

# Experiments for the Lab9500

## Experiment 200 - Counters -

**Purpose:** In this experiment we will investigate a special kind of state machine called a counter. For the most part, the counting or state sequence is unchanging, although counters may have provision for clearing them (resetting them to zero) or presetting them to some starting number. It is highly recommended that you read the chapter in DDR on flip-flops and the one on state machines. Counters are designed to “divide by” an integer. Linear counters generate a straight linear count, which use all possible states of an n-bit counter. The number of states is  $2^n$ , where n is the number of counter bits. When the sequence resets prematurely before the maximum count, a divide-by-m counter is achieved where m is the number of counter states. We call such a counter a modulo m counter, or just mod m for short.

One very important objective of this experiment is to develop a block of code that will generate some slow time-base signals required by other experiments. The Lab9500 employs an AC wall transformer that can be used to get accurate 4Hz or slower clocks. The 60 Hz AC signal is isolated using an opto-isolator, and debounced and squared with a Schmitt trigger circuit. If the 60 Hz signal is divided by 15, then a 4 Hz signal is obtained. Further dividing by two will give 2 Hz and a 1 Hz (one-second) clock. The one-second will be valuable for clocking the traffic light timer or for exercising any counter or shift register.

There is probably more material here than might be accomplished in one assignment, so that the instructor should choose which parts he would like to be done.

### **Experimentation:**

- A. Design a six-bit linear counter using T-flip-flops using the 60Hz signal as the clock input. The “T” equations for such a counter are developed in Chapter 5. Define the counter bits as internal nodes, and use the LED bank, L0, L1, L2, etc. to display the counter. Note: this is close to the exact 1Hz clock we seek, but not quite. The total divide ratio is 64, so the final frequency is 15/16 Hz or .94 Hz.

Note: To obtain 60Hz at the chip JP1, JP2 and JP3 must be installed with the jumper to the right.

- B. In part A) above, the function for each T input is a single product term. However, if we wish to make a mod 15 counter, the K-map will be altered and the reductions not so clean. The obvious pattern would be to go from 0000 to 1110 (0 to 14) and then back to

zero. The result will be three product terms for the LSB and two product terms each for the other flip-flops. That arises the discontinuity in going from a count of fourteen (1110) to a count of zero (0000). Ordinarily on this count, only the LSB would flip. Here, the high three flip and the LSB must be kept from flipping. (Use a state transition table and K-map to arrive at the equations.)

- C. Realize the mod 15 counter (0-14) using D flip-flops. How does it compare with the T flip-flop version? (Use a state transition table and K-map to arrive at the equations.)
- D. There was nothing magic about using the sequence 0-14. We could have used any sequence, as long it is mod 15. Find a sequence where the code need be changed (from the straight mod 16 linear counter) for one of the four flip-flops. Is there more than one such sequence? This method is called SOP (seat-of-the-pants). Can you determine the code for this one bit without mapping it?
- E. Design a 4-bit modula 10 (decade counter). There are two ways to do this. You can make a divide-by-five followed by a divide-by-two. The output will be a square wave. Or you can make a sequence of 0 to 9 which is a divide-by- two followed by a divide-by-five. The duty cycle of the MSB is two up and eight down. Both styles have their use. Use either D or T types. You can either clock this with a one-second clock already designed, or you use one of the manual debounced pushbuttons to advance the counter.
- F. Make a mod 16 counter that resets synchronously whenever the number on switches s0 thru s1 is reached. For example, if the number on the switches is 1001 (9), then the counter will function modulo 10. Use a clock as suggested in E.
- G. Having done part A the traditional way, try taking advantage of the advanced features of ABEL and do the mod 15 counter again using: a) a truth table and b) the ABEL state machine function. Both take a little inputting, but are straightforward. A little copying and pasting will get you there in a hurry.