Laboratory 5: Blinking LED with Buttons on the Control TM4C123GH6PMI Launch Pad

Experiment Sheet

Purpose

The purpose of this laboratory is to learn how to control LEDs based on button presses using ARM assembly language programming. In this experiment, we will check the state of two switches and control the LEDs accordingly.

Essential Knowledge

GPIO Overview

General Purpose Input/Output (GPIO) pins are used in microcontrollers to interface with external hardware devices. Each GPIO pin can be configured as either an input or output. In this experiment, we will configure the GPIO pins to control LEDs and read the states of switches.

Configuring GPIO Pins

To control an LED or read a switch, the corresponding GPIO pin must be set to the correct mode. Input mode will be used for switches, while output mode will be used for LEDs.

LED Control

To turn an LED on, a high voltage (logical '1') is applied to the GPIO pin. To turn it off, a low voltage (logical '0') is applied. In ARM assembly, this involves writing to the GPIO data register.

Delay Implementation

To create a visible blink effect, a delay routine will be implemented. This can be done using a simple loop to create a time delay.

GPIO Port F Addresses and Pin Assignments

Pin	Connection	Address Value (Mask)	Description
PF0	Switch 2	0x01	PF0 is configured as an input pin (SW2).
PF1	Red LED	0x02	PF1 is configured as an output pin, connected to the red LED.
PF2	Blue LED	0x04	PF2 is configured as an output pin, connected to the blue LED.
PF3	Green LED	0x08	PF3 is configured as an output pin, connected to the green LED.
PF4	Switch 1	0x10	PF4 is configured as an input pin (SW1).

GPIO Port F Register Addresses

- **GPIO RCGC (Clock Control Register)**
  - **Address:** 0x400FE608
  - **Description:** Enables the clock for GPIO ports.
- **GPIO Port F Lock Register**
  - **Address:** 0x40025520
  - **Description:** This register locks the GPIO ports, particularly necessary to unlock **PF0**.
- **GPIO Port F Commit Register**
  - **Address:** 0x40025524
  - **Description:** Allows modifications to the PF0-PF4 pins.
- **GPIO Port F Direction Register**
  - **Address:** 0x40025400
  - **Description:** Determines if pins are input or output.
- **GPIO Port F Digital Enable Register**
  - **Address:** 0x4002551C
  - **Description:** Enables digital functionality for GPIO pins.
- **GPIO Port F Pull-Up Register**
  - **Address:** 0x40025510
  - **Description:** Enables pull-up resistors for specific pins.
- **GPIO Port F Data Register**
  - **Address:** 0x400253FC
  - **Description:** Reads or writes data to the GPIO pins.

Task for Students

Your task is to modify the assembly code to handle the following scenarios based on the switch states:

1. **Both switches pressed:**
   - When **SW1** (PF4) and **SW2** (PF0) are both pressed, **turn on the green LED** (PF3).
   - **Implement a delay** so that the green LED stays on for a noticeable period before checking again.
2. **No switch pressed:**
   - When neither **SW1** nor **SW2** is pressed, **turn on all LEDs** (PF1, PF2, PF3).
   - **Implement a delay** to allow all LEDs to remain on for a visible period.
3. **SW1 pressed (only):**
   - When **SW1** (PF4) is pressed, **turn on the red LED** (PF1).
   - **Include a delay** to make sure the red LED stays on for a clear duration.
4. **SW2 pressed (only):**
   - When **SW2** (PF0) is pressed, **turn on the blue LED** (PF2).
   - **Add a delay** to allow enough time for the blue LED to remain visible.

This experiment will require checking the status of the switches and using conditional logic to control the LEDs accordingly. Additionally, implement delays to ensure that the changes are observable to the human eye.