ECSE 318 – Lab 2 – Free Cell

To create the tableau in free cell we chose to represent it as a two-dimensional array that held 6 bit entries. This meant that the entire tableau could be referenced with one variable as shown here: reg [5:0] col [7:0][15:0];. We chose to represent home and free similarly as one-dimensional arrays, and we also used an integer array called largest to store the location of the lowest (therefore playable)) card in each of the 8 columns. We also used 2 temporary variables one called card to remember the card that we're picking up and one called homeSpot which remembered which home pile the card was eligible to go in because we did not assign a specific home for each suite, rather filled them as the game progressed. To improve readability, we also added all the cards as local parameters so that they could be easily referenced as opposed to only using their six-bit value. We also used a temporary variable to store the source and destination while debugging and decided to retain this feature as it protects the program from allowing an incorrect move to be made if the source and destination are changed before the negative edge of the clock.

After the board state is initialized, we begin on the positive edge of the clock resetting the source and destination validity to 0 and emptying the card that we were holding then we have two casex statements that check if the source we are pulling the card from is valid and if the destination, we are trying to place said card is valid. The efficiency of this function is greatly helped by the fact that we used a 2-dimensional tableau to build this because we can reference the any move for any card using just three case statements. After validity of the move is checked on the positive edge of the clock, we then complete the move on the negative edge of the clock to make sure that our data never desyncs when this happens if the source and destination are both valid then the card will be deleted from the source and appended to the destination. If the destination is a home spot it will simply rewrite the card that was previously written, so no card can ever be removed from the home spot.

There are several embedded display statements in the code but most of these have been commented out to clear the terminal. **For the win to be reported in the terminal there must be 10 units of delay after the final move** to allow the final move to process otherwise the simulation will end and the wind will not be declared, i have modified my test bench to show this and correctly declare the win. Additionally, **the TB must be compiled BEFORE the Problem2.v, otherwise it will overwrite because the freeCellPlayer modules have the same name and return gobbledygook.**