Class Report 4

ELC 4396 System on a Chip

Jordan Edwards
Baylor University

November 2, 2023

Introduction

For this assignment, I added SPI functionality to my system and demonstrated its functionality by building a small demo with the built-in accelerometer and seven-segment displays.

Implementation

In Vivado, I added the necessary components to the MMIO bridge and routed all of the communication wires through to the top level of the project. I was then able to export the bit file in order to continue working on the project in Vitis. Once in Vitis, I opened the example files to understand the exact commands that the accelerometer was expecting. With this information, I modified his function in order to return my own data type, a Vector3. This data type allowed me to easily access all three components of the acceleration data as well as normalize the result in order to get a simple unit vector in the direction of "up." I used this to simply decide whether the X or Y component was larger, and then I used that information to light up a single edge of a single digit of the seven-segment display.

```
19  Vector3 getAccel(SpiCore *spi_p){
20      const uint8_t RD_CMD = 0x0b;
21      const uint8_t DATA_REG = 0x08;
22      const float raw_max = 127.0 / 2.0;  //128 max 8-bit reading for +/-2g
23
24      int8_t xraw, yraw, zraw;
25      Vector3 val;
26
27      spi_p->set_freq(400000);
28      spi_p->set_mode(0, 0);
29      // read 8-bit x/y/z g values once
30      spi_p->assert_ss(0);    // activate
31      spi_p->transfer(RD_CMD);  // for read operation
32      spi_p->transfer(DATA_REG);  //
33      xraw = spi_p->transfer(0x00);
34      yraw = spi_p->transfer(0x00);
35      zraw = spi_p->transfer(0x00);
36      spi_p->deassert_ss(0);
37      val.setX((float) xraw / raw_max);
38      val.setY((float) yraw / raw_max);
39      val.setZ((float) zraw / raw_max);
40      return val;
41  }
```

Figure 1: Get acceleration function

```
uint8_t target;
Vector3 accel = getAccel(&spi);
accel.normalize();
```

Figure 2: Simple fetch and normalization with Vector3

```
if(abs(accel.getX()) > abs(accel.getY())){
    if(accel.getX() > 0){
        target = top;
    }
    else{
        target = bottom;
    }
}
else{
    if(accel.getY() > 0){
        target = left;
    }
    else{
        target = right;
    }
}
sseg.write_1ptn(target, 0);
sleep_ms(100);
```

Figure 3: Directionality tests

Results

The code ran successfully on the board; when tilted, the correct segments on the seven-segment display were illuminated. Because there is no filtering and only accelerometer data is used, shaking the board can cause the display to change rapidly even if it is tilted at a steep angle.