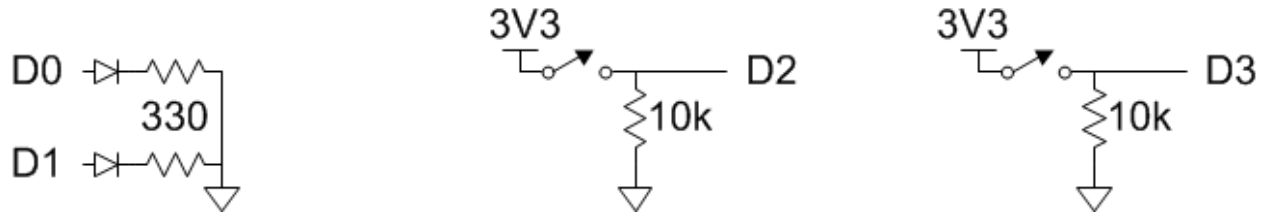


## Lab 8: Simon Game SOLUTIONS

*Digital Design and Computer Architecture: RISC-V Edition (Harris & Harris, Elsevier © 2021)*

1. Please indicate how many hours you spent on this lab. This will be helpful for calibrating the workload for next time the course is taught.
2. [2] Schematic. Remember that pins on adjacent side of pushbutton connect when the button is pressed. Pins on opposite sides are always connected. Any value between  $330\ \Omega$  and  $75\ \Omega$  should work for the LEDs. The switch pulldown value is also approximate.



3. [3] C code

```
// lab7.c
// Josh Brake
// 1/8/2020

// Uses direct register writes instead of EasyREDVIO library

#include <stdio.h>

extern void delay(int delay_ms);

#define LED0 4    // GPIO 4
#define LED1 2    // GPIO 2
#define LED2 5    // GPIO 5
#define BUTTON0 23 // GPIO 23
#define BUTTON1 22 // GPIO 22
#define N 500000

#define L 12
char sequence[L] = "GRGRGRRRGRRG";
char cur_level_sequence[L] = "000000000000";

// Output enable register
volatile unsigned long * GPIO0_output_en = (unsigned long*) 0x10012008;
// Output XOR register (to configure output as active high)
volatile unsigned long * GPIO0_out_xor = (unsigned long*) 0x10012040;
// Function select register
volatile unsigned long * GPIO0_iof_en = (unsigned long*) 0x10012038;
// Output value register
volatile unsigned long * GPIO0_output_val = (unsigned long*) 0x1001200C;
// Input value register
volatile unsigned long * GPIO0_input_val = (unsigned long*) 0x10012000;
// Input enable register
volatile unsigned long * GPIO0_input_en = (unsigned long*) 0x10012004;
```

```

void setup()
{
    // Configure LED outputs
    *GPIO0_output_en |= (1 << LED0) | (1 << LED1) | (1 << LED2);
    // *GPIO0_out_xor |= (1 << LED0) | (1 << LED1);
    *GPIO0_iof_en &= ~(1 << LED0) | (1 << LED1) | (1 << LED2));

    // Configure button inputs
    *GPIO0_input_en |= (1 << BUTTON0) | (1 << BUTTON1);
}

void blink_led(int led_num, int delay_len)
{
    *GPIO0_output_val |= (1 << led_num);
    delay(delay_len);
    *GPIO0_output_val &= ~(1 << led_num);
    delay(delay_len);
}

void digitalWrite(int led_num, int value)
{
    if(value == 1)
    {
        *GPIO0_output_val |= (1 << led_num);
    }
    else if (value == 0)
    {
        *GPIO0_output_val &= ~(1 << led_num);
    }
}

int main()
{
    setup();

    int cur_level = 2;
    int blink_dur = 300;

    while(1)
    {

        // Fill current pattern for comparison
        for(int j = 0; j < cur_level; j++)
        {
            if(j < cur_level)
            {
                cur_level_sequence[j] = sequence[j];
            }
            else
            {
                cur_level_sequence[j] = '0';
            }
        }

        // Display length of test sequence
        delay(4*blink_dur);

        for(int i = 1; i <= cur_level; i++)
        {

```

```

    blink_led(LED2, blink_dur);
}

delay(4*blink_dur);

// Display pattern
for(int i = 1; i <= cur_level; i++)
{
    char cur_led = cur_level_sequence[i-1];
    if (cur_led == 'G') blink_led(LED0, blink_dur);
    else if (cur_led == 'R') blink_led(LED1, blink_dur);
}

char temp_sequence[L];

// Check inputs
for(int i = 1; i <= cur_level; i++)
{
    char button_press_val = '0';
    while(1)
    {
        delay(10);

        volatile unsigned long current_input = *GPIO0_input_val;

        volatile unsigned long button0_pressed = (current_input >> BUTTON0) & 1;
        volatile unsigned long button1_pressed = (current_input >> BUTTON1) & 1;

        if(button0_pressed == 1)
        {
            button_press_val = 'G';
            blink_led(LED0, blink_dur);
            break;
        }
        else if (button1_pressed == 1)
        {
            button_press_val = 'R';
            blink_led(LED1, blink_dur);
            break;
        }
    }

    temp_sequence[i-1] = button_press_val;
}

int pattern_match = 1;
for(int i = 1; i <= cur_level; i++)
{
    if(temp_sequence[i-1] != cur_level_sequence[i-1])
    {
        pattern_match = 0;
        cur_level = 2;
    }
}

// Display success message pattern
if(pattern_match == 1)
{
    for(int i = 0; i < 10; i++)

```

```

    {
        blink_led(LED2, 50);
    }
}

// Next level logic
if(cur_level == L)
{
    cur_level = 2;
    delay(3*blink_dur);
}
else
{
    cur_level++;
}
}
}

```

4. [3] Does your Simon game work? **YES**
5. [1XC] Extra credit