Understanding Pulse-Width Modulation

EE332/493 Embedded Systems Hardware/Software Spring 2021

Lab 4

Alec Bakholdin

Submitted November 10, 2021

Github link: <https://github.com/embedded-systems-2-fall-2021-labs/lab-4-Alec-Bakholdin-Rutgers>

Purpose/Objective - To get us to dig deep into coding with IP. Getting any sort of output was difficult and rewarding.

Theory of Operation – In vivado, I added an AXI Timer, attached its interrupt output to the Zynq Processor, and made the pwm0 port external. Next, I created a constraint file and mapped pwm0\_0 to one of the LEDs (I chose the third LED at pin M14). In Vitis, I copied the modified driver files for xtmrctr.h and xtrmctr.c from Xilinx’s PWM example, and used the example code as a guide to write my own.

A few things were off in the example:

1. The PMW period was far too large, so the human eye could see that the signal was not continuous.
2. The example would end after just 4 different duty cycles.

The solutions to these issues were simple. For the first one, I just changed around the declared constants, reducing PWM\_PERIOD and increasing CLOCKS\_PER\_DUTYCYCLE. I also changed the names of a few of the constants to match what I was doing. For HighTime, instead of dividing by an incrementing divisor, I made it a linear function of time, wherein HighTime = step/steps\_per\_dutycycle\*pwm\_period. This can also be phrased as HighTime/PWM\_PERIOD = DutyCycle, and since step would increment each iteration of the outer while loop, the duty cycle would steadily creep up. To fix the second issue, all I had to do was reset the counter (Div in the example’s case, Step in my case) to 1 every time we reached the maximum step count (10 in my case). The idea was that DutyCycle would approach 100, and once it hit 100, it would reset back to a very low number (1/10).

Please note I added the virtual inputs for vlab, so the images are different from what they would be if I had done it physically

Diagram, schematic

Description automatically generated

Graphical user interface, application

Description automatically generated

Chart, bar chart

Description automatically generated

Schematic

Description automatically generated

Diagram, schematic

Description automatically generated