Lab Report #3

Designing Clock of 11MHz Using Software and Hardware

Amilton Pensamento

Derek Stahl

1. Objectives

The objective of this lab was to generate an 11MHz square wave with a 50% duty cycle using a hardware technique and software technique. With the software technique, we were allowed to use any of LPC1769's clocks as well as its PLLs and dividers. For the hardware technique, we were given a 4MHz crystal and then tasked with using dividers and a PLL to generate an 11MHz clock signal.

2. Design

Hardware

For the hardware design, firstly we focus on getting our crystal oscillator circuit to produce a 4 MHz frequency. Then we used a clock divider in asynchronous mode to divide the oscillator's frequency by 4, then using the PLL chip with a low-pass filter circuit and another clock divider in synchronous mode with an AND gate to multiply the frequency by 11 to finally obtain an 11 MHz output. To achieve the required frequencies, we initially calculated the values for the resistors and capacitors to use in the circuit.

Hardware Equations:

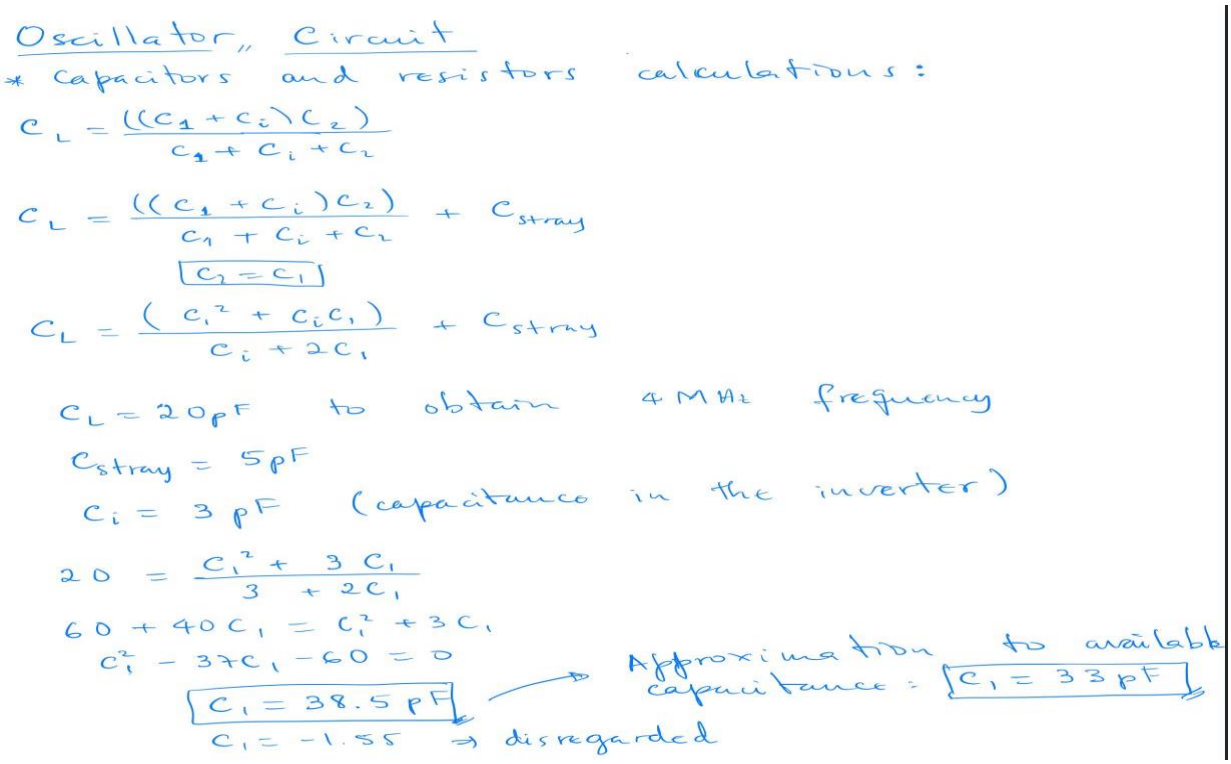

Figure 1: Oscillator circuit equations for 4MHz

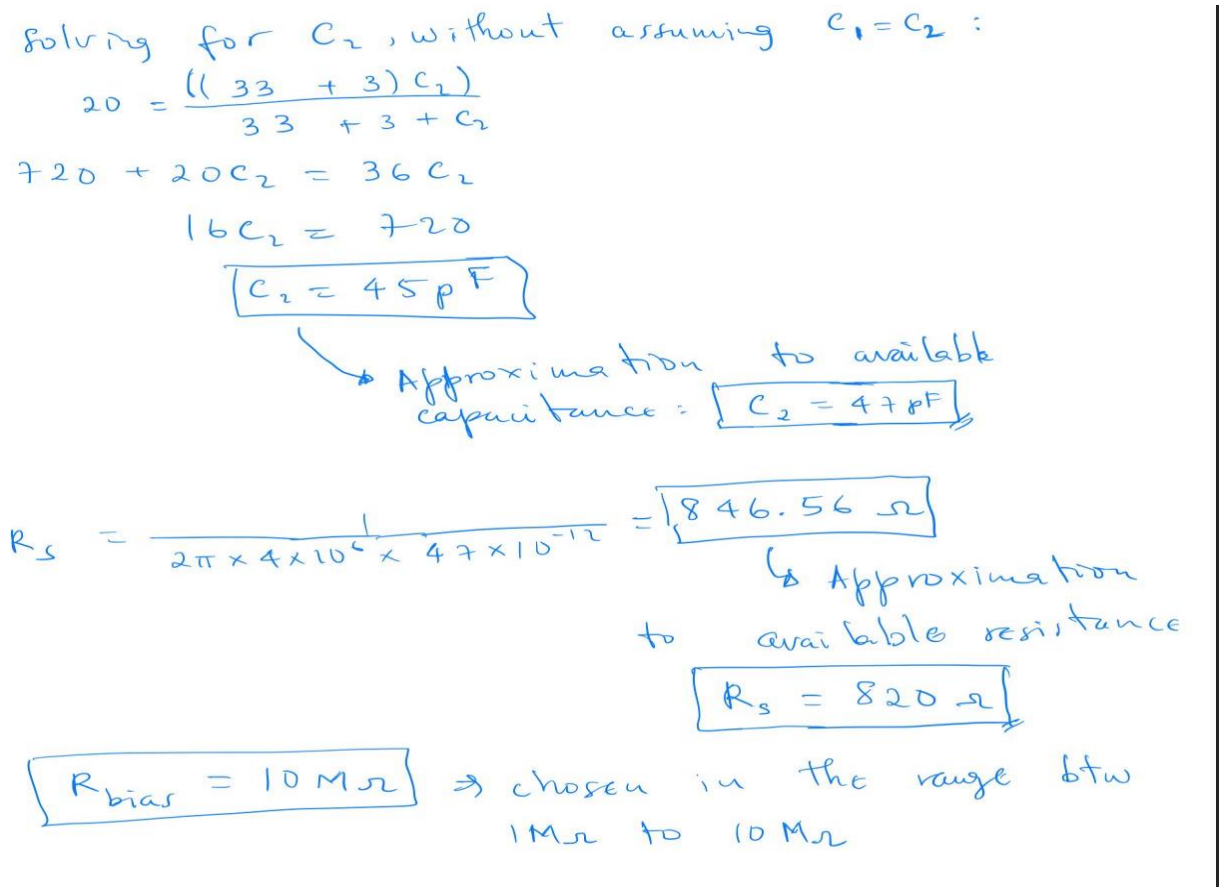

Figure 2: Continuation of Oscillator circuit equations

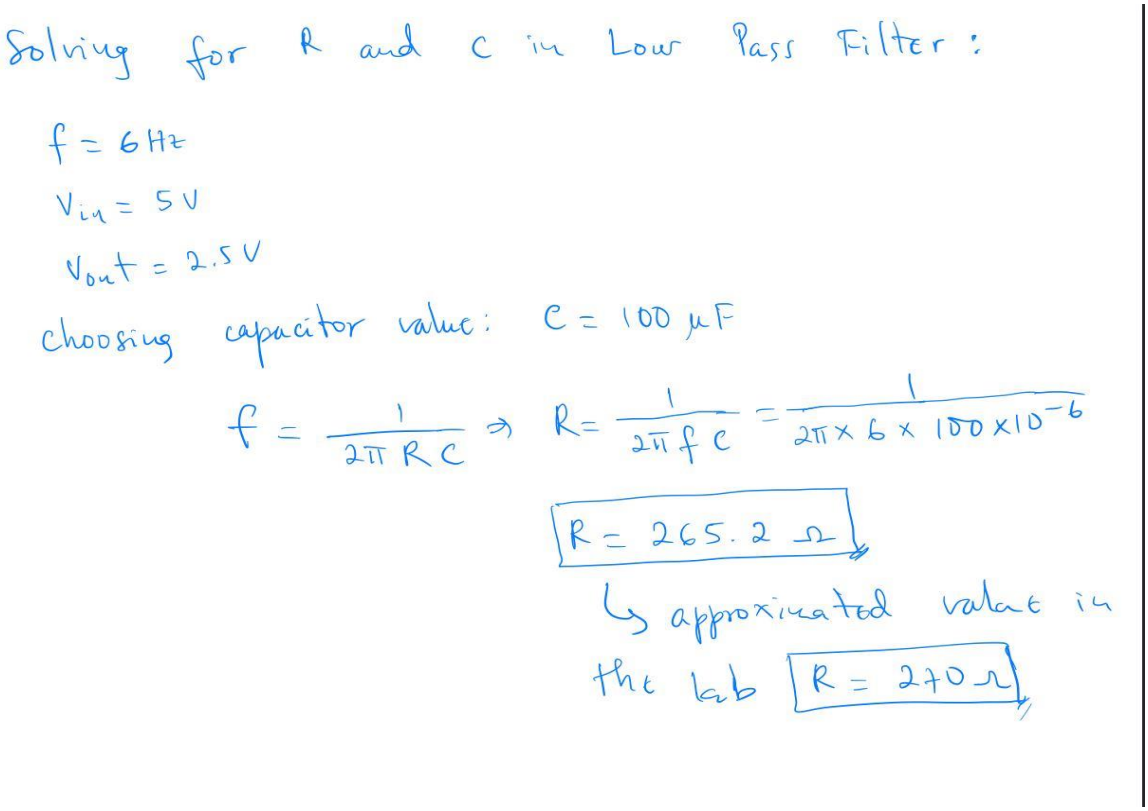

Figure 3: Low-pass filter equations

Software

For the design of the software, we first focused on declaring all the registers we would need for phase locked loop, clock selection, clock configuration, division, and the pin select. For the phase locked loop, we ended up using PLL0. The clock that was used was the internal R-C oscillator that runs at 4MHz. We then picked a value that was both divisible by 8 and 11, and was above 275 so we could solve for our M and K. After figuring out these values, we wrote the code to disconnect and disable the PLL initially, write the configuration and division values, wait for the frequency to lock, then enable and connect the PLL again.

Software Equations

$M = (CCLK * N)/(2 * F_{IN})$

N = 1 Pick CCLK = 352MHz F_{IN} = 4MHz

M = (352MHz)/(8MHz)

M = 44, when entering software, M should be minus 1 its value so **M = 43**

Final Clock = CCLK/K

11MHz = 352MHz/K

K = 32 , when entering software, K should be minus 1 its value so **K = 31**

3. Oscilloscope Prints

Software

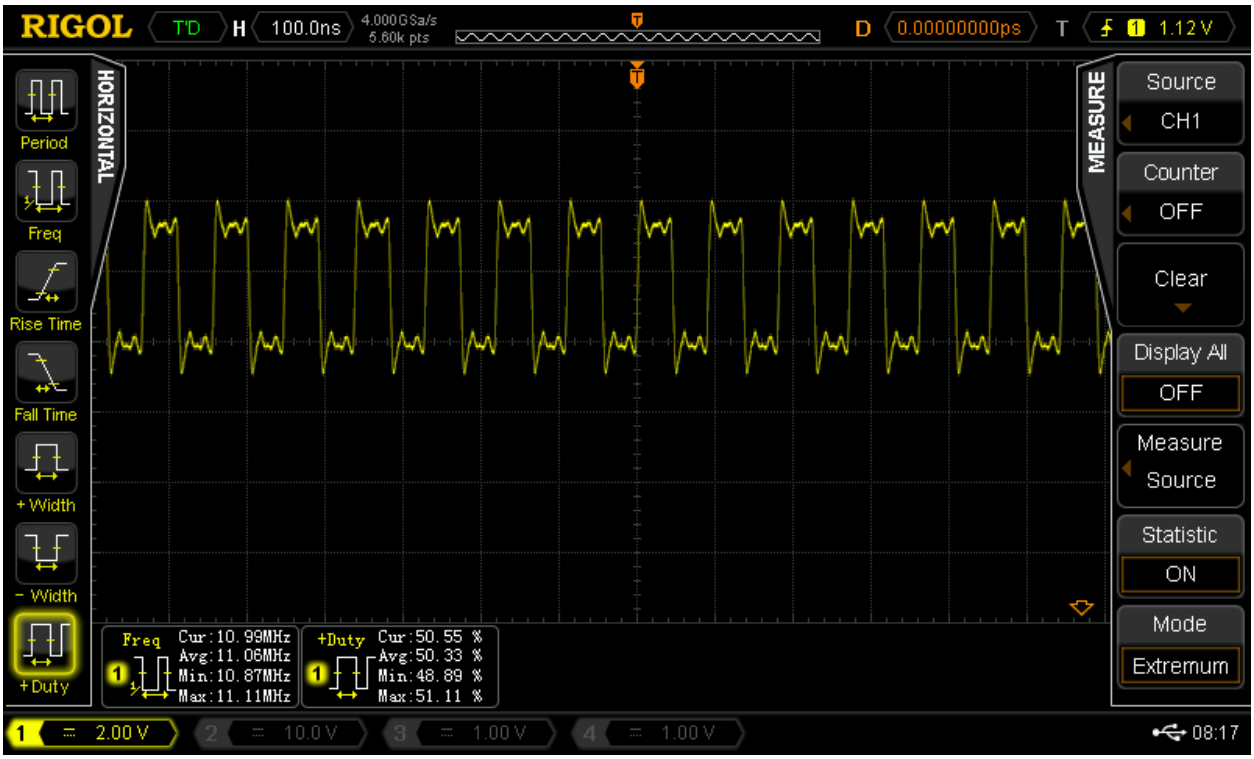

Figure 4: Software 11MHz oscilloscope reading

Hardware

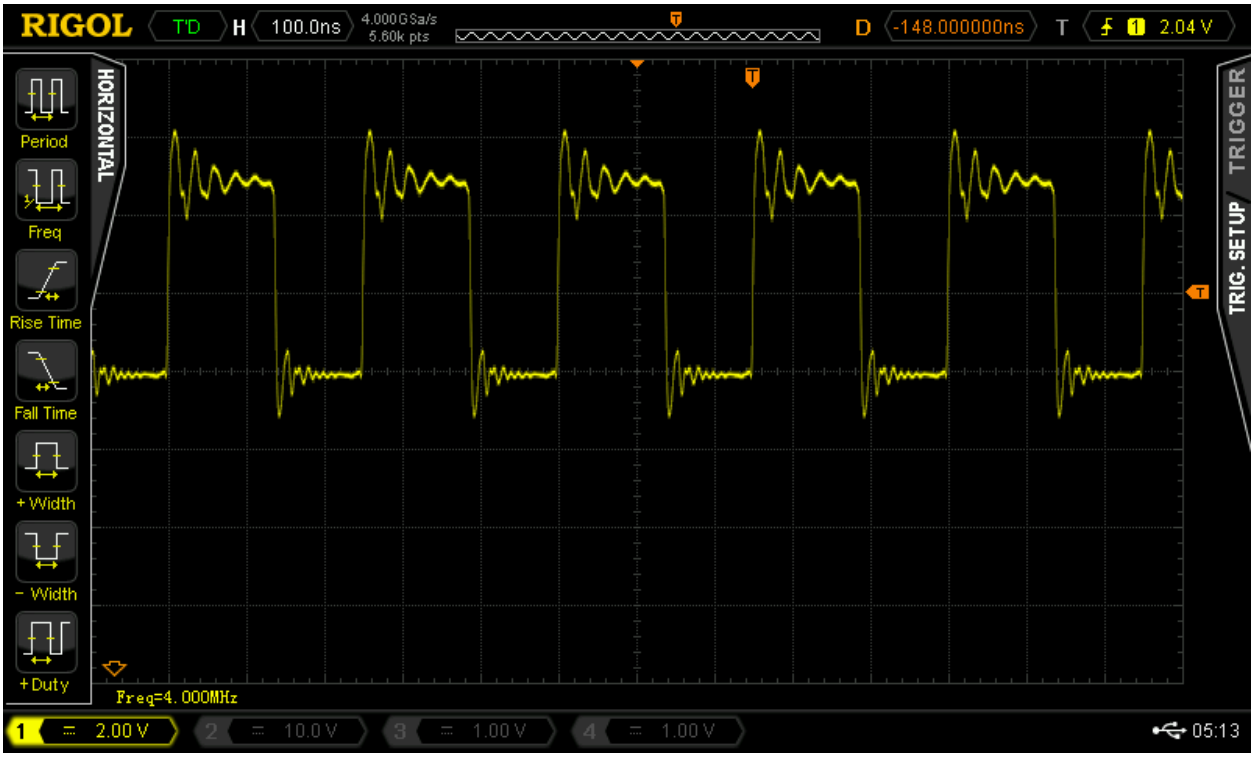

Figure 5: Hardware 4MHz oscilloscope reading

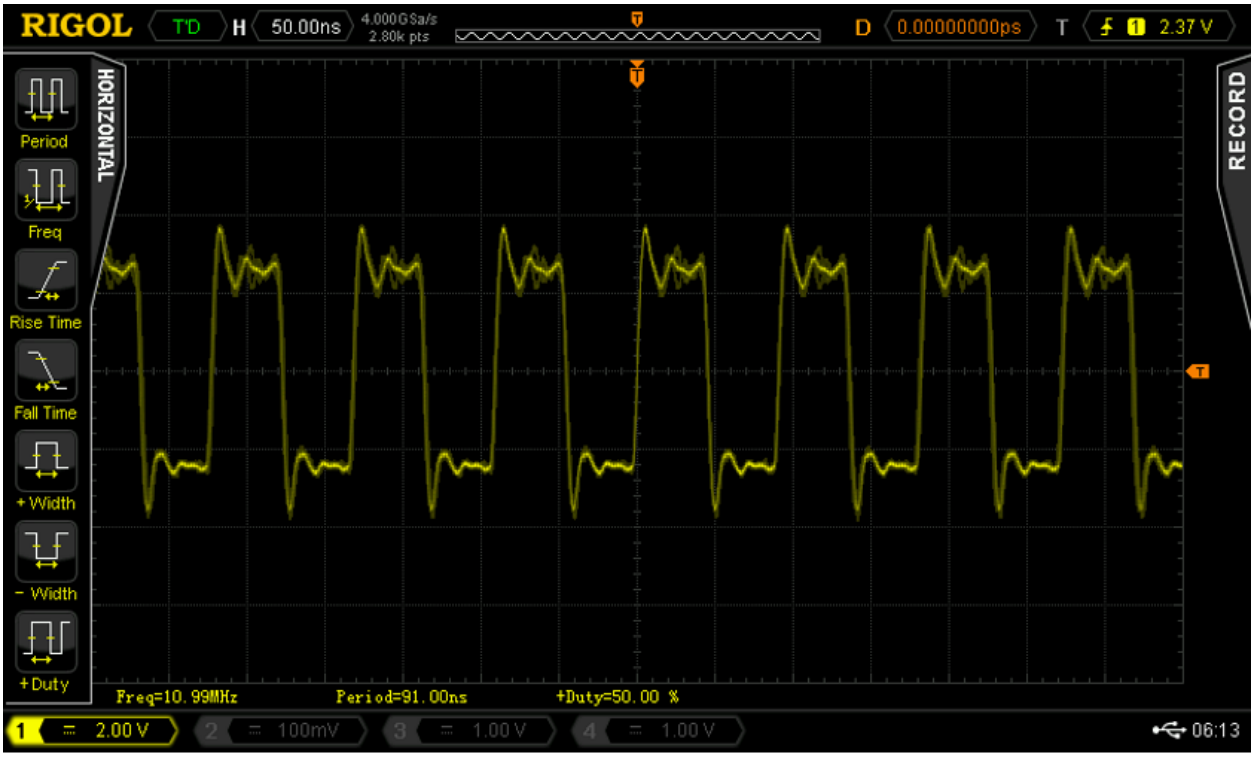

Figure 6: Hardware 11MHz oscilloscope reading

4. Details of the Solution

Complete Schematic

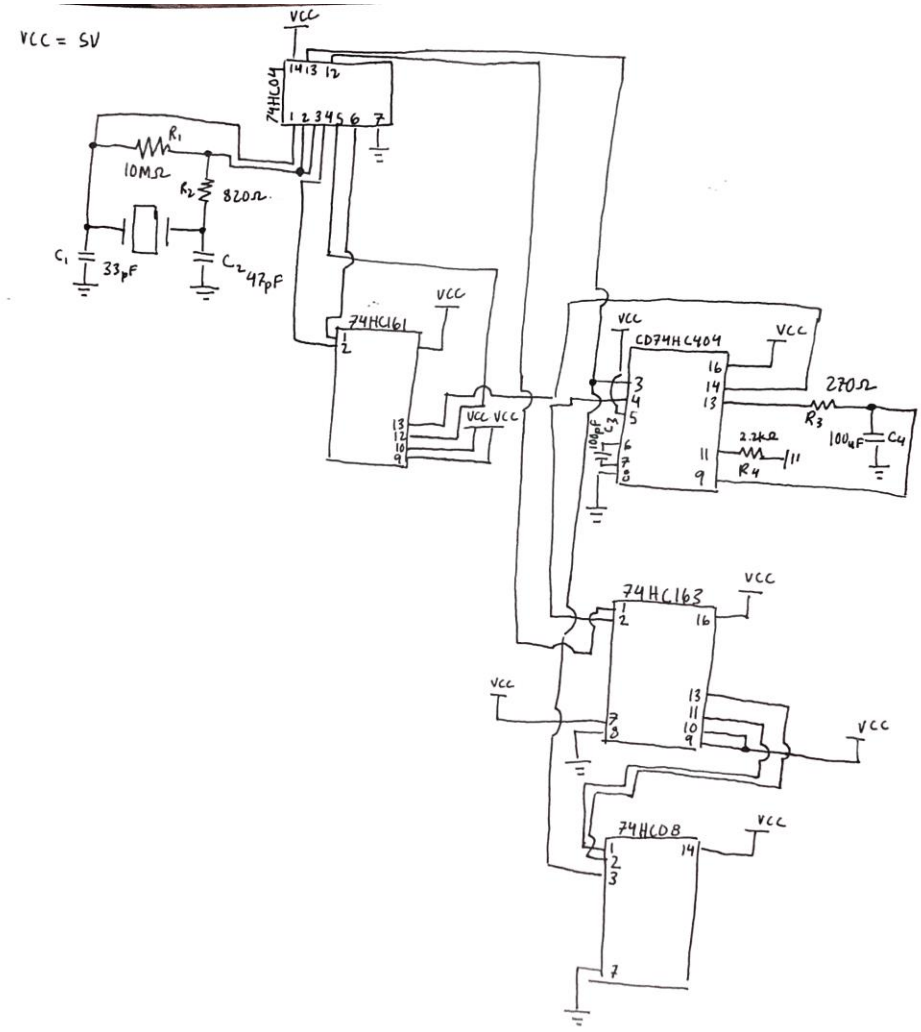

Figure 7: Schematic of hardware design

Source Code

```
/*
===============================================================================
 Name        : HW3.c
 Author      : Derek Stahl and Amilton Pensamento
 Version     :
 Copyright   : $(copyright)
 Description : main definition
===============================================================================
*/


#ifdef __USE_CMSIS
#include "LPC17xx.h"
#endif

#include <cr_section_macros.h>

// TODO: insert other include files here

// TODO: insert other definitions and declarations here

#define CLKSRCSEL (*(volatile unsigned int *)0x400FC10C)   //This register selects
the clock

//REGISTER DEFINITIONS OF PLL0
#define PLL0CON (*(volatile unsigned int *)0x400FC080)     //This register is the
control register
#define PLL0CFG (*(volatile unsigned int *)0x400FC084)     //This register is the
configuration register
#define PLL0STAT (*(volatile unsigned int *)0x400FC088)    //This register is the
status register
#define PLL0FEED (*(volatile unsigned int *)0x400FC08C)    //This register is the
Feed register

#define  CCLKCFG (*(volatile unsigned int *)0x400FC104) //This is the division
register
//clkcfg
#define CLKOUTCFG (*(volatile unsigned int *)0x400FC1C8) //clock configuration of the
clock output pin

//PIN SELECT 3 for the CLKOUT PIN
#define PINSEL3 (*(volatile unsigned int *) 0x4002C00C)

int main(void) {



	//Enables Port 1 bit 27 as the CLKOUT Pin
	PINSEL3 = (PINSEL3 | (1<<22)) & ~(1<<23);

	CLKOUTCFG = 0;      //selects the CPU clock as the output pin
	CLKOUTCFG |= (1<<8); //enable clk
```

```
    // 1. Disconnect PLL0 with one feed sequence if PLL0 is already connected.

       PLL0CON &= ~(1<<1);
       PLL0FEED = 0xAA; PLL0FEED = 0x55;

       //2. Disable PLL0 with one feed sequence.

       PLL0CON &= ~(1<<0);
       PLL0FEED = 0xAA; PLL0FEED = 0x55;
       //3.
       CCLKCFG = 0;

       //4. select 4MHz register
       CLKSRCSEL = 0;
       PLL0FEED = 0xAA; PLL0FEED = 0x55;

       //5. Write to the PLL0CFG and make it effective with one feed sequence. The
PLL0CFG can only be updated when PLL0 is disabled.
       PLL0CFG |= 43; // (0x2C<<0);     //multiplies by 44
       PLL0FEED = 0xAA; PLL0FEED = 0x55;



       //6. Enable PLL0 with one feed sequence.
       PLL0CON |= (1<<0);
       PLL0FEED = 0xAA; PLL0FEED = 0x55;




       //8. Wait for PLL0 to achieve lock   if (((FIO1PIN>>31) & 1) == 1)
       while(((PLL0STAT>>5) & 1) != 1){
              //do nothing

       }
       //7.  Change the CPU Clock Divider setting for the operation with PLL0. It is
critical to do this before connecting PLL0.
       CCLKCFG = 31; //this divides 2m/n by 32


    // 9. Connect the PLL0
       PLL0CON |= (1<<1);
       PLL0FEED = 0xAA; PLL0FEED = 0x55;







    return 0 ;
}
```

ECE 4273

Lab Demonstration Sign-off

Assignment Number	#3
Team Members Demoing	Derek Stahl
	Amilton Pensamento
Date	9/28/21 , 10/1/21
Time	
Witnessed by	Gopi

Were all objectives completed?

✓ Yes

_____ No

If "No", describe which objectives were completed or not completed (whichever is easiest):

Hardware Soln Done! @ Gopi 10/09/28