Lecture notes - Introduction to Counters and CPLDs

Counter Basics

A counter is one of the more useful digital circuits you will encounter. Counters are used in many applications. Examples include frequency dividers, and frequency counters. A quick example of a digital counter would be the circuit that drives your stopwatch. This circuit takes at least two inputs, a clock signal from the quartz oscillator, and an input from a button you press. It starts counting when you press the button once, and stops when you press the button again.

A digital counter has the following characteristics:

1. A maximum number of counts before it rolls over (returns to zero.) This is referred to as the counter's modulus.
2. It can count in either direction (ascending, from low to high, or descending, from high to low.)
3. It is either synchronous or asynchronous. That is, it counts with the system clock, or it counts independently of the system clock.
4. It can function either as a monostable or an astable circuit. The former means that it runs once and stops. The latter means that it will run forever, or until interrupted.

A counter is a registered circuit. That is, it has some logic that allows it to store information. This also implies that the counter circuit is a finite state machine (FSM). Therefore, it has several discrete states that it can be in. These states can only change at the rising or the falling-edge of the system clock.

For a simple counter, we may have only two states, a state A, and a state B (figure 1). In state A, the counter will continue to increment as long as the value of the count is less than the maximum value. When the counter reaches its maximum value, it goes into state B, where it is reset to zero and returns to state A.

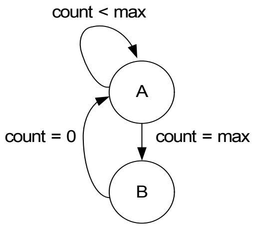


Figure 1; A two state counter

Examples: Ways to generate a sequence.

1. A four-bit ripple counter.

The following is an example of a four-bit ripple counter. This is a circuit that will count from 0 to 15, and then stop. It is constructed with a four-bit register, and an adder. The output of the register is delayed and fed into one of the inputs of the adder. This makes it a Mealy machine rather than a Moore machine. This is because its output is a function of its current inputs, and its history.

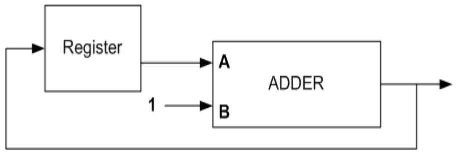


Figure 2: Block diagram of a 4-bit Ripple Counter.

In the previous figure, we have omitted the control logic in order to save space. This counter is called a modulus-16 counter, because it's four bits allows us to represent sixteen different things (2^4).

2. A grey-coded counter.

Another type of counter is a grey-coded counter. You may recall grey coding from when you were first introduced to K-Maps. Remember that a grey-code is one where only one bit changes at a time. The following tables show the difference between a three-bit modulus-8 ripple counter, and a three bit grey-coded counter.

Ripple	Grey code
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0100
0110	0101
0111	0111

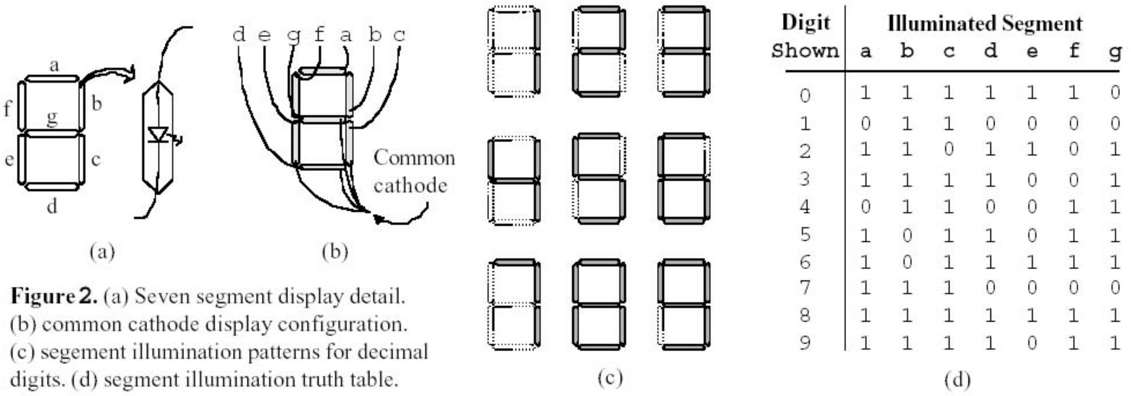
Note that in the grey-coded sequence, only one bit changes at a time. Yet, we have still represented all the possible bit combinations. Developing a grey-coded counter requires a little more work than a simple ripple-counter. For the grey-coded counter, use a finite state-machine (FSM) to do the work (i.e. figure 1)

---

7 Segment display *(from XCR datasheet, Digilent Inc.)*

A seven-segment display is build by putting together 7 LEDs as shown in figure 2.a. The XCR board has a modular 2-digit common cathode, seven-segment LED display. In a common cathode display, the seven cathodes of the LEDs forming each digit are connected to a common circuit node. In order to light up a segment; a “1” (VCC in our case) must me set up to the anode of the corresponding LED.

Therefore, in order to display digits from 0 to 9 we need to build a “translation table” as the one shown in figure 2.c and 2.d. (Figure 2 was taken from XCR datasheet, Digilent Inc.)

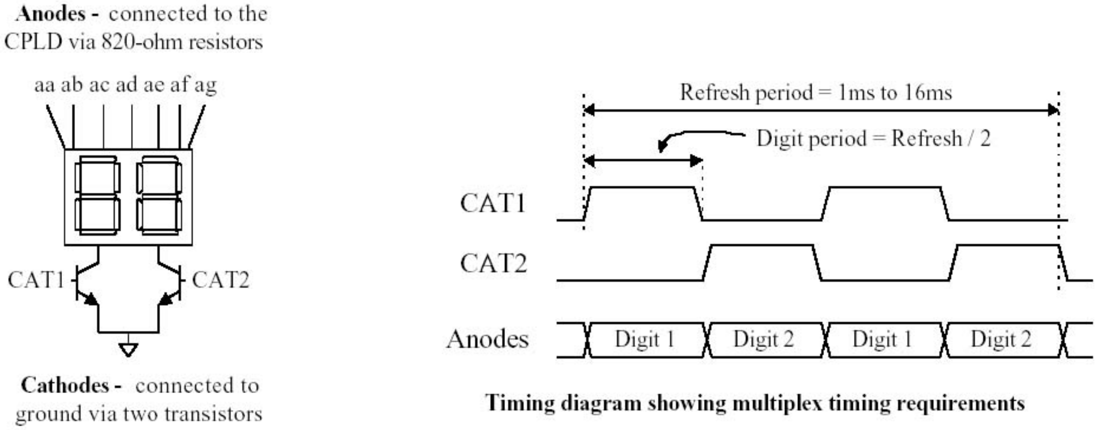


Digit Shown	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Figure 2. (a) Seven segment display detail. (b) common cathode display configuration. (c) segement illumination patterns for decimal digits. (d) segment illumination truth table.

On the XCR board, the two-digit display has two common cathode nodes labeled CAT1 and CAT2. Both cathodes, and therefore both digits, can be independently turned on and off by driving the CAT1/2 signals to a ‘1’ or a ‘0’ respectively. The anodes of similar segments on both displays are also connected together into seven common circuit nodes labeled AA through AG. Thus, each anode for both displays can be turned on and off independently. This connection scheme creates a multiplexed display, where driving the cathode signals and corresponding anode patterns of each digit in a repeating, continuous succession can create a stable 2-digit display. Even though each digit is illuminated only half time, the human eye will be “tricked” into seeing continuously illuminated digits. To appear bright and continuously illuminated, both digits should be driven once every 1 to 16ms (for a refresh frequency of 1 KHz to 60 Hz).

A display controller must assure that the correct anode pattern is present when the corresponding cathode signal is driven. To illustrate the process, if CAT1 is driven high while AB and AC are driven high, then a ‘1’ will be displayed in digit position 1. Then, if CAT2 is driven high while AA, AB and AC are driven high, a ‘7’ will be displayed in digit position 2. If CAT1/AB, AC are driven for 8 ms, and then CAT2/AA, AB, AC are driven for 8 ms in an endless succession, the display will show “17” and the observer cannot tell that both digits are not continuously illuminated. An example of a timing diagram is provided below.



Timing diagram showing multiplex timing requirements

Figure 3; Dual-digit common cathode display and timing diagram