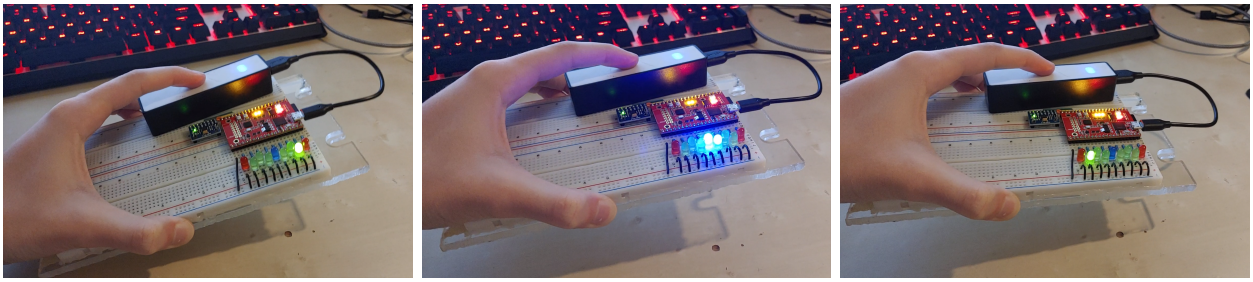
Digital Bubble Level

Vladislav Pomogaev - 26951160

September 27, 2021

1 Introduction

This device forms a digital bubble level; also called a spirit level. It can be helpful in levelling things horizontally.



(a) (b) (c)

Figure 1: Photo stills of the device in action. When you tilt the device to the left as in a), the LEDs to the right light up. As you tilt the device more to the right, the lit LED moves from right to left as in b) and c). When level, the blue center LED lights up.

This project consists of a synthesised Verilog design running on an Efinix Xyloni FPGA development board. An accelerometer (MPU-6050) and 9 LEDs are connected to this board.

The Verilog design consists of an FSM (Finite State Machine), and an I^2C controller IP block from Efinix. The FSM can send commands to the accelerometer through the I^2C controller, which acts as the master. Upon reset, the FSM wakes the accelerometer by writing to a register. Then, in a loop, the FSM requests the acceleration measured in one axis from the accelerometer, and converts the acceleration to a grey-code-like signal which is then output to the LEDs. All of this is done while implementing the communication protocol of the I^2C controller. The FSM must request data in the correct format, provide slave address, data, command byte, and number of data bits, wait for response, and check for errors during every read and write command.

2 Overview

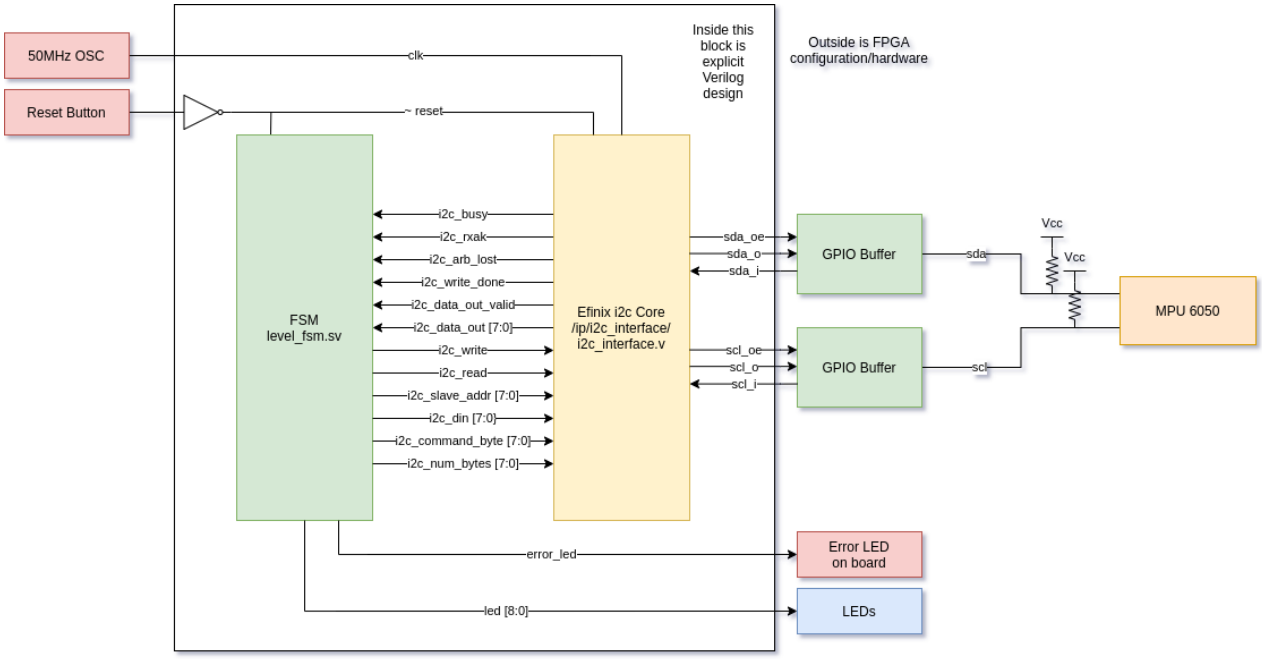


Figure 2: Block diagram of the FSM and how it forms the bubble-level system. The level FSM talks to the I2C core, which communicates to the MPU via a set of buffered GPIO blocks on the FPGA. A reset button resets all FSMs to their initial state. The design runs is compiled for a 50MHz clock constraint.

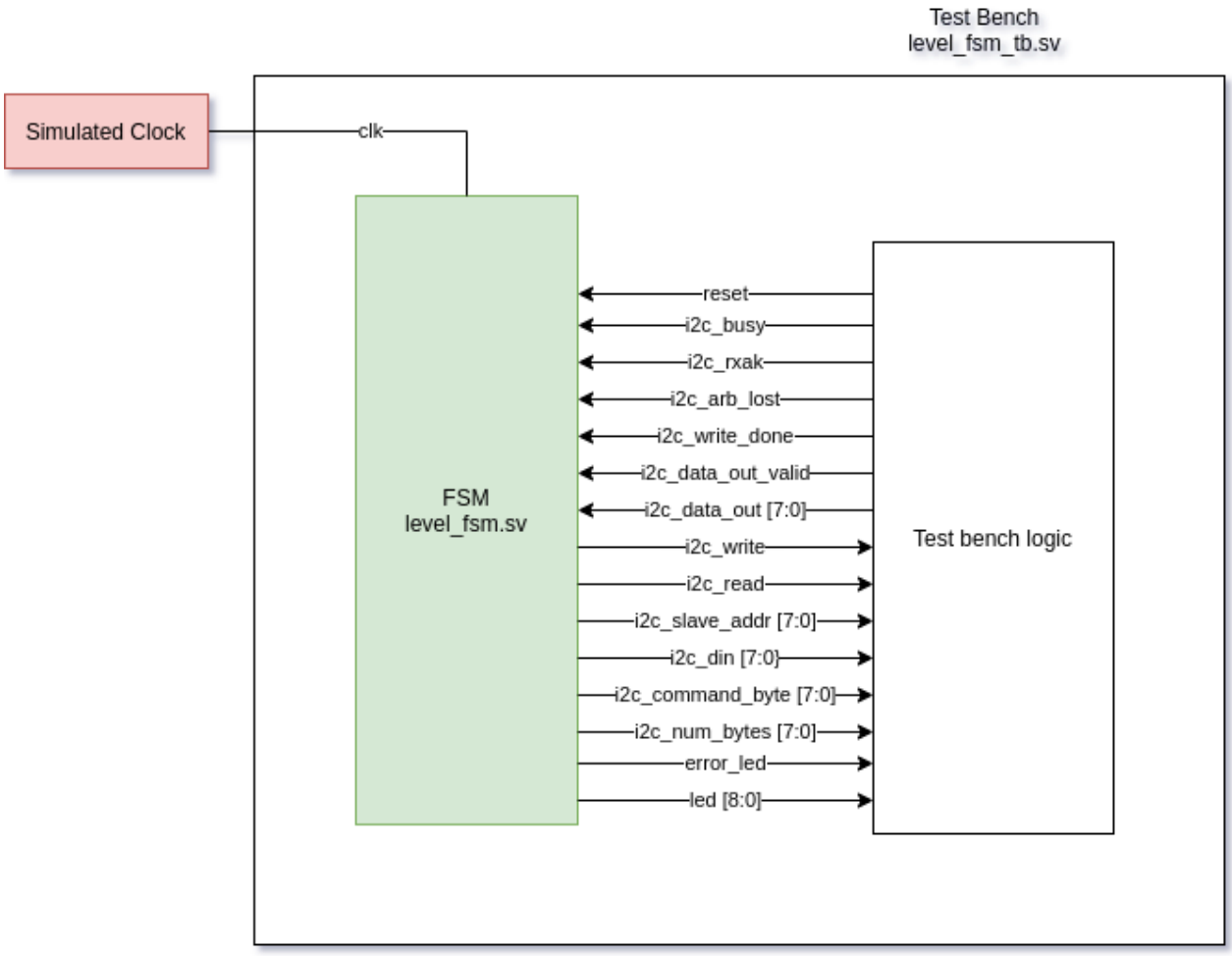


Figure 3: Block diagram of how the level FSM (not the Efinix i2c core) is connected to the test bench. The test bench is a simple piece of logic that provides timed inputs to the level FSM and asserts that the outputs of the FSM are what are expected.

The I2C core provided by Efinix was autogenerated and came with it's own test-bench. Since I did not write this IP or the test bench, I do not include the source code for this report.

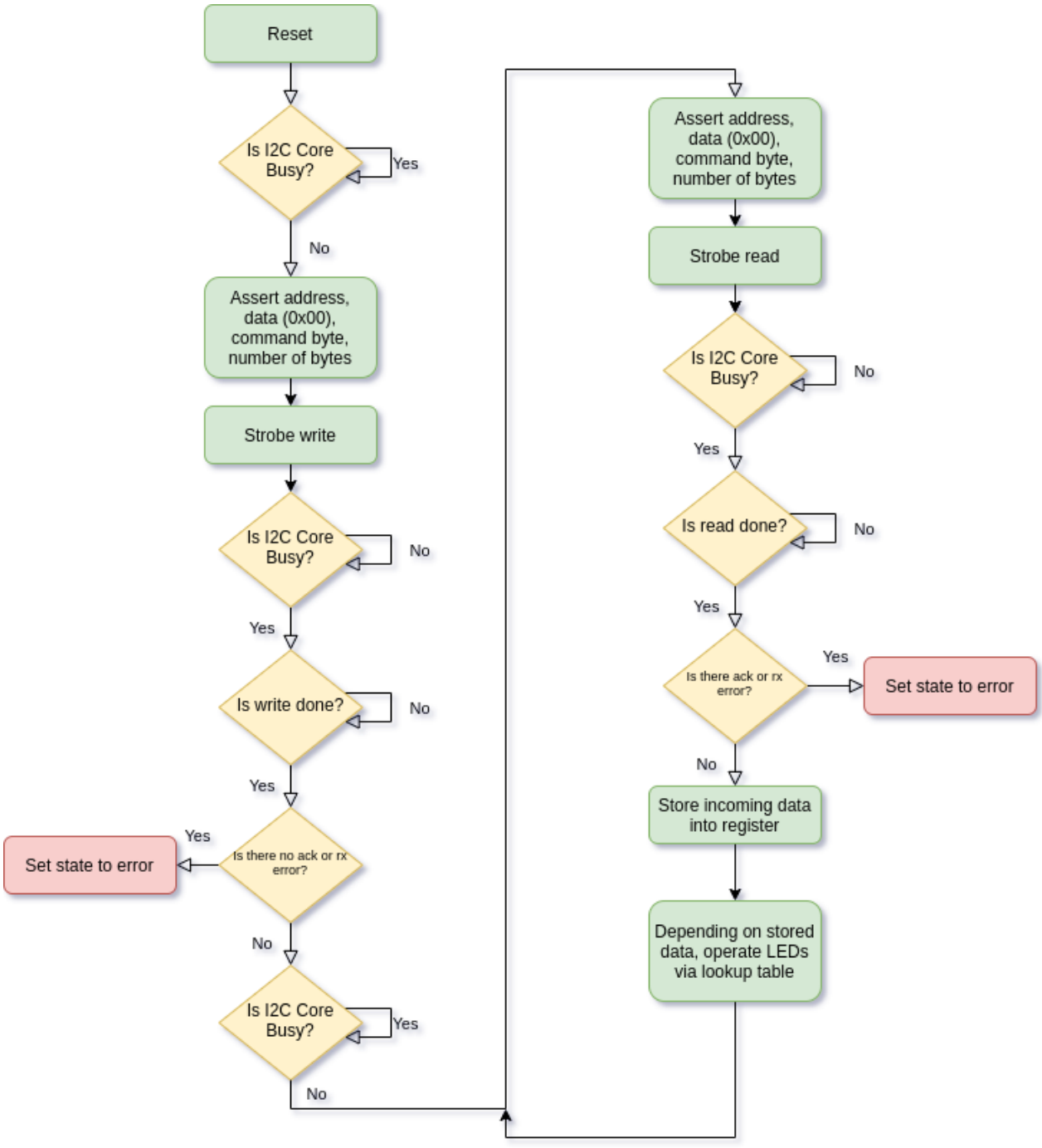


Figure 4: Flow diagram of the internal functioning of the FSM. The assertion of the addresses, data, command byte is done even though it is slightly redundant in this situation to allow for easy expansion of functionality by changing those lines to be tri-state and allowing other modules to write to them.

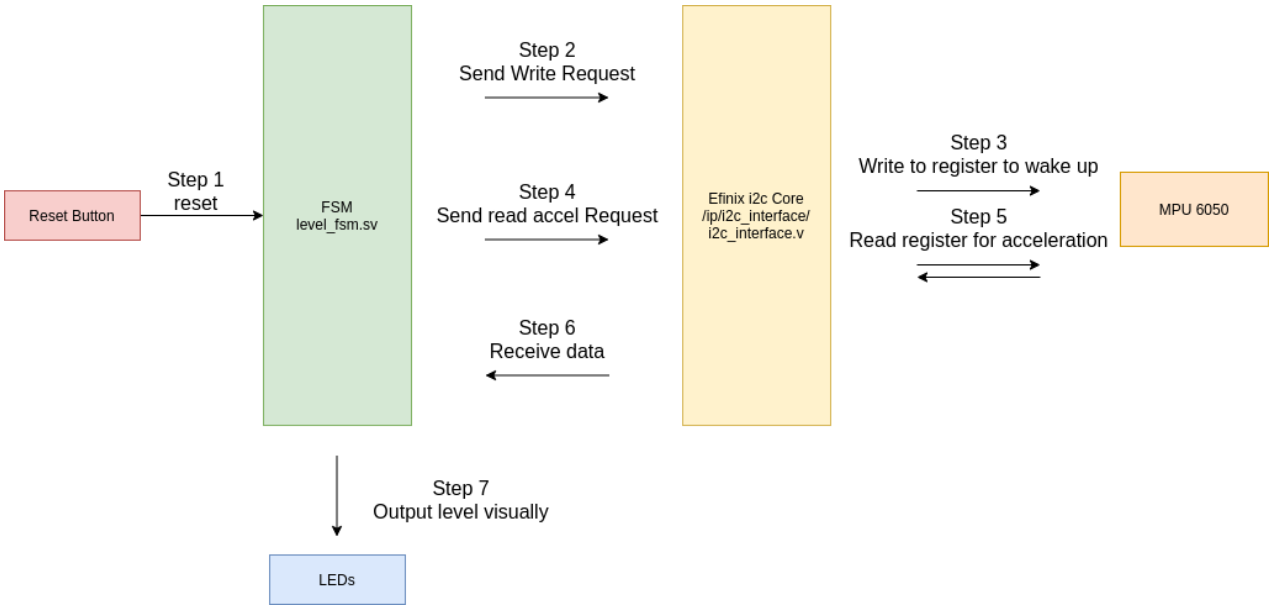


Figure 5: Data flow diagram for the FSM.

3. A test bench and at least one module for your FSM. You can have more than two modules if your FSM requires it. Define the input and output of each module and a purpose/description of each state for each module. In your test bench, please put comments on how this will test your FSM. [20] 7. A copy of your code (in Font Size 8) [10] [as a separate file] 8. Simulation waveform results [30].