Guessing Game

Design Document

Name: Carmen Tam

CSE 490/590 Computer Architecture

Initial Description

1. Design a two player guessing game.
2. The game starts with the four digit seven segment display illuminated with “PL 1” indicating that player #1 should enter a number between 0x0000 and 0xFFFF.
3. After the number has been entered, player #1 should press the center momentary push button (C).
4. The seven segment displays are used to indicate this by now displaying “PL 2”, indicating it is player #2’s turn.
5. Player #2 repeatedly enters four digit numbers until they correctly guess the number entered by player #1.
6. The left push button (L) is used to latch D0 , the upper push button (U) is used to latch D1 . The lower push button (D) is used to latch D2 . The right push button (R) is used to latch D3 .
7. When the entire number has been entered, the center push button (C) should be pressed.
8. When an incorrect number is entered, the display should indicate if the guess was too high (“2 HI”) or too low (“2 LO”).
9. When the correct guess is entered the LEDs should blink in celebration, and the number of guesses required should be displayed on the seven segment display.
10. The number of guesses should be displayed as a decimal number (NOT hex).
11. Initially all of the DP are off, after each incorrect guess, one of the DP lights up and remains on until the end of the game.
12. If 4 incorrect guesses are made, the seven segments should show “LOSE” for 3 seconds, and the game should be reset.

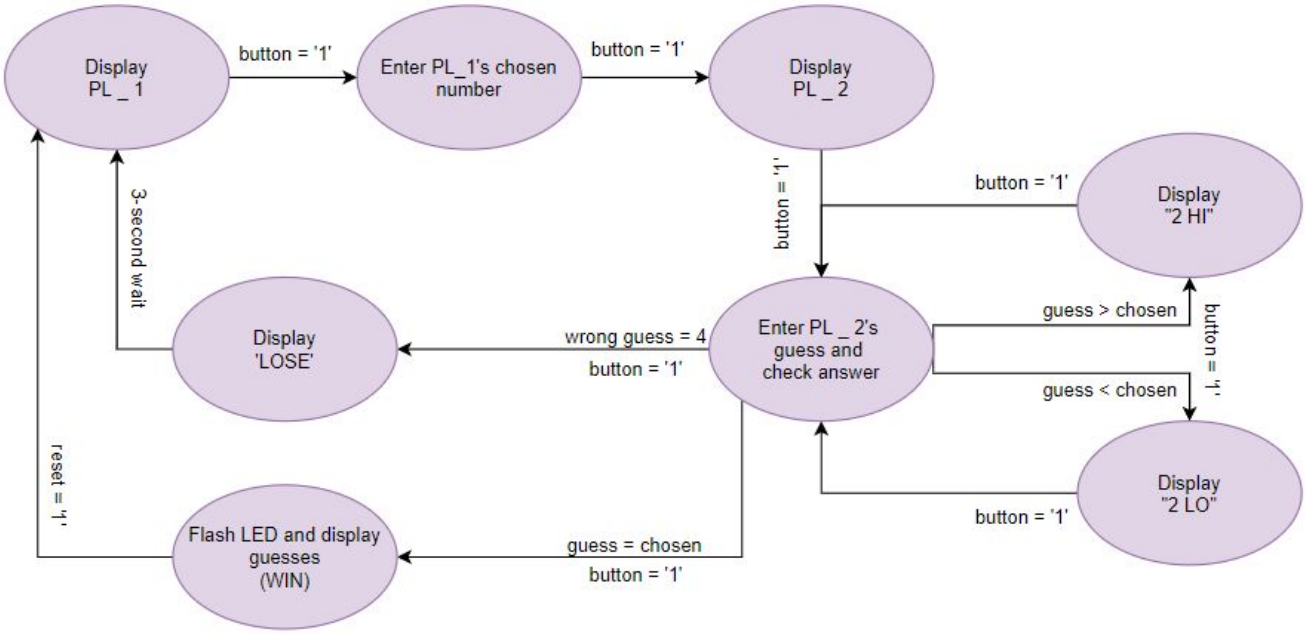
1. Modifications

Some of the modifications that had been made to the initial description of the guessing game were highlighted in the list above. Changes made included:

- At the beginning of the game, the seven segment display would illuminate with “PL 1.” However, the description had stated that this display would indicate the player to enter their number on the appeared screen. Instead, the program will illuminate “PL 1” and wait for the user to press the “center button” in order to ‘start’ their turn. When the user presses the center button on the “PL 1” screen, the display will appear with 4 zeros for the user to change their ‘chosen input.’
- When the correct guess is entered, only one LED, the most left one will flash on and off to indicate player 2’s victory.
- When the user wins the game, the number of guesses should be displayed on the board as a decimal number. However, the program will show the number of wrong guesses the user has entered.
  - **For example,** if the user had gotten the correct answer on their first try, the winning screen will display a zero as they had not had any incorrect attempts.
- Initially, the decimal points will be off unless the number of incorrect guesses are increasing. The same number of decimal points as incorrect guesses will appear on the display. Although the description explains that the decimal points remain on the screen until the end of the game, the program will show the number of decimal points on the input screen for player 2 instead.
  - **For example,** if the user has 2 incorrect guesses, the rightmost two decimal points will appear on the screen. However, when the “2 HI/2 LO” screen is displayed, the points will not appear until it is the user’s turn to input their next guess.

2. Implementation

- Using a clock cycle to implement a counter for the left-most two bits allowed traversing through the anodes smoothly in order to display a specific input or instruction.
- State Machine: Using a state machine, I divided each state in the following:
    A. Display PL_1
    B. Enter Player 1's chosen number.
    C. Display PL_2
    D. Enter Player 2's guess and calculate result : result will change the next state depending on the guess inputted by player 2.
    E. Display "2HI" : if guess was higher than chosen number.
    F. Display "2LO" : if guess was lower than chosen number.
    G. WIN : If guess was the same as the chosen number.
    H. LOSE : If the number of guesses reaches 4.



- Whenever a directional button is pressed, latches the values of the switches to a signal that converts the 4 bit input into an 8 bit signal for the seven segment display.
- When traversing through each anode, the specific seven segment is displayed and chosen based on which state the program is currently in. Using if statements for each state, the seven segment display will display accordingly to what is needed.
- Using another timer that is slower, traverses through its clock cycle allowing an LED to flicker every second when the player wins. Also counts for three seconds after the player loses in order to restart the game.

3. Mistakes

When it came to programming the guessing game, there were many new introduced approaches that I had wanted to implement, however, would impact my code negatively.

- During the beginning, when drafting my code, I had run into many problems caused by multi-driven nets which I couldn't seem to avoid. I had kept going around the problem trying different ways of overcomplicating it, creating new variable holders, and even changing where the signal would be changed. However, upon restarting my code again, I realised that I was unnecessarily relying on one process the whole time which caused the seven segment display to not show letters outside of the 0x000 - 0xFFFF range.
- Another problem I ran into was a debouncing issue. Since my program ran as a state machine, each state would be accessed through the center button press. However, whenever I had pressed the button, multiple states would be skipped since multiple signals from the button were registered. I had to find a way to debounce the signal and limit where the states change in order to limit the risk of skipping states.
- Decimal points and the wrong guess counter was always off. It had been an issue caused by the debouncing issue since it would trigger multiple times within a state, leading the counter to increment too fast. After changing where the counter would be triggered, the counting became more smooth and allowed the game to go on longer.
- When trying to implement decimal points for each digit on the display, I had created a whole new process for the switches to be passed through to create a decimal point next to their seven segment display. However, realizing how redundant and complicated it was to switch each anode, I realized subtraction 0x1000000 from the original seven segment signal would have done the same effect and alot more efficiently.

4. Limitations

Without the possession of a board, the ability to test and improve on specific aspects of the program was difficult with limited submissions. Some submissions were also not regarded after the week period.

5. GAME Instructions

1. Upon start up, PL 1 should be displayed on the screen.
2. Press the center button once (**CAUTION**: Not too hard or it will skip to PL 2 screen)
3. You will be greeted with the display of " 0 0 0 0. " Enter your desired combination using the 4 switches on the right-most side of the board and latching the value to each segment using the directional buttons. (Flip the switches, and press a directional key. Will change the segment corresponding to the button. YOU WILL SEE YOUR CHOSEN NUMBERS BEFORE CONFIRMATION).
4. When you have your desired combination, press the center button once. (**CAUTION**: Not too hard)
5. You will now see the PL 2 screen. **** ON THIS SCREEN**** Please switch off all your switches and press each directional button to erase PL 1's number. (If not done, you will see player 1's chosen combination upon hitting the center button).
6. Press the center button. You should see the display "0 0 0 0."
7. Input your guess as Player 2 using the switches and directional buttons.
8. When ready, press the center button to enter in your guess.
9. Your screen will either say 2 HI or 2 LO if you were incorrect. Press the center button again to confirm.
10. You will see your previous guess on the display now, you can write over this answer. Press the center button to enter in your next guess.
11. A decimal will appear next to your digits when you are inputting in your guess (these are your lost lives). ***** DECIMALS WILL NOT APPEAR ON THE 2 HI OR 2 LO SCREEN *****
12. Repeat steps 9-11 until you win or have the LOSE screen appear.
13. When you win, the left-most LED will start flashing and the number of WRONG GUESSES you've made will appear on the screen. (IF you got it right on the first try, it will be a 0).
14. When you guess 4 incorrect times, you will see the 'LOSE' screen. This screen will appear for a few seconds before restarting the game. You will be greeted with the PL 1 screen again.
15. Enjoy !