NAME:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ECE-C304 Lab – Assignment # 3

(3A,3C and 3D due 6th Week)

The lab is based on the handout supplied

*Note: Always use the part no CY8C29466-24PXI*

Lab 3A-Global Output

3A-1 What is the frequency of each variable clock?

VC1\_**24 MHz**\_ VC2\_**1.5 MHz**\_ VC3**\_48\_MHz\_**

3A-2 Having set all these parameters, what is the output frequency of the PWM signal?

**5.9 KHz**

3A-3 How would changing the compare type from **LessThan** to **LessThanOrEqual** change the

waveform?

**It slightly increases the duty cycle. With LessThan Duty cycle can’t be 100% but can be 0%. With LessThanOrEqual the duty cycle can be 0% but can’t be 100%.**

3A-4 Is LED1 on?

**Yes**

Change pulse width to 200. Is the LED brighter or dimmer now? Why?

**The LED is brighter because the average voltage has grown larger due to the longer length of pulses.**

Change the pulse width back to 32 for 3A-5 to 3A-8

3A-5 Are the readings consistent with your predictions?

Predicted Frequency = **6 kHz** Actual frequency = **6 kHz**

Predicted Pulse Width = **47 kHz** Actual Pulse Width = **47 kHz**

3A-5

Find out the registers and bits which determine VC1, VC2,VC3 clock divisors as well as those

which determine the period and pulse width of the PWM waveform. Include code in main.c

to set these values without depending on the GUI and demonstrate working. **(submit code**)

#include <m8c.h> // part specific constants and macros

#include "PSoCAPI.h" // PSoC API definitions for all User Modules

#include "DBB00.h"

void main(void)

{

OSC\_CR1 = 0x0F; // Sets VC1 & VC2

OSC\_CR3 = 0x02; // Sets VC3 divider

OSC\_CR4 &= ~0x03;

OSC\_CR4 |= 0x03; // Sets VC3 Source

DBB00\_WritePeriod(0xFF);

DBB00\_WritePulseWidth(0x20);

// M8C\_EnableGInt ; // Uncomment this line to enable Global Interrupts

// Insert your main routine code here.

DBB00\_Start();

while(1);

}

3A-6 How should these two waveforms differ?

**The waveforms should differ in the length of the pulses.**

3A-7 Does your observation confirm your prediction?

**Yes**

3A-8 Is this difference apparent when viewing LEDs?

**Yes, the P0[0] connected LED is brighter.**

Initials\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lab 3C – Global Output

3C-1 What is the frequency of VC3?

**46.9 kHz**

3C-2 What is the frequency of this PWM output?

**0.184 kHz**

3C-3 What is the frequency of this PWM output?

**0.185 kHz**

3C-4 At what frequency is this LED blinking? (This is not possible on the scope. Count how many times the light blinks in 10 secs on your watch and divide by 10 to get the frequency)

**1.5 Hz**

3C-5 Use the oscilloscope to view the output. Explain what is happening.

**The pulse width is alternating between the PWM1 and PWM2 pulse widths.**

3C-6 How must you change the clock so that it blinks at 1 Hz rate?

**The clock needs to be 16 MHz to get it to blink at 1Hz.**

3C-7 Do so by changing only the VC1, VC2 ,VC 3 parameters

**VC1 = 3, VC2 = Anything, VC3 = 256.**

Initials\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lab 3D – Modifying PWM Pulse Width under Software Control

3D-1 Describe how this algorithm works. Replace it with the algorithm explained in class

#include <m8c.h> // part specific constants and macros

#include "PSoCAPI.h" // PSoC API definitions for all User Modules

#include "PWM8\_1.h"

#include "LCD\_1.h"

unsigned char count = 0;

unsigned char pulseWidth = 0;

void main(void)

{

// M8C\_EnableGInt ; // Uncomment this line to enable Global Interrupts

// Insert your main routine code here.

PWM8\_1\_Start();

LCD\_1\_Start();

LCD\_1\_Init();

INT\_MSK0 |= 0x40;

M8C\_EnableGInt;

count = 0;

while(1){

M8C\_Sleep;

pulseWidth = PWM8\_1\_bReadPulseWidth();

LCD\_1\_Position(0,4);

//s = pulseWidth;

//LCD\_1\_PrData(pulseWidth);

LCD\_1\_Position(1,4);

LCD\_1\_PrHexByte(pulseWidth);

if(PRT1DR == 0x08){

count++;

}

else{

count = 0;

}

if(count == 3){

pulseWidth++;

PWM8\_1\_WritePulseWidth(pulseWidth);

count = 0;

}

}

}

Initials\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3D-2

LCD displaying pulse width in hex, with the modified algorithm

Initials\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Time\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_