

EEC 180 Lab 5

Bradley Manzo

916953788

Objective and Introduction: The objective of this lab is to construct two dependent counters in such a manner as to alter the speed of the counter, as well as the behavior including counting up, down, to half of the max, or to run continuously.

Prelab:

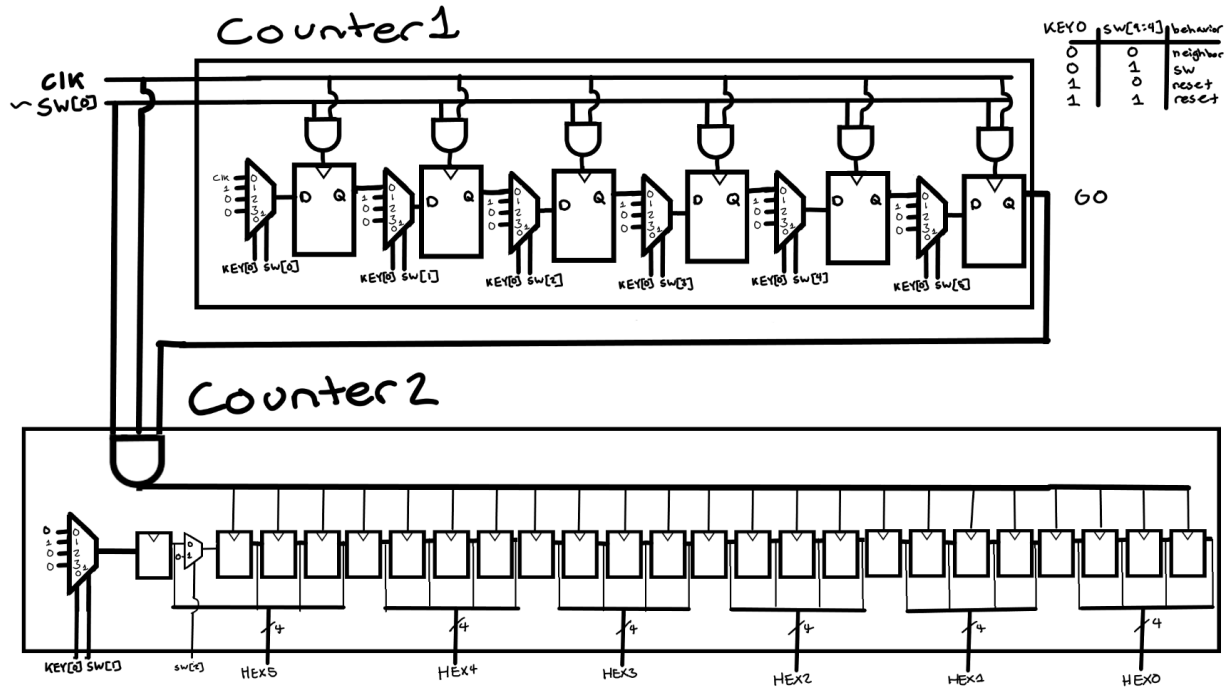


Figure 1: top.v Block Diagram

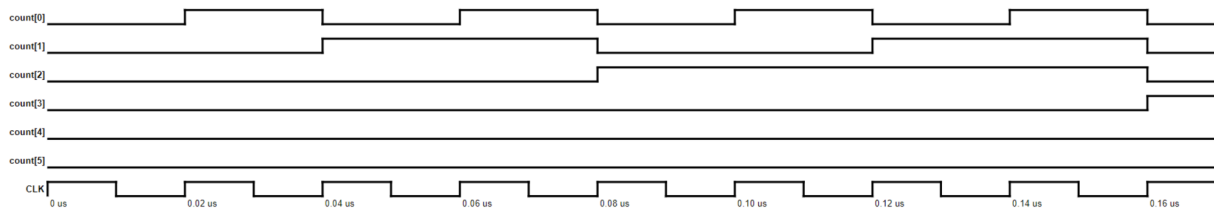


Figure 2: counter1.v Timing Diagram

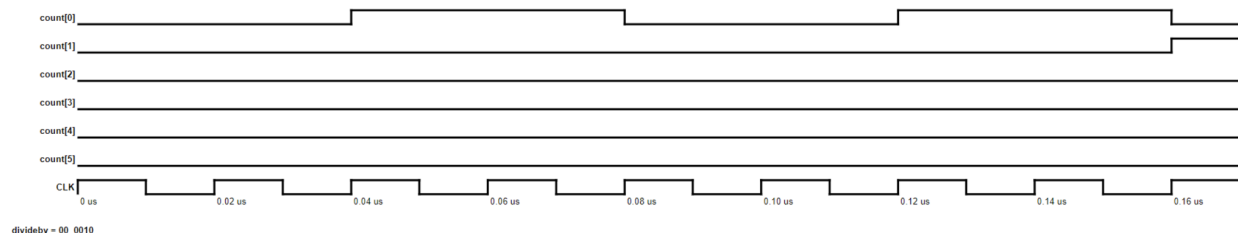


Figure 3: counter2.v Timing Diagram

- 1) one least significant digit increment: $50 \text{ MHz} = 0.02 \mu\text{s}$
- 2) a) divideby = 000001: $16,777,215 \times 0.02 \mu\text{s} = 0.3355$
- b) divideby = 110010: $(16,777,215 \times 0.02 \mu\text{s}) \times 50 = 16.7772$

Results:

```
# Case: Normal Operation to 6
# count: 0, go: 1
# count: 1, go: 0
# count: 2, go: 0
# count: 3, go: 0
# count: 4, go: 0
# count: 5, go: 0
# count: 0, go: 1
# count: 1, go: 0
# count: 2, go: 0
# count: 3, go: 0
# count: 4, go: 0
# count: 5, go: 0
# count: 0, go: 1
# count: 1, go: 0
# count: 2, go: 0
# count: 3, go: 0
# count: 4, go: 0
# count: 5, go: 0
# count: 0, go: 1
# count: 1, go: 0
# count: 2, go: 0

# Case: Normal Operation to 15
# (Reset to zero)
# count: 0, go: 1
# count: 1, go: 0
# count: 2, go: 0
# count: 3, go: 0
# count: 4, go: 0
# count: 5, go: 0
# count: 6, go: 0
# count: 7, go: 0
# count: 8, go: 0
# count: 9, go: 0
# count: 10, go: 0
# count: 11, go: 0
# count: 12, go: 0
# count: 13, go: 0
# count: 14, go: 0
# count: 0, go: 1
# count: 1, go: 0
# count: 2, go: 0
# count: 3, go: 0
# count: 4, go: 0
# count: 5, go: 0

# Case: Reset Mid-Operation
# (Reset to zero)
# count: 0, go: 1
# count: 1, go: 0
# count: 2, go: 0
# count: 3, go: 0
# count: 4, go: 0
# count: 5, go: 0
# count: 6, go: 0
# count: 7, go: 0
# count: 8, go: 0
# count: 9, go: 0
# count: 10, go: 0
# (Reset to zero)
# count: 0, go: 1
# count: 1, go: 0
# count: 2, go: 0
# count: 3, go: 0
# count: 4, go: 0
# count: 5, go: 0
# count: 6, go: 0
# count: 7, go: 0
# count: 8, go: 0
# count: 9, go: 0
# count: 10, go: 0

# Case: Divideby set to 0 Mid-Operation
# (Reset to zero)
# count: 0, go: 1
# count: 1, go: 0
# count: 2, go: 0
# count: 3, go: 0
# count: 4, go: 0
# count: 5, go: 0
# count: 6, go: 0
# count: 7, go: 0
# count: 8, go: 0
# count: 9, go: 0
# count: 10, go: 0
# (Reset to zero)
# count: 0, go: 1
# (Divideby set to 0)
# count: 0, go: 0
# count: 0, go: 0
# count: 0, go: 0
# count: 0, go: 0
# count: 0, go: 0
# count: 0, go: 0
# count: 0, go: 0
# count: 0, go: 0
# count: 0, go: 0
# count: 0, go: 0
# count: 0, go: 0
```

Figure 4: Unit tests for Counter 1

```

# Case: free up                                # Case: run up, halfmax = 6
# count:      0, LEDR:      0 # count:      0, LEDR:      0
# count:      1, LEDR:      0 # count:      1, LEDR:      0
# count:      2, LEDR:      0 # count:      2, LEDR:      0
# count:      3, LEDR:      0 # count:      3, LEDR:      0
# count:      4, LEDR:      0 # count:      4, LEDR:      0
# count:      5, LEDR:      0 # count:      5, LEDR:      0
# count:      6, LEDR:      0 # count:      6, LEDR:      0
# count:      7, LEDR:      0 # count:      6, LEDR:      0
# count:      8, LEDR:      0 # count:      6, LEDR:      0
# count:      9, LEDR:      0 # count:      6, LEDR:      0
# count:     10, LEDR:      0 # count:      6, LEDR:      0
# count:     11, LEDR:      0 # count:      6, LEDR:      0
# count:     12, LEDR:      0 # count:      6, LEDR:      0
# count:     13, LEDR:      0 # count:      6, LEDR:      0
# count:     14, LEDR:      0 # count:      6, LEDR:      0
# count:     15, LEDR:      0 # count:      6, LEDR:      0
# count:     16, LEDR:      0 # count:      6, LEDR:      0
# count:     17, LEDR:      0 # count:      6, LEDR:      0
# count:     18, LEDR:      0 # count:      6, LEDR:      0
# count:     19, LEDR:      0 # count:      6, LEDR:      0
# count:     20, LEDR:      0 # count:      6, LEDR:      0

# Case: free down                                # Case: run down, halfmax = 6 # Case: Reset Hold
# count:      0, LEDR:      0 # count:      0, LEDR:      0 # count:      0, LEDR:      0
# count:     19, LEDR:      0 # count:     19, LEDR:      0 # count:      1, LEDR:      0
# count:     18, LEDR:      0 # count:     18, LEDR:      0 # count:      2, LEDR:      0
# count:     17, LEDR:      0 # count:     17, LEDR:      0 # count:      3, LEDR:      0
# count:     16, LEDR:      0 # count:     16, LEDR:      0 # count:      4, LEDR:      0
# count:     15, LEDR:      0 # count:     15, LEDR:      0 # count:      5, LEDR:      0
# count:     14, LEDR:      0 # count:     14, LEDR:      0 # count:      6, LEDR:      0
# count:     13, LEDR:      0 # count:     13, LEDR:      0 # count:      7, LEDR:      0
# count:     12, LEDR:      0 # count:     12, LEDR:      0 # count:      8, LEDR:      0
# count:     11, LEDR:      0 # count:     11, LEDR:      0 # count:      9, LEDR:      0
# count:     10, LEDR:      0 # count:     10, LEDR:      0 # count:     10, LEDR:      0
# count:      9, LEDR:      0 # count:      9, LEDR:      0 # count:      0, LEDR:      0
# count:      8, LEDR:      0 # count:      8, LEDR:      0 # count:      1, LEDR:      0
# count:      7, LEDR:      0 # count:      7, LEDR:      0 # count:      2, LEDR:      0
# count:      6, LEDR:      0 # count:      6, LEDR:      0 # count:      3, LEDR:      0
# count:      5, LEDR:      0 # count:      6, LEDR:      0 # count:      4, LEDR:      0
# count:      4, LEDR:      0 # count:      6, LEDR:      0 # count:      5, LEDR:      0
# count:      3, LEDR:      0 # count:      6, LEDR:      0 # count:      6, LEDR:      0
# count:      2, LEDR:      0 # count:      6, LEDR:      0 # count:      7, LEDR:      0
# count:      1, LEDR:      0 # count:      6, LEDR:      0 # count:      8, LEDR:      0
# count:      0, LEDR:      0 # count:      6, LEDR:      0 # count:      9, LEDR:      0
# count:      0, LEDR:      0 # count:      6, LEDR:      0 # count:     10, LEDR:      0

```

Figure 5: Unit tests for Counter 2

Conclusion:

After testing my counters and verifying the behavior on the DE10-Lite Board, it is clear that counter1 divides the frequency and multiplies the time to count up or down. The operation of top.v should be clear from the unit tests for counter1.v and counter2.v.