**California State University,**

**Northridge**



**System on Chip with Lab**

**ECE 520**

**Mini Project 1**

**May 17, 2021**

**Spring 2021**

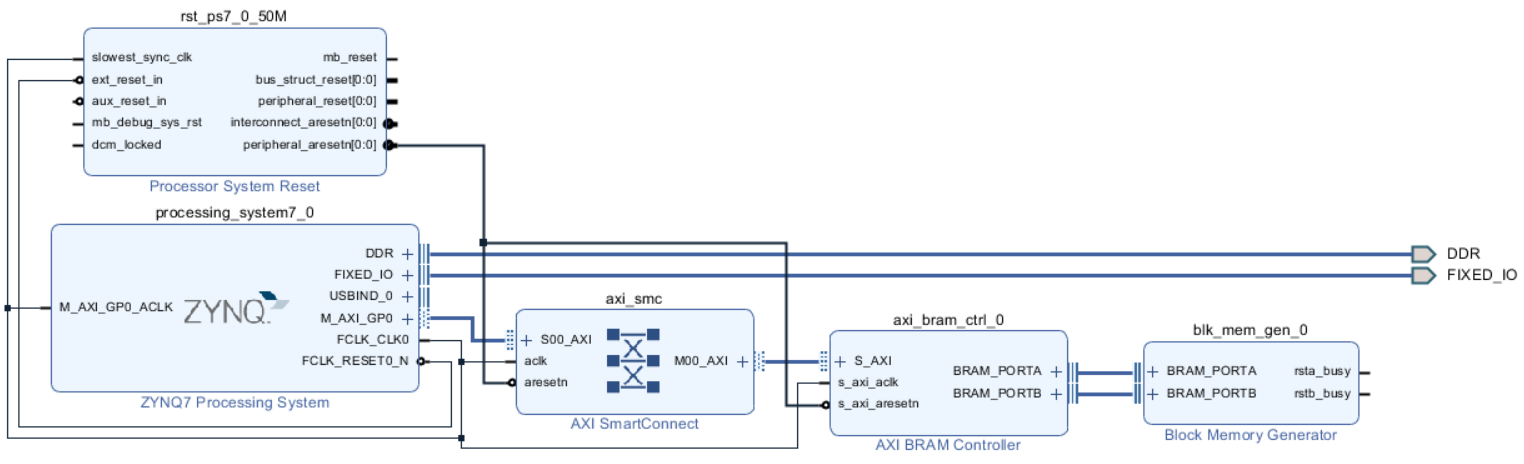
**Eros Lachica**

**Department of Electrical and Computer Engineering**

**Instructions**

Mini Project 1 focuses on creating a perioding waveform using five parameters which are the Final value, Drop value, Step horizontal value, step vertical value, and period. The Zynq Processing System (PS) and the block RAM (BRAM) was used. The BRAM stored the values of the parameters and arithmetic operations are used to modify the values stored in the BRAM; the values are read from memory to the terminal in the Xilinx SDK to be plotted. The values then can be parsed into any application that will be able to graph a stair step graph. In this case Excel was used.

The block diagram is shown below which shows the inclusion of the AXI BRAM Controller and the Block Memory Generator. The only change needed was to reconfigure the Block Memory Generator to be the memory type of “True Dual Port RAM”. A bitstream is then generated for the block design and the exported to hardware for the software application.



*Figure 1: Block Diagram of Waveform Generator*

**Results**

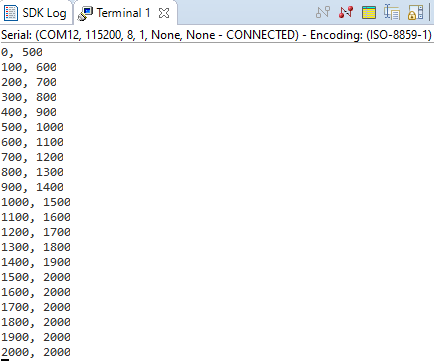
**int** finalValue = 2000;

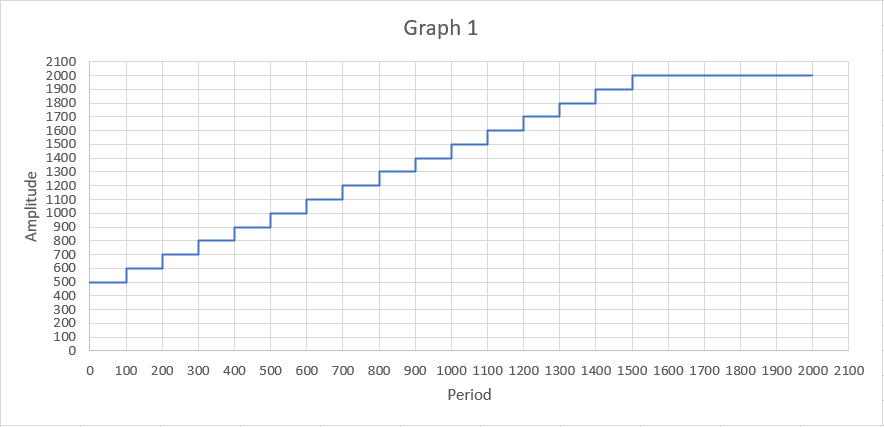
**int** dropValue = 1500;

**int** stepHorizontal = 100;

**int** stepVertical = 100;

**int** period = 2000;





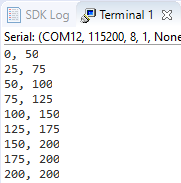
**int** finalValue = 200;

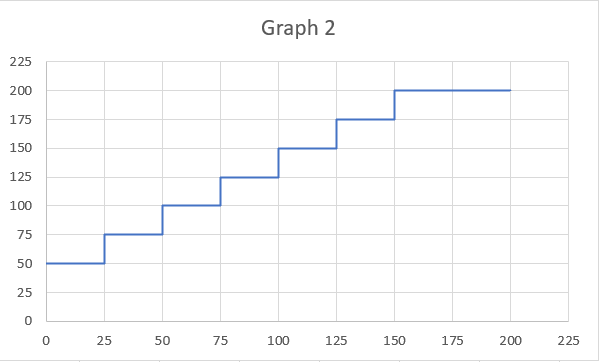
**int** dropValue = 150;

**int** stepHorizontal = 25;

**int** stepVertical = 25;

**int** period = 200;





**Source Code**

**#include** <stdio.h>

**#include** "xil\_printf.h"

**#include** "xil\_io.h"

**#include** "xparameters.h"

**#define** BRAM\_BASEADDR XPAR\_AXI\_BRAM\_CTRL\_0\_S\_AXI\_BASEADDR

// Parameters for Waveform Generator

// Can be changed according to user specifications

**int** finalValue = 200;

**int** dropValue = 150;

**int** stepHorizontal = 25;

**int** stepVertical = 25;

**int** period = 200;

// Function that will generate the waveform

uint32\_t **generateWaveform**(uint16\_t finalVal, uint16\_t dropVal, uint16\_t horizontalVal, uint16\_t verticalVal, uint16\_t per) {

// the variables will be written into memory cells of BRAM

// addresses are accessed via an 8 bit line

Xil\_Out16(BRAM\_BASEADDR, per);

Xil\_Out16(BRAM\_BASEADDR + 2, finalVal);

Xil\_Out16(BRAM\_BASEADDR + 4, dropVal);

Xil\_Out16(BRAM\_BASEADDR + 6, horizontalVal);

Xil\_Out16(BRAM\_BASEADDR + 8, verticalVal);

**int** y = finalVal - dropVal;

**int** x = 0;

Xil\_Out16(BRAM\_BASEADDR + 10, x);

Xil\_Out16(BRAM\_BASEADDR + 12, y);

// Incrementing the addresses for the x and y axis values

**for** (**int** i = 0; i < (per/horizontalVal)\*4; i = i+4) {

x = x + horizontalVal;

y = y + verticalVal;

// change address for x and y if value greater than finalVal

**if**(y > finalVal){

y = finalVal;

}

Xil\_Out16(BRAM\_BASEADDR + 14 + i, x);

Xil\_Out16(BRAM\_BASEADDR + 16 + i, y);

}

**return** BRAM\_BASEADDR;

}

// Parsing the BRAM\_BASEADDR

// Reads the values written into the memory cells of the BRAM

**void** **displayData**(uint32\_t addr) {

**for**(**int** i = 0; i <= (Xil\_In16(addr+0)/Xil\_In16(addr+6))\*4; i = i+4) {

xil\_printf("%d, %d\n\r", Xil\_In16(addr+10+i), Xil\_In16(addr+12+i));

}

}

**int** **main**() {

// Generates the waveform using Parameter Values

// and display the plot values in terminal

displayData(generateWaveform(finalValue, dropValue, stepHorizontal, stepVertical, period));

**return** 0;

}