Assignment 2: LED Pattern Generator

# Learning Objectives:

* Use PLP branch instructions
* Read from a memory mapped I/O device

# The Task:

Write a program in PLP assembly that repeatedly reads the value of the switches (address: 0xf0100000) and displays a pattern on the LED array based on what switches. Each time the switch value is read, the pattern should be displayed (regardless of whether the switch value has changed or not since the last time it was read). The table below indicates the pattern that should be displayed for each possible switch setting:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Switch**  **Number** | **Hexadecimal**  **Switch Value** | **Binary**  **Switch Value** | **Decimal**  **Switch Value** | **LED Pattern** |
| **0** | 0x00000001 | 0b00000001 | 1 | Turn on all 8 LEDs and then turn off all 8  LEDs |
| **1** | 0x00000002 | 0b00000010 | 2 | Turn on all even numbered LEDs and  then turn off all 8 LEDs |
| **2** | 0x00000004 | 0b00000100 | 4 | Turn on all odd numbered LEDs and then  turn off all 8 LEDs |
| **3** | 0x00000008 | 0b00001000 | 8 | Cycle through all 8 LEDs in order with  only one LED on at a time (a marquee) |
| **Other** | Other | Other | Other | All LEDs off |

***Hint:*** Logical shifts are not required to complete this project, but they can be used to make your program shorter and more readable than hard coding every value to be written to the LEDs. Shifts can also be useful to generate the value that you compare with the value of the switches.

# Deliverables:

1. *assignment2.plp*