Kiana Peterson and Nattapon Oonlamom
CSE 474: Intro to Embedded Systems
December 7th, 2023
Lab 5: Create Your Own Lab

**Procedure**

The lab required us to implement a new functionality using TM4C1294NCPDT. Our objective was to use the Pulse-Width Modulator (PWM) to dim an on-board LED in a loop as suggested by the lab document.

**Figure 23-1. PWM Module Diagram**

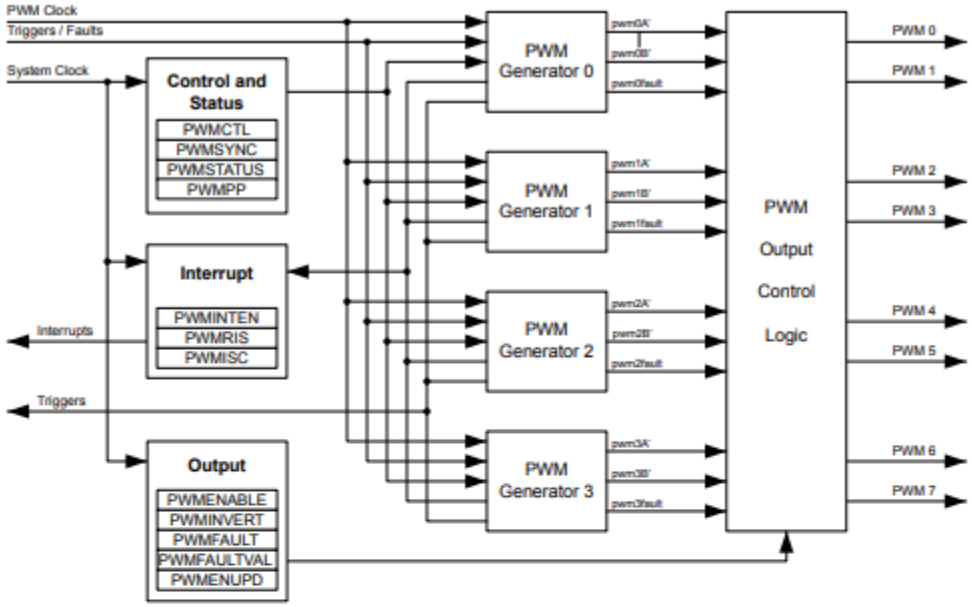


**Figure 23-2. PWM Generator Block Diagram**

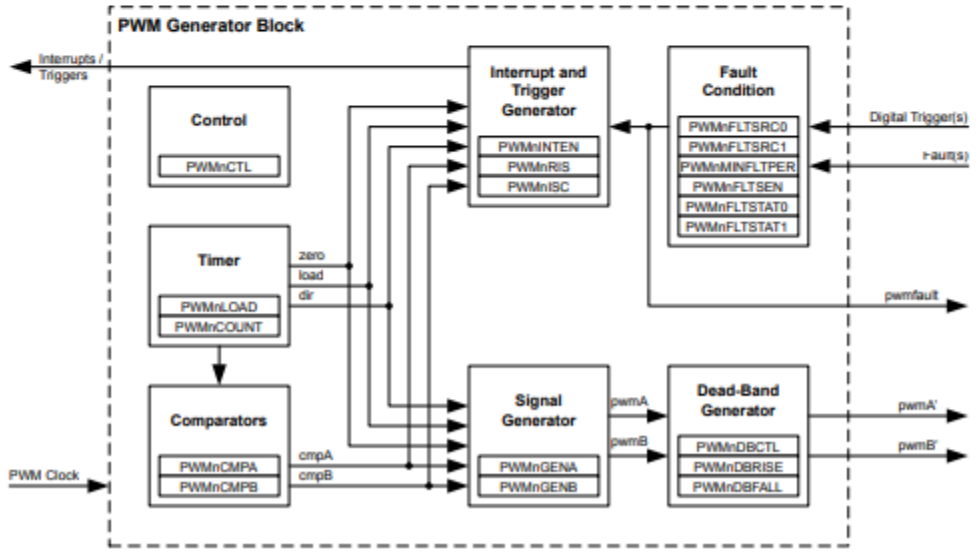


Figure 1: PWM Diagrams from Datasheet

Initially, we had several ideas about how we wanted to implement the PWM, varying from a dimming traffic light controller to creating moving LEDs that dim. Eventually, we decided to focus on dimming one on-board LED.

Our approach was to first initialize the on-board LED4 similar to Lab 1 Task 1a. After having done so, we created a PWM initialization function, going through the steps noted in the datasheet. Initially, we replicated the steps, but with a 16MHz system clock. This meant a 25% duty cycle on M1PWM2 and 75% on M1PWM3 with a 25kHz frequency PWM. We also chose PWM generator 1 instead of 0 because we thought having the PWM and the LED on the same port would cause an issue. In the end, we realized the way to connect the PWM to the LED was through having the same port, in which we chose M0PWM0 at port PF0. We also realized we did not need two PWMs, we could initialize just one of them and change the duty cycle as needed to dim or brighten the LED. The block diagram for the PWM can be found in Figure 1. We calculated the ticks in a period using the clock frequency (16MHz) divided by the PWM frequency (10kHz) to get 16000 ticks per second. We then loaded this value minus 1 into PWMLOAD0 and used the PWMCMPA0 to generate the desired duty cycle (we chose 0). We used the for loop to modify the duty cycles smoothly from 100% to 0% and back. This process is then looped infinitely in main. This meets the requirements as we implemented the PWM, a new functionality.

| Pin Name | Pin Number | Pin Mux / Pin Assignment | Pin Type | Buffer Type | Description |
|---|---|---|---|---|---|
| M0FAULT0 | 46 | PF4 (6) | I | TTL | Motion Control Module 0 PWM Fault 0. |
| M0FAULT1 | 61 | PK6 (6) | I | TTL | Motion Control Module 0 PWM Fault 1. |
| M0FAULT2 | 60 | PK7 (6) | I | TTL | Motion Control Module 0 PWM Fault 2. |
| M0FAULT3 | 81 | PL0 (6) | I | TTL | Motion Control Module 0 PWM Fault 3. |
| M0PWM0 | 42 | PF0 (6) | O | TTL | Motion Control Module 0 PWM 0. This signal is controlled by Module 0 PWM Generator 0. |

Figure 2: PWM Signal Pins from Datasheet

**Result**

Our on-board LED4 lit up gradually, then dimmed until off. Then, the process began again in an infinite loop. As a result, the code successfully functions as expected, passing all the requirements for this lab!