**Laboratory 4**

**Stepper Motor Operation**

**Post-Lab**

**Team Number 5**

**Authors**

**Christopher Wilcox**

**James Nystrom**

**Christopher Wilcox**

**James Nystrom**

**LAB 4 – Stepper Motor Operation**

For this laboratory, we entered the unknown realm of stepper motors. We used a Vextra stepper motor coded a subroutine called step to control it using our board. We demonstrated the use of one phase, two phase, and half step modes of the stepper motor through the use of phase and current.

Our subroutine functions by taking in the values in R0 and R1 and outputs the next appropriate stage of the motor. We use a counter that increments/decrement by 1 from 0-4 or 0-7 depending on the mode. The mode also determines how this counter is used with the lookup. For mode 00 (One Phase On), the count is doubled to be used as an index for LUT items 0,2,4,6. For mode 01 (Two Phase On), the count is doubled and then incremented by one to be used as index for items 1,3,5,7. For the mode 10 (Half-Step), we use the count as is as we need every index of the list. Finally, if the Mode 11 is used, we do nothing as there is no implementation included in our subroutine for a fourth mode.

We encountered multiple issues in producing our code. The issue that caused the largest time sink for us was the workings of a subroutine in an interrupt. We were not properly storing the link register prior to the subroutine, and for this reason, we were being directed to bad memory. This caused a seemingly random exception that put us into Data Abort, but after close inspection and much time spent on testing, we found the source of the problem and were able to find a remedy.

We also had some issues with our look-up table for the motor. We had mixed convention partway through the orginal calculations, and for this reason, had incorrect values. Below is the final implementation and operation of the stepper motor. Steps 1,3,5,7 are used for ‘One Phase On’ and steps 2,3,6,8 are used for ‘Two Phase On.’

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Half-Step** | |  |  |  |  |  |  |  |  |
|  | **Step** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **Top-Bottom** | **PHASE1** | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
|  | **I1-1,I0-1** | 00 | 00 | 11 | 00 | 00 | 00 | 11 | 00 |
| **Left-Right** | **PHASE2** | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
|  | **I1-2, I0-2** | 11 | 00 | 00 | 00 | 11 | 00 | 00 | 00 |

We also attempted to increase the step rate to 800/sec. By doing this, we found that the half-step mode still functions, while both single stepping modes were broken. This happens because we are asking the motor to change current direction at too rapid of a rate. It cannot keep up, and fails to properly function.

**Attached You Will Find:**

1. **Main.s** – this file contains the main section of code
2. **Exceptions.s** - this file is included as I modified it and placed IRQ\_Handler in a separate file
3. **Step.s** – this file contains the Step subroutine
4. **IRQ\_Handler.s** – this contains the IRQ\_Handler, which handles the 7-segment display and the steppper motor function.
5. **Quartus FPGA Schmatic** – This file shows the FPGA configuration as used for this laboratory