KEY[0] to reset entire board. The HEX[5:0] will display “0” for the times of wins, losses and ties.
3. When we begin first, if it’s our move, the console will print out “Your Move. Enter a column (on the board): “ to ask for column. We input the column we want to drop the disc using SW[6:0], and then we press KEY[3] to indicate that we’ve made a move. Then the console will print out “Got input from board: “ and also print the input got from console in two decimal digits. Then the console will print out the current grid after the move, our move will be shown as number “1” on the console grid. The led matrix will light up the red light on the corresponding column and row.

    i. If it’s the other player’s move, the console will print out “Their Move. Enter a column (on other board): “ to wait for the input from the other player. It will also print out “Got Input from other board: “ and display the input from the other player. Then the console will print out the current grid after the move, the other player’s move will be shown as number “2” on the console grid. The led matrix will light up the green light on the corresponding column and row. Note that the grid printed on the console and the led matrix will both display the discs from bottom row to simulate the bottom of the disc.
4. If the other player begin first, the output is basically the same except that we will be player 2 and number “2” and the green light will represent us.
5. After several plays. If one player wins, the game will end. If we win, the console will print out “You Won” on the console. Otherwise, the console will print out “They Won”. Also times of wins, losses and ties from us will be printed to the console as two decimal digits for each and also displayed to the HEX of the board beginning from “00 00 00”.
6. If no one wins until the entire grid is filled, whoever’s turn to move shall use SW[7] to reset the board so that we can begin next game. And the times of ties will increase. If we reset the board, the console will print out “Game reset by us”. Else it will display “Game reset by other player”. Times of wins, losses and ties from us will be printed to the console as

two decimal digits for each and also displayed to the HEX of the board beginning from “00 00 00”. Note that SW[7] is not only used for resetting when the game ties. It can reset the board whenever needed.

7. When one player wins or the game ties. The result of the game will stay for about 1 second and then the board will restart automatically. A new game will start. The console will print out “Starting a new game”. Whoever plays second in the first round will play first but the player is still player 2.
8. The system can lock the move of a player. When it’s not our turn to play, the system will not receive data from our switches so that it won’t change even if we make a move. When it’s our turn, the system will not receive data from the other board
9. The system can detect false move. In the case that the input column is larger than the grid or the position the disc is dropped is already full, when it’s our move, the console will print out “You made an invalid move: “ and display the column we input. When it’s the other player’s input, we will send back to the other player value 15 to tell them that they’ve made a false input. When we try to input two columns at the same time, the console will also print out “You made an invalid move: 15” to indicate that our move is invalid