California Polytechnic State University Pomona

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

Digital Circuit Design Verilog

ECE 3300L

Report #8

Prepared by

Heba Hafez 017353323

Sean Wygant 017376658

Group Y

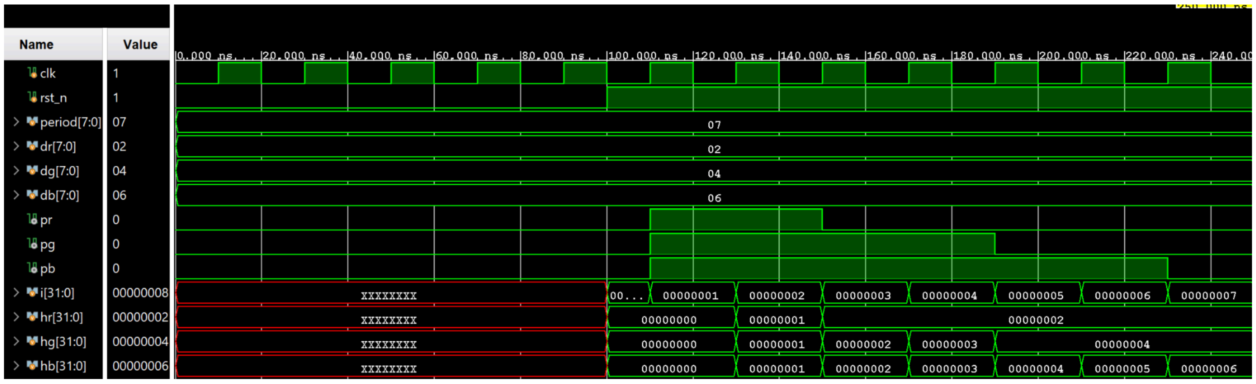
Mohamed Aly

August 12, 2025

Objective: In Lab 8 students will design and implement a hardware-controlled RGB LED brightness and color controller on the Nexys A7 FPGA using Pulse Width Modulation (PWM). The system uses a single push button to sequentially load four parameters, PWM period (resolution), red duty cycle, green duty cycle, and blue duty cycle. The parameters are put into registers, enabling smooth color blending and brightness control.

Code and Explanation (4 slot loading and RES+1):
As there is only one load button but we have 4 different cycles to run (PWM resolution, red duty cycle, green duty cycle, and blue duty cycle) we have to implement 4 slot loading. What this means is that we use a finite state machine (FSM) to cycle through the slots. When you press the button it will move to the next slot. For instance slot 0 is load PWM resolution, slot 1 is load red duty cycle, slot 2 is load green duty cycle, and slot 3 is load blue duty cycle. After slot 3, when the button is pressed it wraps back to slot 0. With the LED's you can visually see this as the active slot will appear.

For RES+1 in the pwm_core module, eff_period or the effective period is calculated using RES + 1. This means RES from 0 to 7 is added to one to match the period length. As RES can be anywhere from 0 to 8 the +1 is there to ensure the clock cycle is accurate because without it the output duty cycle and frequency may be off.



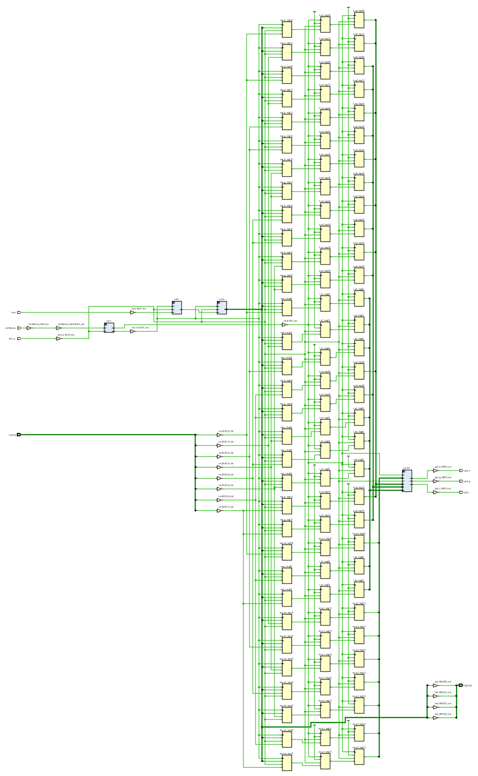
```
Time resolution is 1 ps
R=2/8 G=4/8 B=6/8 (expected: ~2,4,6)
$finish called at time : 250 ns : File "C:/Users/Sean/Documents/Summer 25 CPP/ECE3300L/Lab 8/Lab 8.srcs/sim_1/new/pwm_core_tb.v" Line 41
```

Vivado (LUTs, FFs, Power):

```
1. Slice Logic
--------------

+-------------------------+------+-------+------------+-----------+-------+
|        Site Type        | Used | Fixed | Prohibited | Available | Util% |
+-------------------------+------+-------+------------+-----------+-------+
| Slice LUTs*             |  111 |     0 |          0 |     63400 |  0.18 |
|   LUT as Logic          |  111 |     0 |          0 |     63400 |  0.18 |
|   LUT as Memory         |    0 |     0 |          0 |     19000 |  0.00 |
| Slice Registers         |  152 |     0 |          0 |    126800 |  0.12 |
|   Register as Flip Flop |  152 |     0 |          0 |    126800 |  0.12 |
|   Register as Latch     |    0 |     0 |          0 |    126800 |  0.00 |
| F7 Muxes                |    0 |     0 |          0 |     31700 |  0.00 |
| F8 Muxes                |    0 |     0 |          0 |     15850 |  0.00 |
+-------------------------+------+-------+------------+-----------+-------+
* Warning! The Final LUT count, after physical optimizations and full implementation, is typically lower. Run opt_design after synthesis, if not already completed,
Warning! LUT value is adjusted to account for LUT combining.
Warning! For any ECO changes, please run place_design if there are unplaced instances
```

```
1.1 Summary of Registers by Type
--------------------------------

+-------+--------------+-------------+--------------+
| Total | Clock Enable | Synchronous | Asynchronous |
+-------+--------------+-------------+--------------+
| 0     |            _ |           - |            - |
| 0     |            _ |           - |          Set |
| 0     |            _ |           - |        Reset |
| 0     |            _ |         Set |            - |
| 0     |            _ |       Reset |            - |
| 0     |          Yes |           - |            - |
| 6     |          Yes |           - |          Set |
| 146   |          Yes |           - |        Reset |
| 0     |          Yes |         Set |            - |
| 0     |          Yes |       Reset |            - |
+-------+--------------+-------------+--------------+
```

```
1. Summary
----------

+--------------------------+--------------+
| Total On-Chip Power (W)  | 0.086        |
| Design Power Budget (W)  | Unspecified* |
| Power Budget Margin (W)  | NA           |
| Dynamic (W)              | 0.003        |
| Device Static (W)        | 0.084        |
| Effective TJA (C/W)      | 4.6          |
| Max Ambient (C)          | 84.6         |
| Junction Temperature (C) | 25.4         |
| Confidence Level         | Low          |
| Setting File             | ---          |
| Simulation Activity File | ---          |
| Design Nets Matched      | NA           |
+--------------------------+--------------+
* Specify Design Power Budget using, set_operating_conditions -design_power_budget <value in Watts>
```



Video Link:
https://youtu.be/wIoApTur-44

Reflections:
In this lab, a fully functional RGB LED PWM controller was implemented, simulated, and tested on the Nexys A7 FPGA. The given design successfully cycled through four load slots, adjusted PWM resolution, and varied LED brightness for each color channel, producing accurate and flicker-free output. Simulation results matched the expected duty cycle behavior, and the hardware test confirmed correct timing, color mixing, and response to button presses. This exercise revisited the concepts of digital design modularity, clock domain crossing, debouncing techniques, and the practical use of PWM in controlling analog-like outputs with digital hardware.

Partner Contributions: The work on this lab was 50/50. While this lab required less lab work than the others we worked on the lab together in person, with Sean running the lab on his computer. The lab report was also split up.