Lab 5 Prelab: Process Speed Using Control Interrupts

Owen Blair

# Goal and Background Information

The goal of this lab is to implement an interrupt driven management scheme. This is done by utilizing two interrupts: change notice interrupt and Timer 1 Interrupt flag. When an interrupt is set the CPU will be forced to a specified memory location and will execute the user code and then go back to the background tasks. For example, when the T1IF (Timer 1 Interrupt Flag) is set the CPU will save its context and then jump to whatever code is specified to execute when the flag goes high. At the end of the specified code there is a user written code to reset the flag so it can be used again (in this case in another millisecond). The same goes for the Change Notice Interrupt Flag (CNIF) that will go high whenever the button state changes.

The code the CPU will be forced to when T1IF gets set contains the stepTimer counter variable decrement, stepper state machine, and outputToStepper functions from the previous lab. The difference for this lab is that there will not be a counter for when the button inputs should be sampled as that will be handled by the CNIF and that the if statement that was previously in the while(1) loop will now be executed within the T1IF code and the while(1) loop will be empty. It should also be noted that the stepDelay variable counter and the stepperState variable will need to be a global variable so that the code in the interrupt will be able to access and modify the variable value.

The CNIF code will contain the readButtons and decodeButtons functions from the previous lab and will also include a new delay that will handle button debounce. When the CPU is forced to the change notice section of code by the interrupt handler it will first execute a delay for button debounce, read the buttons, and then decode the buttons and return the global variables to be used in the T1IF code. The global variables

Graphical user interface, diagram, application, Teams

Description automatically generated**Data Flow Diagram**

**Control Flow Diagram**

Diagram

Description automatically generated