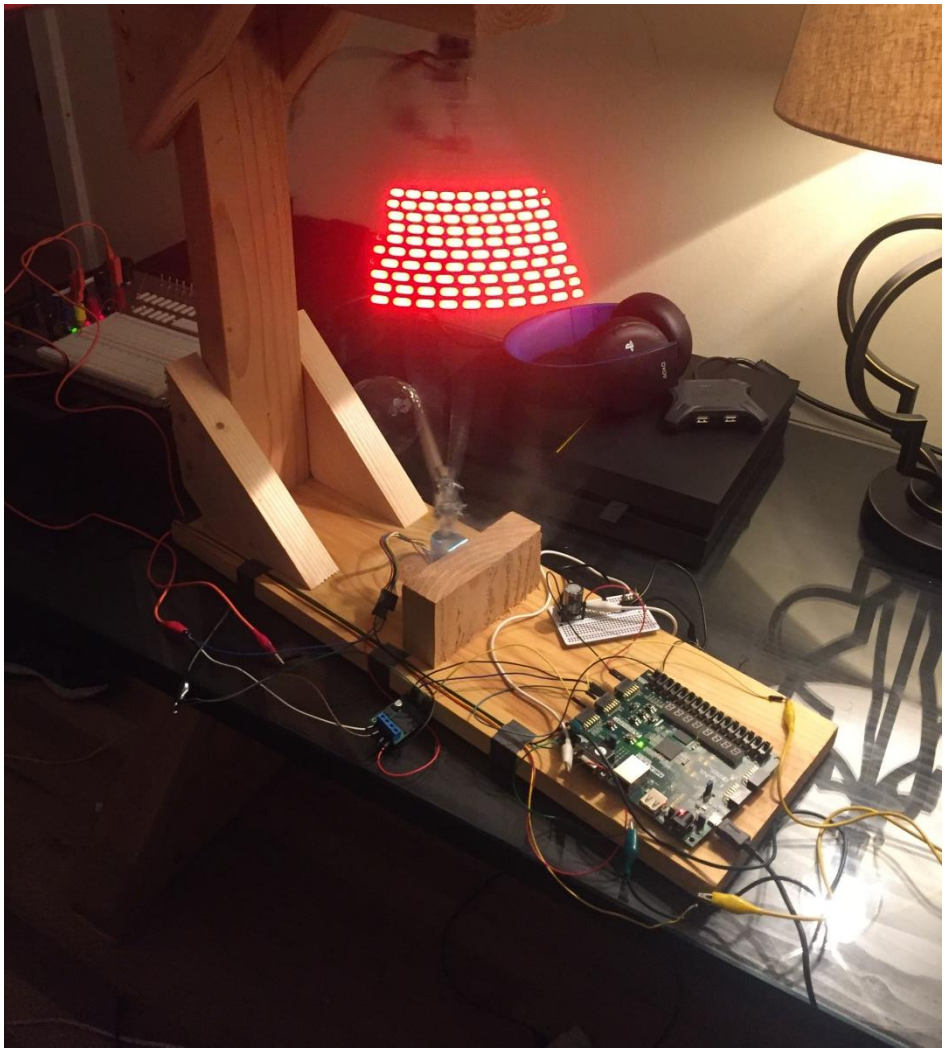
ECE 540- Final Project

Persistence of Vision



Teammates-

Brandon Biodrowski

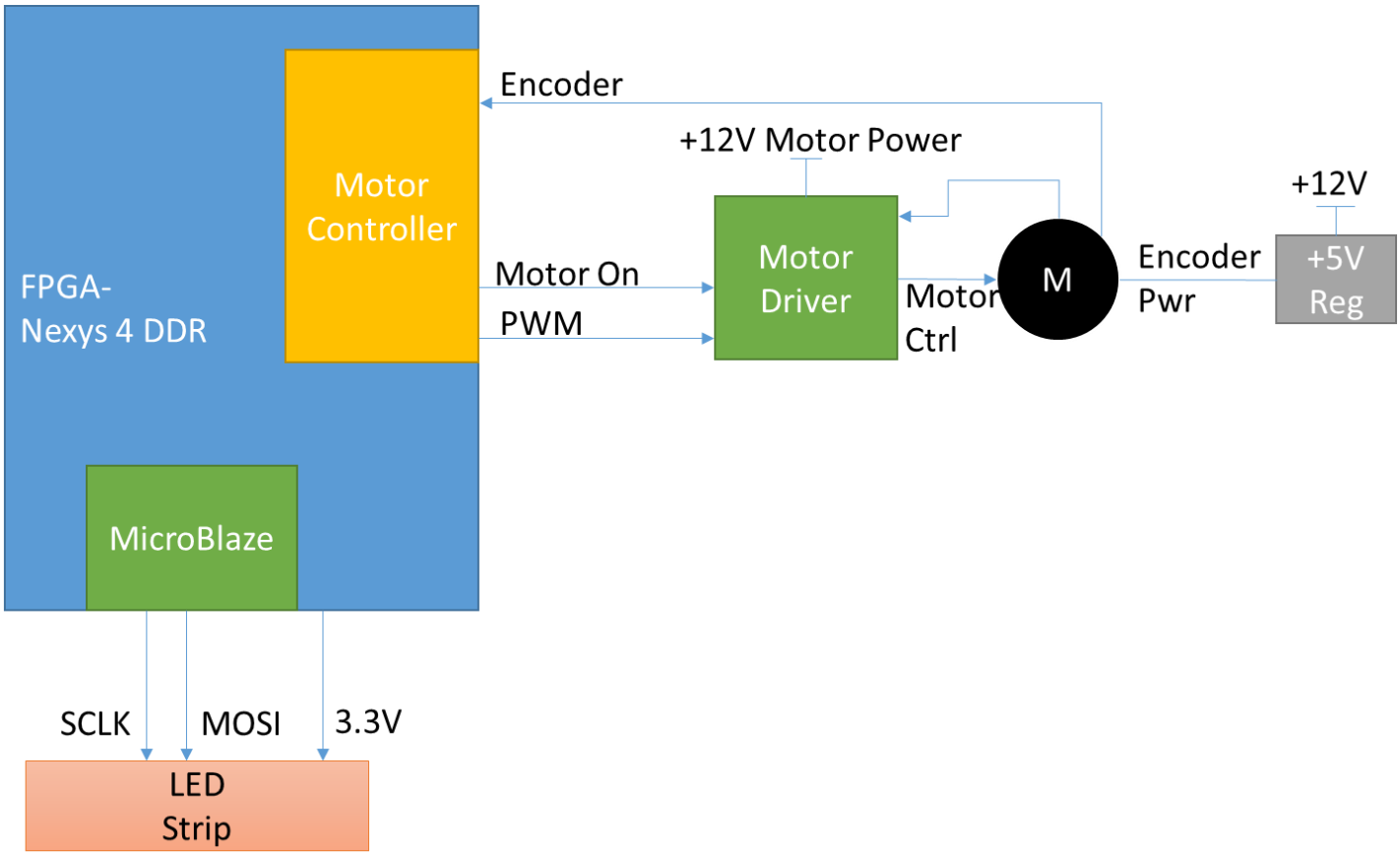
Nitish Kumar Marikal

Venkata Anil Kumar Viswanadha

1. Introduction-

Persistence of vision refers to the optical illusion whereby multiple discrete images blend into a single cohesive image in the human mind. Similar concept is implemented through our project. An acrylic ring is rotated on Y-axis and a LED strip is pasted on the thickness across the ring. And when the ring is rotated speed enough the programmed LEDs would seem like displaying an image but they were displaying only a part of the image at a particular point in time.

2. Block Diagram-



3. Materials Required-

In order to implement the project, following were the materials used-

- Nexys 4DDR FPGA,
- Microblaze- Softcore microprocessor,
- Brushed DC motor with an encoder,
- Motor Driver,
- LED strip,
- Slip ring,
- Acrylic Ring,
- Wood to build the supporting structure.

4. Description-

POV was divided into four parts,

- Motor Control,
- LED programming,
- Building the supporting Structure,
- Communication between Microprocessor and LED strip,

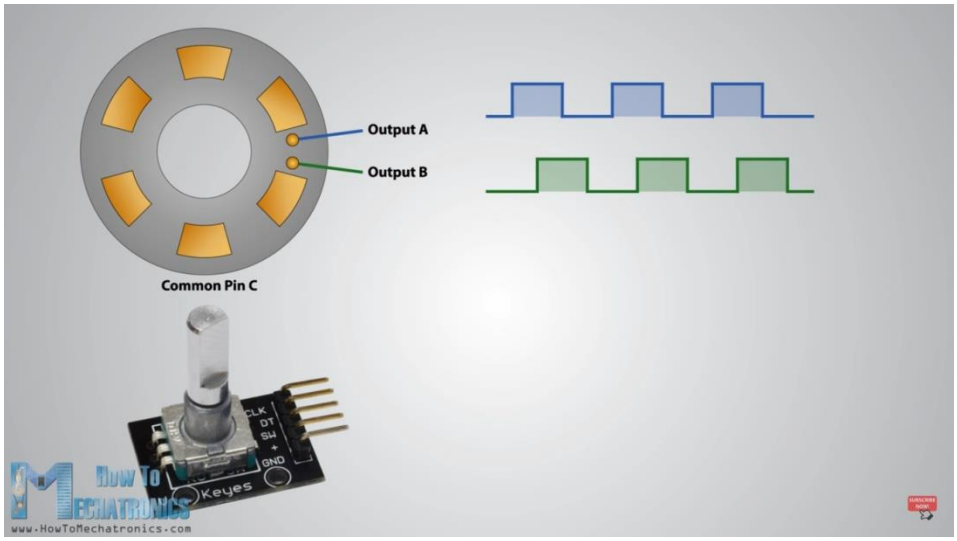
4.1 Motor Control-

Motor control was implemented in three steps through FPGA,

i. Reading the encoder,
ii. A control loop,
iii. PWM wave generator.

4.1.1 Reading the Encoder-

Encoder known as rotary encoder or shaft encoder is an electro-mechanical device that converts the angular position or motion of a shaft or axle to an analog or digital code. The encoder used was a 48 CPR (Cycles per Revolution).



48 pulses are considered for both the outputs from the encoder, with both the edges. Two outputs are generally used to determine whether the motor rotates in clockwise/anti-clockwise direction with phase difference. As our application can run in any of the direction, a single output with single edge is considered i.e. 12 pulses are considered for one complete rotation of motor.

- 'read_encoder' block reads the output from motor encoder and counts the 'encoder period' which will be an output to Control loop.

4.1.2 Control Loop-

This loop was used to control the speed of the motor using a PI control method and the outputs were used to drive a PWM signal to drive the motor. Change in duty cycle of the PWM signal resulting in varying the speed of the motor.

4.1.3 PWM generation-

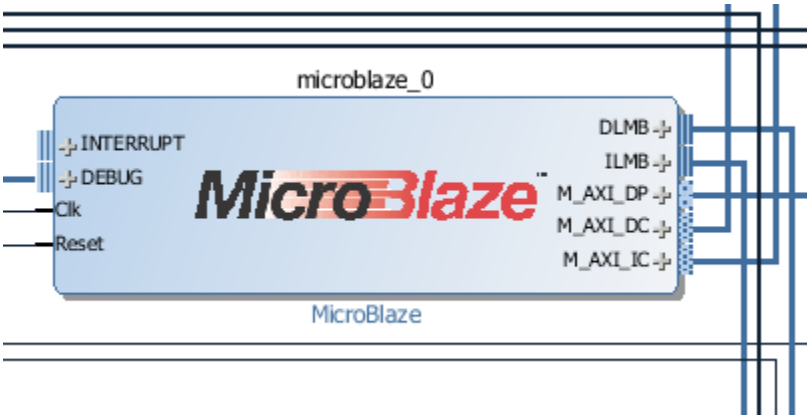
A PWM signal is used to drive the motor. Based on the output generated from the control loop, the duty cycle is varied. A duty cycle varied, the speed of the motor is controlled. A 50% duty cycle means, 50% of time current is supplied to motor and 50% of time its cut off. As the duty cycle is increased the longer is current provided and thus the speed of the motor increases. And when decreased, the speed of the motor reduces.

4.1.4 Motor Driver-

A motor driver was used in this project, which helped in replacing the discrete components required for the motor to run smoothly. The driver also helps in easy plug and play option with the motor.

4.2 LED programming -

A microprocessor is used to program the LED output and in this project when have implemented Microblaze, a Xilinx 32 bit softcore processor.



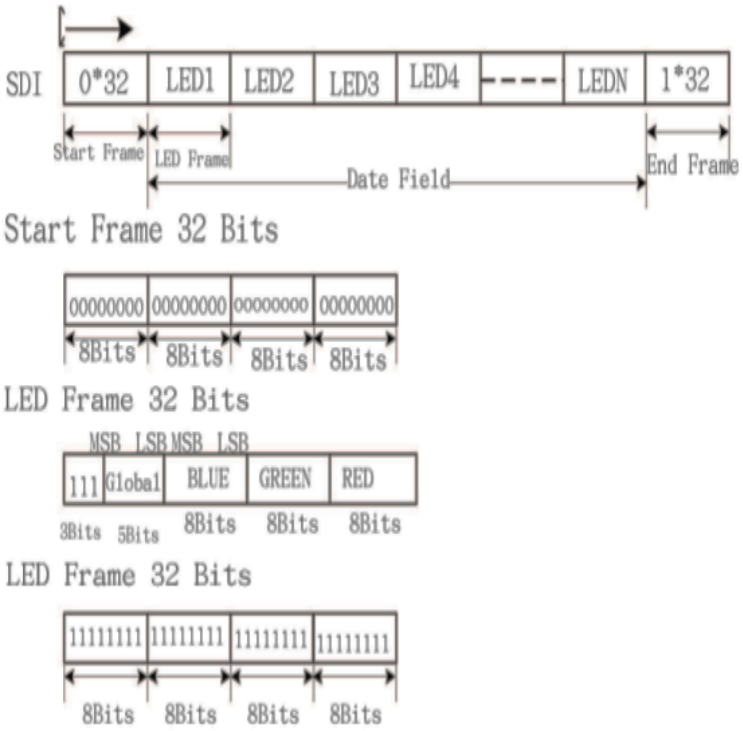
The advantages of using microblaze than other processors were-

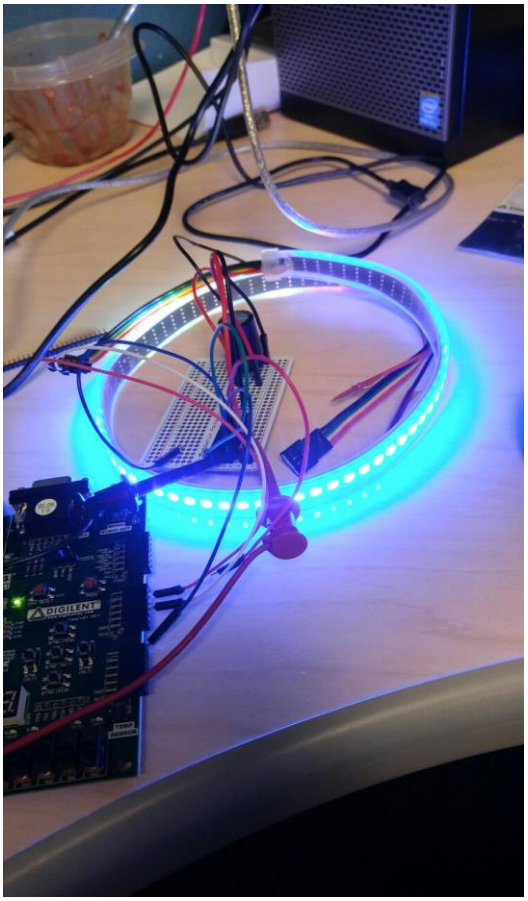
- Highly configurable peripheral configuration using IP cores in Vivado,
- Plug and play ease,
- Drop in SPI IP block and use SPI API in Xilinx SDK,
- Contains 32KB SRAM, which can be used to hold an image,
- Has an additional 16KB cache configuration.

To program the LEDs following were the steps implemented-

a) Continuous strings of data was sent to address a particular LED,
b) To begin LEDs a frame of all 0's was sent,

c) Then required data was sent continuously,
d) To end a frame of all 1's were sent.

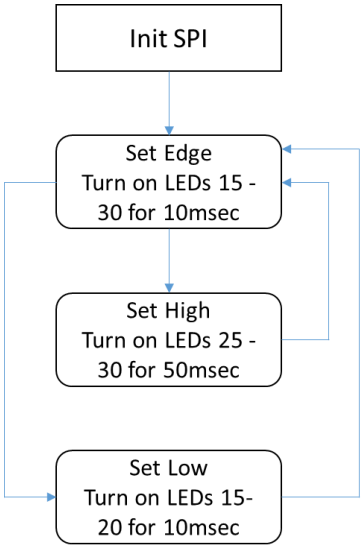




To program a particular LED, in the data SDI, LED Frame would be 8'h0 or else 32 bit data to control the brightness, level of red, blue and green. In this project brightness was set to 1/6th of full brightness.

LEDs consumed low power ~10mA/LED and thus could be run by FPGA with 3.3 V

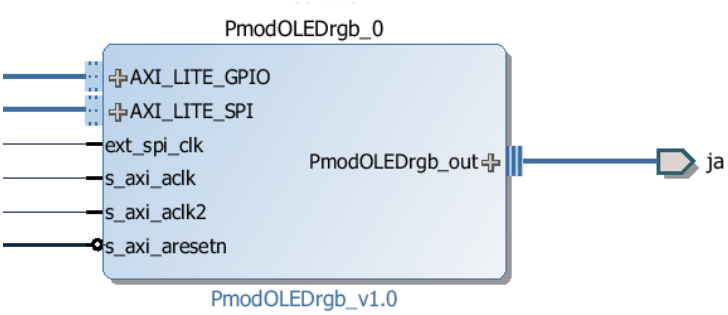
4.2.1 LED control for Square Wave-



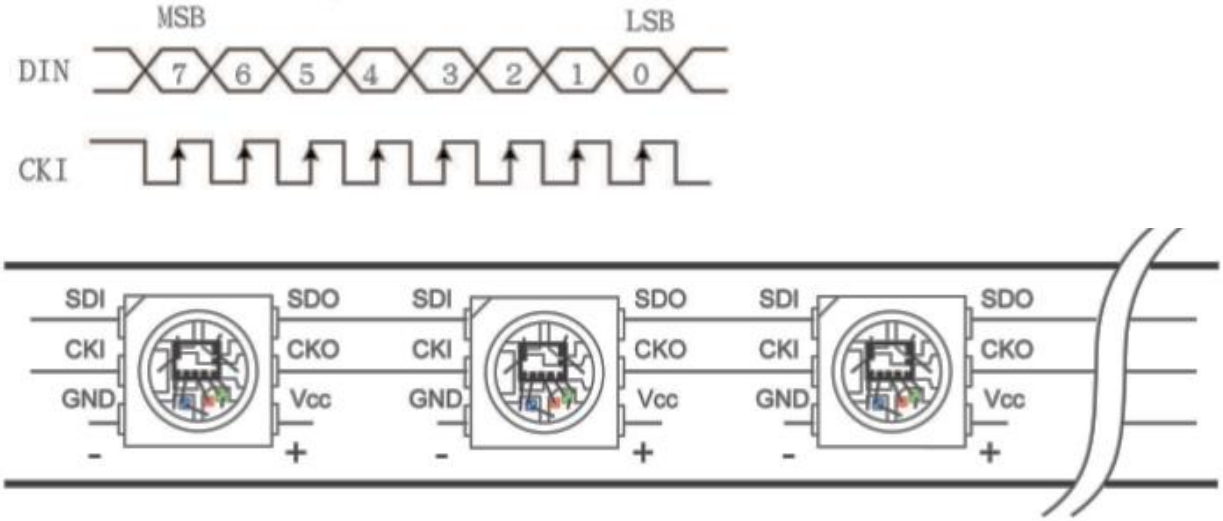
@500RPM or ~ 120msec per rev
Rising edge on for 1/12 of the rev

4.3 Communication between Microprocessor and LED-

To transfer the data from microblaze to LED strip, SPI interface was implemented.

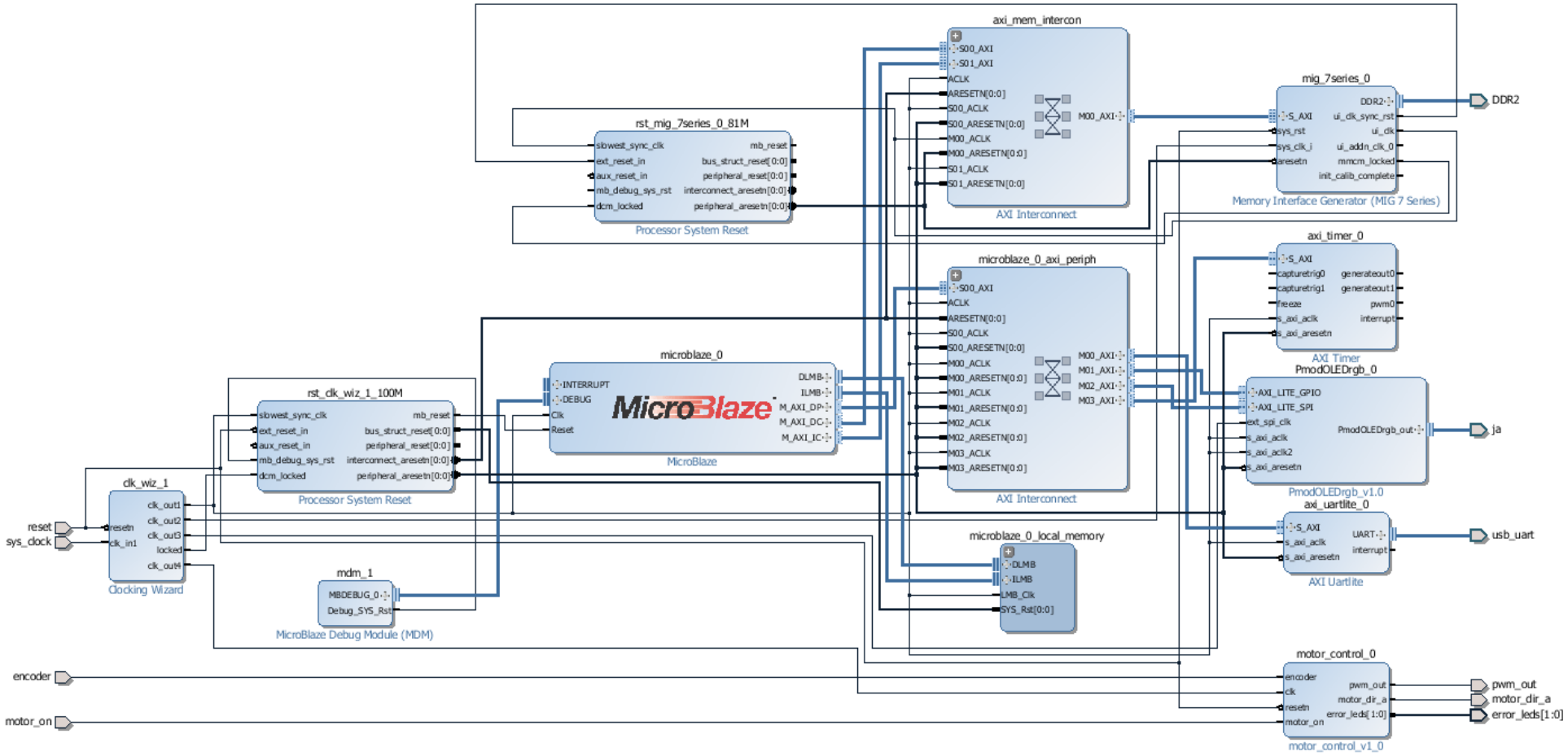


To drive the LED a clock signal with 3.125 MHz frequency was used-



5. System On Chip-

Following is the Block Diagram implemented for the project



6. Building the supporting Structure-

The ring of the structure was built from Acrylic sheet using laser cutter. The model required for laser cutting was built in inkscape, the model used is shown below:

As the whole ring is of dia 14 inches (for outer circle) the sheet available in the LID prototyping Lab was only 16x12(inches), the ring was cut in half and two half's were bolted with clips.

The supporting structure was built from wood pieces and the result was-

To avoid wire from wrapping, slip ring was used which needed to be placed above the shaft of motor.

To make the shaft rotate the acrylic ring, shaft of motor was glued with the clips with the ring. And the motor was clamped to another wood piece with the motor clamps.

The structure supported the structure before the LED strip was pasted to the ring and motor run with speed greater than 400 rpm, but after the strip was glued to the ring, speeds higher than 450 rpm were difficult to balance due to imbalance weights on the ring. This limited us with a motor speed of 400 rpm.

While making the electrical connections- Supply to all the components were given from three different supplies, as the system was not running from a single supply. Even though the encoder was a part of the motor it required a different power source. Programming the LED to produce an image, which required the motor to spin faster than the present speed, would make the structure unstable. Communication between Microcontrollers from (Microblaze) SDK to the LED strip took longer than we anticipated.

7. Conclusion and Things we learned-

LED strip was spun fast enough to blend multiple images as a single image seen by the human eye. And to achieve this we learned-

- Programming Microblaze (Softcore Microprocessor) using C language onto an FPGA,
- Communication interfaces such as UART and SPI,
- Programing an LED according to the required output,
- Use of various IP blocks provided by Vivado,
- How to build a model in inkscape for laser cutting and 3D modelling in tinkercad (even though 3d model was not used in our project)
- How to use pmods- OLED, DPOT.

Work Distribution-

- ✓ Motor Control- Nitish, Brandon
- ✓ Microblaze, SPI- Brandon, Anil
- ✓ Mechanical and Electrical structure- Anil, Brandon and Nitish

8. References-

1. https://www.xilinx.com/support/documentation/sw_manuals/mb_ref_guide.pdf
2. https://cdn-shop.adafruit.com/datasheets/APA102.pdf
3. https://www.pololu.com/product/3214
4. https://www.pololu.com/product/1451
5. https://hackaday.io/project/8881-yapg
6. https://reference.digilentinc.com/learn/programmable-logic/tutorials/nexys-4-ddr-getting-started-with-microblaze/start
7. https://reference.digilentinc.com/learn/programmable-logic/tutorials/pmod-ips/start
8. https://reference.digilentinc.com/learn/programmable-logic/tutorials/htsspisf/start
9. https://reference.digilentinc.com/learn/programmable-logic/tutorials/nexys-4-ddr-programming-guide/start