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

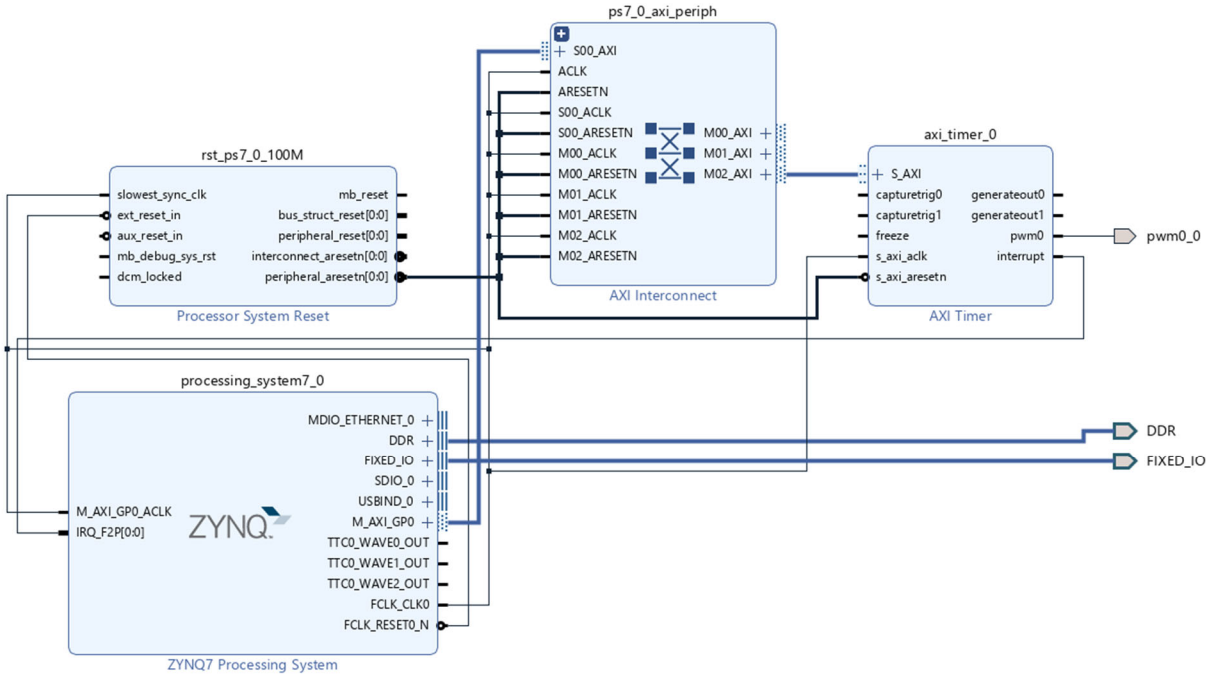
Purpose/Objective - To get us to dig deep into coding with IP. Getting any sort of output was difficult and rewarding.

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.

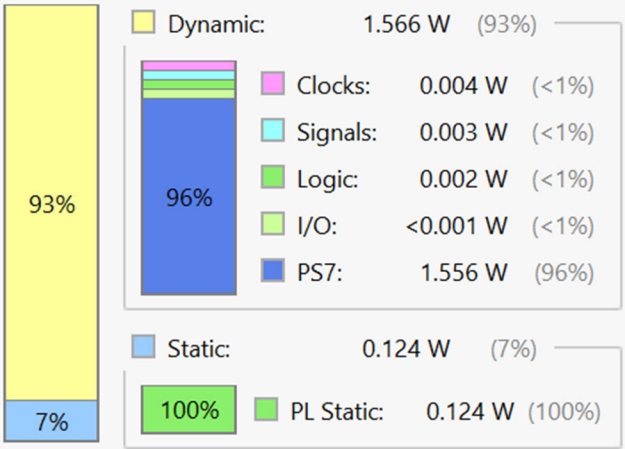
A few things were off in the example:

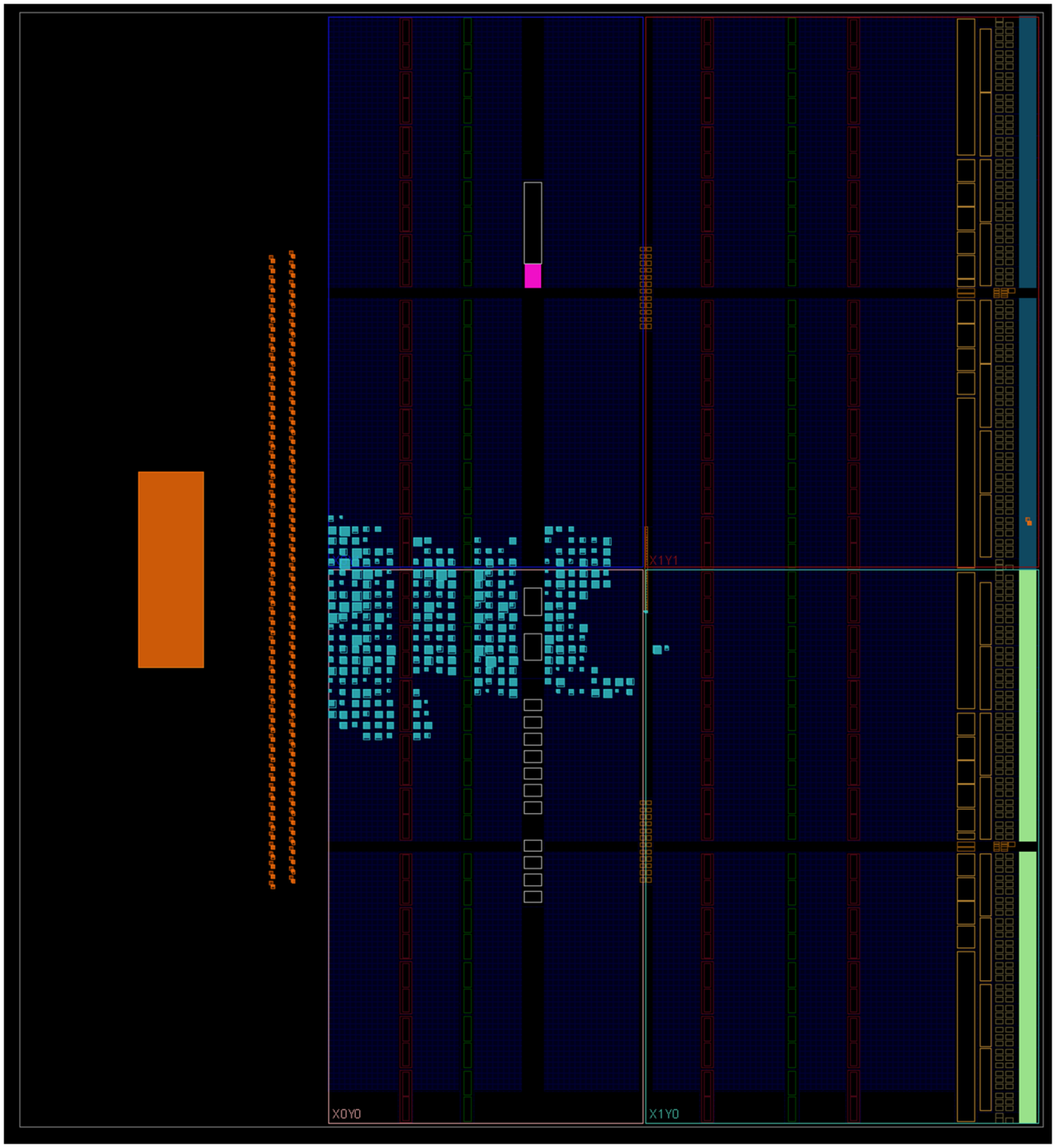
1. The PMW period was far too large, so the human eye could see that the signal was not continuous.
2. The example would end after just 4 different duty cycles.

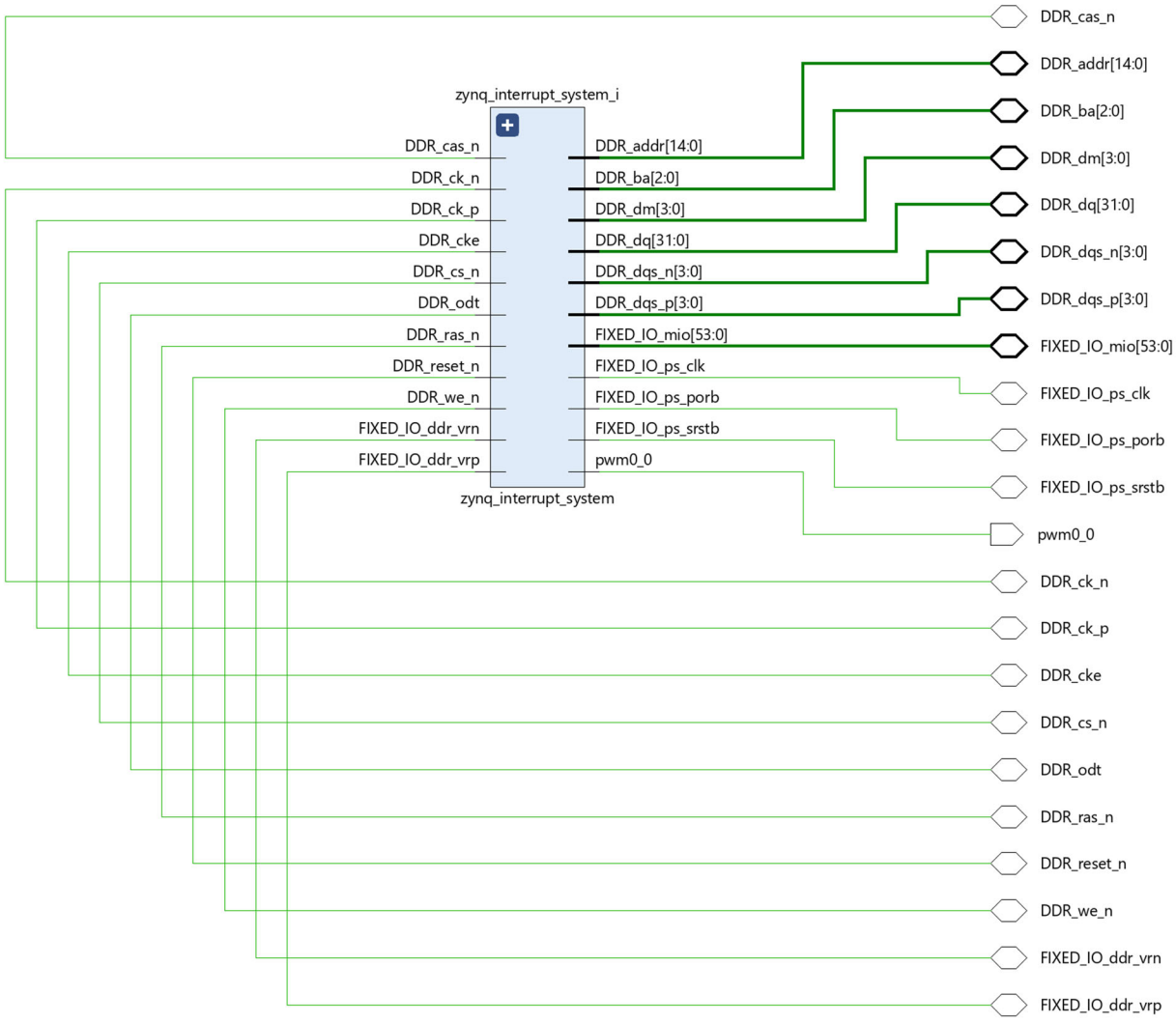
The solutions to these issues were simple. For the first one, I just changed around the declared constants, reducing PWM_PERIOD and increasing CLOCKS_PER_DUTYCYCLE. I also changed the names of a few of the constants to match what I was doing. For HighTime, instead of dividing by an incrementing divisor, 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).




| Resource | Estimation | Available | Utilization % |
|---|---|---|---|
| IO | 1 | 100 | 1.00 |




X1Y1
X0Y0
X1Y0


zynq_interrupt_system_i
DDR_cas_n
DDR_ck_n
DDR_ck_p
DDR_cke
DDR_cs_n
DDR_odt
DDR_ras_n
DDR_reset_n
DDR_we_n
FIXED_IO_ddr_vrn
FIXED_IO_ddr_vrp
DDR_addr[14:0]
DDR_ba[2:0]
DDR_dm[3:0]
DDR_dq[31:0]
DDR_dqs_n[3:0]
DDR_dqs_p[3:0]
FIXED_IO_mio[53:0]
FIXED_IO_ps_clk
FIXED_IO_ps_porb
FIXED_IO_ps_srstb
pwm0_0
zynq_interrupt_system
DDR_cas_n
DDR_addr[14:0]
DDR_ba[2:0]
DDR_dm[3:0]
DDR_dq[31:0]
DDR_dqs_n[3:0]
DDR_dqs_p[3:0]
FIXED_IO_mio[53:0]
FIXED_IO_ps_clk
FIXED_IO_ps_porb
FIXED_IO_ps_srstb
pwm0_0
DDR_ck_n
DDR_ck_p
DDR_cke
DDR_cs_n
DDR_odt
DDR_ras_n
DDR_reset_n
DDR_we_n
FIXED_IO_ddr_vrn
FIXED_IO_ddr_vrp