California Polytechnic State University Pomona

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

Dgtl Circuit Dsgn Verilog Lab

ECE 3300L Section E01

Report # 6

Prepared by

Arvin Ghaloosian (017153422)
Vittorio Huizar (016826784)
Group H

Presented to

Mohamed Aly (Mohamed El-Hadedy)

Due - July 29, 2025

DESIGN OVERVIEW:

For Lab 6, we built a more advanced digital system combining two BCD up/down counters, a basic 4-bit ALU, and a control-display mechanism using the 7-segment display. We used the Nexys A7's onboard 100 MHz clock and divided it down using a clock divider that selects from 32 different speeds via SW[4:0]. The two BCD counters (units and tens) operate independently, each with its own direction switch (SW7 and SW8).

The ALU accepts these two BCD values and can either add or subtract them based on control switches SW6 and SW5. We then display the result on two digits of the 7-segment display, with a third digit showing the control nibble {SW8, SW7, SW6, SW5}. The raw BCD outputs are mirrored on LEDs for debug visibility.

MODULE HIERARCHY:

- top_lab6.v
    - clock_divider
    - bcd_counter (units)
    - bcd_counter (tens)
    - alu
    - control_decoder
    - seg7_scan

Top Module:

Explanation:
The top-level module coordinates everything and handles pin mapping from the XDC file.

- BTN0 (active-low) resets both BCD counters to 0.
- SW[4:0] selects the speed by choosing a bit from the 32-bit counter.

- SW7 and SW8 independently control the counting direction for the units and tens counters.
- SW6–SW5 select the ALU operation: 00 = add, 01 = subtract.
- The ALU result is shown on AN0 and AN1, while the control nibble is shown on AN2.
- LED[3:0] shows the units BCD value, LED[7:4] shows the tens BCD value.

---

clock_divider Module:

Explanation:
This module implements a 32-bit free-running counter driven by the 100 MHz system clock. A 32×1 mux picks a specific bit from the counter (based on SW[4:0]) to serve as the clk_div output, controlling the rest of the system's timing. The lower the switch value, the slower the count rate.

---

bcd_counter Module (x2):

Explanation:
Each instance holds a BCD digit (0–9).

- On clk_div rising edge:
    - If dir_bit is 1, the counter increments (rolls over from 9 to 0).
    - If dir_bit is 0, the counter decrements (rolls under from 0 to 9).
- Reset (BTN0 active-low) clears the count to 0.

---

alu Module:

Explanation:
This module receives two 4-bit BCD values (units and tens) and performs:

- 00: Addition (result = A + B, max 18)

- 01: Subtraction (result = A - B, wraps under)
- Other values: result = 0
  The output is an 8-bit result, split across two 7-segment digits.

---

control_decoder Module:

Explanation:
It groups SW8–SW5 into a 4-bit control nibble for easy display on AN2. This lets us visually verify switch positions.

---

seg7_scan Module:

Explanation:
This handles the 7-segment multiplexing. At ~1 kHz rate, it cycles between:

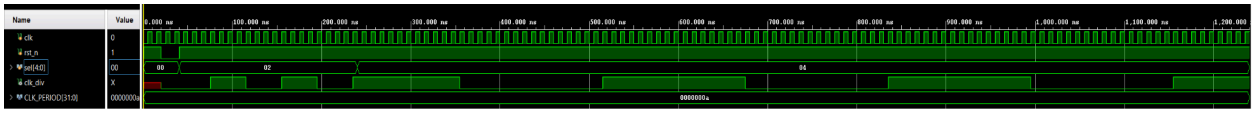
- AN0: result[3:0]
- AN1: result[7:4]
- AN2: control nibble

The module uses a small prescaler and lookup logic to convert 4-bit nibbles to 7-segment patterns.

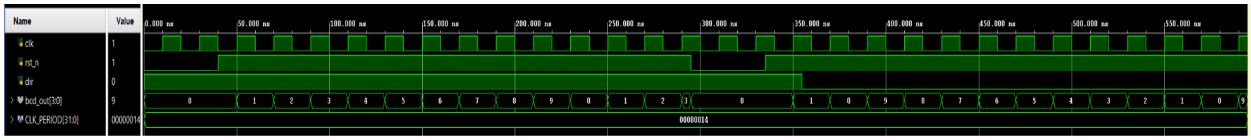
TEST BENCH:

Explanation:
Each module was tested in isolation using behavioral simulation:

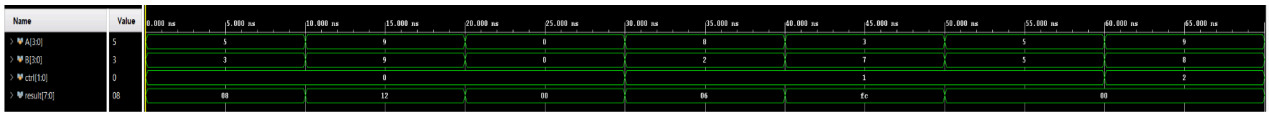
- clock_divider_tb.v: Verified different speed selections from SW[4:0].



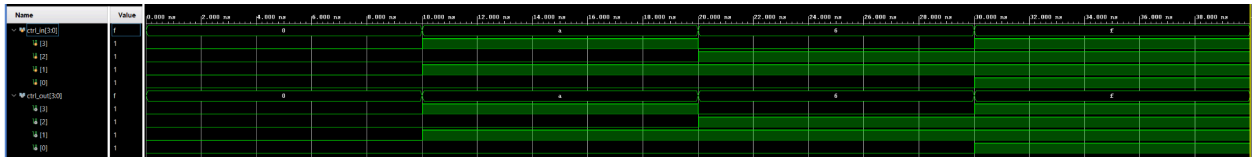
- bcd_counter_tb.v: Tested both increment and decrement behavior on direction toggle.



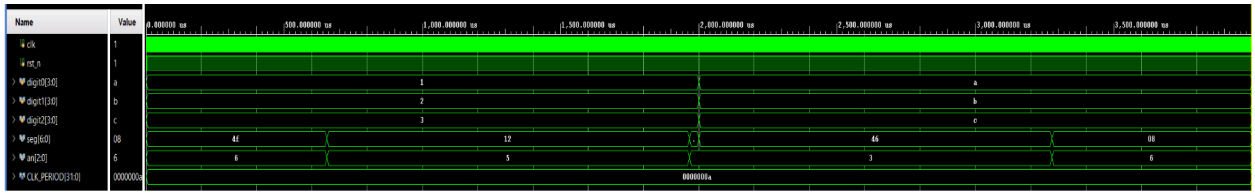
- alu_tb.v: Verified addition, subtraction, and edge cases (overflow and underflow).



- control_decoder_tb.v: Confirmed that {SW8, SW7, SW6, SW5} converted to the correct nibble.

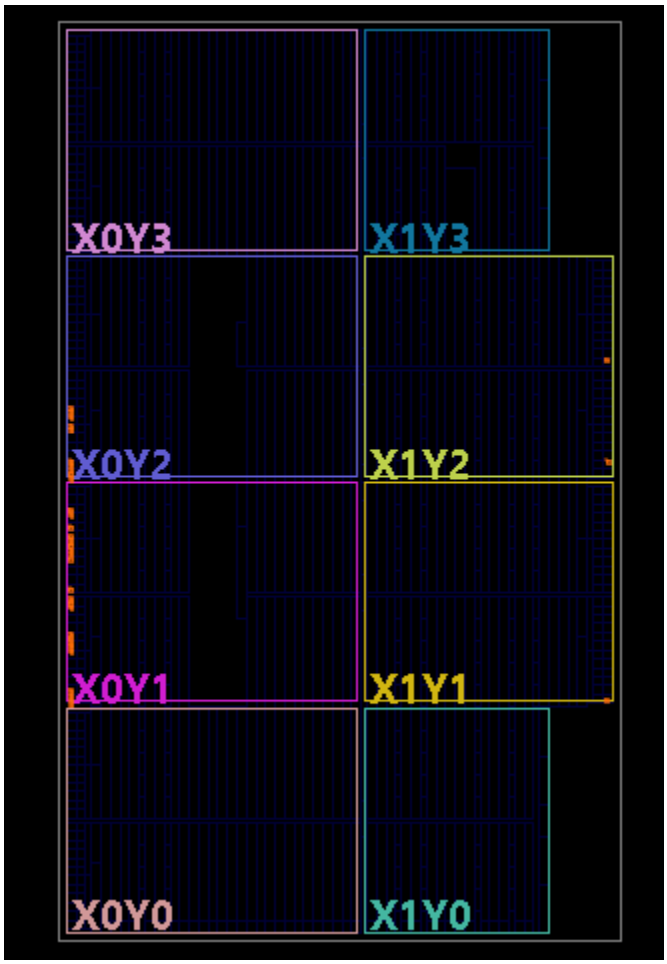


- seg7_scan_tb.v: Confirmed digit multiplexing at a stable ~1 kHz refresh rate.



Simulations confirmed correct functionality of each module. Waveforms matched expected timing.

SYNTHESIS SCHEMATIC:



RESOURCE UTILIZATION:

Name	Slice LUTs (63400)	Slice Registers (126800)	F7 Muxes (31700)	Bonded IOB (210)	BUFGCTRL (32)
N top_lab6	41	59	4	35	1

CONTRIBUTIONS:

Arvin Ghaloosian (50%)

- Implemented and verified bcd_counter and control_decoder
- Led on-board testing and logic verification
- Developed clock_divider and alu modules

Vittorio Huizar (50%)

- Ran testbenches and handled simulation debugging
- Integrated seg7_scan and verified switch mapping
- Cleaned up top-level module and constraint file

REFLECTION:

This lab brought all the prior concepts together into one cohesive design. It reinforced how to modularize a system and build from small verified blocks. We especially liked how the control nibble made verifying switch settings intuitive. One challenge was ensuring ALU wraparound behaved correctly, adding extra logic to handle underflow taught us good defensive design. Next time, we'd like to explore button debouncing and maybe support more ALU operations.

Link to demo video:

https://youtube.com/shorts/BBnnrli3Ipo