Computational Logic Circuits (Online Lab 2)

Synchronous Counters

Leonardo Fusser, 1946995

Experiment Performed on **17 April 2020**
Report Submitted on **22 April 2020**

Department of Computer Engineering Technology
Computational Logic Circuits
Subash Handa

VANIER
CÉGEP / COLLEGE
Learning today Leading tomorrow

Table of Contents

OBJECTIVES

- To understand the function and application of synchronous counters.
- To understand how to design a synchronous counter.
- To verify and simulate the function of a 3-bit synchronous counter.

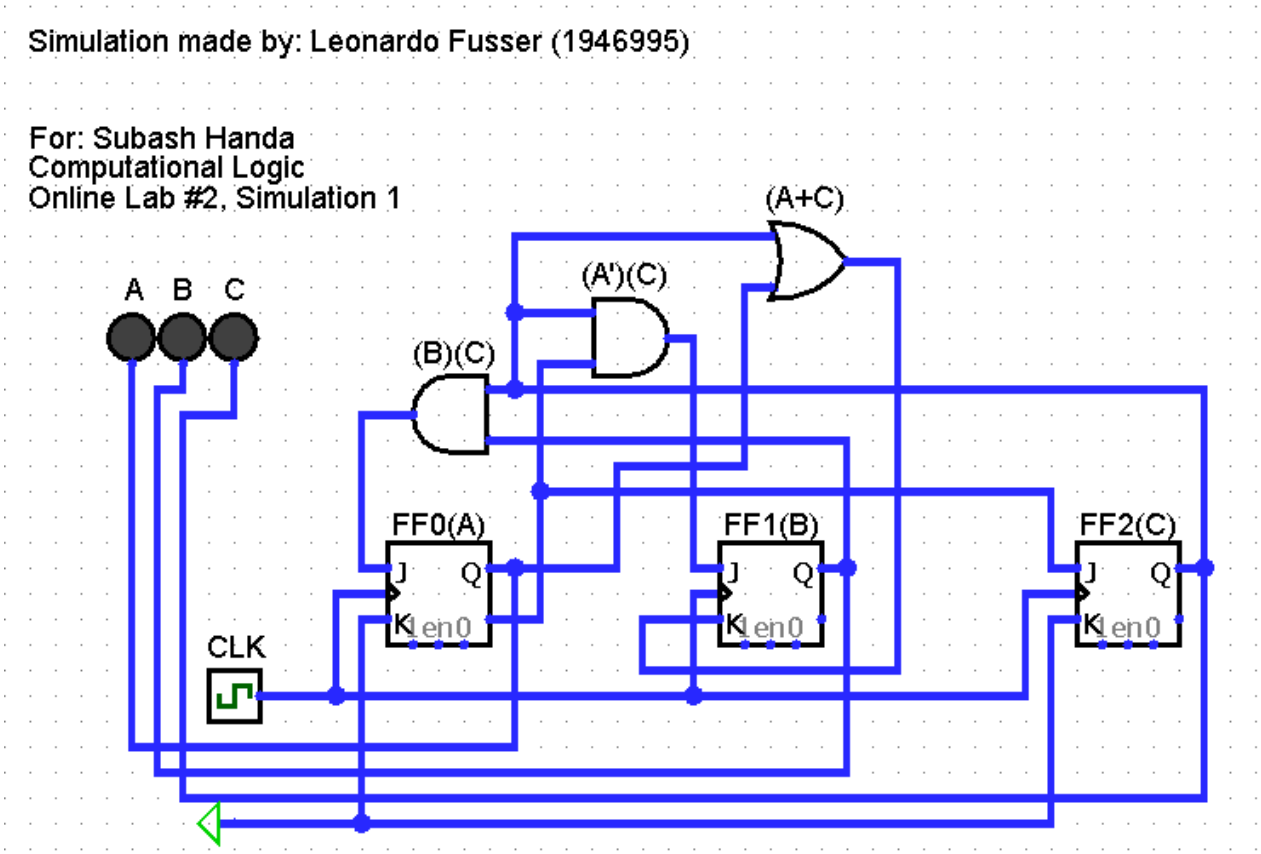
DESIGN

Experiment

- There was one simulation done in this lab. First, we were asked to do general research on synchronous counters. Following the research, we were asked to design a 3-bit synchronous counter that goes through a certain sequence. The simulations were done in "Logisim". Below outlines the work done in the lab.

SIMULATIONS

Simulations from the Experiment



Custom 3-bit synchronous counter simulation (above)

Custom 3-bit synchronous counter truth table

State	(PRESENT)			(NEXT)			J-K Flip Flops		
	A_N	B_N	C_N	A_{N+1}	B_{N+1}	C_{N+1}	JA/KA	JB/KB	JC/KC
0	0	0	0	0	0	1	0 X	0 X	1 X
1	0	0	1	0	1	0	0 X	1 X	X 1
2	0	1	0	0	1	1	0 X	X 0	1 X
3	0	1	1	1	0	0	1 X	X 1	X 1
4	1	0	0	0	0	0	X 1	0 X	0 X
5	1	0	1	0	0	0	X 1	0 X	X 1
6	1	1	0	0	0	0	X 1	X 1	0 X
7	1	1	1	0	0	0	X 1	X 1	X 1

FF0: output = A

FF1: output = B

FF2: output = C

Custom 3-bit synchronous counter k-maps

A\BC	*00*	*01*	*11*	*10*
0	0	0	1	0
1	X	X	X	X

JA = (B)(C)

A\BC	*00*	*01*	*11*	*10*
0	X	X	X	X
1	1	1	1	1

KA = (1)

A\BC	*00*	*01*	*11*	*10*
0	0	1	X	X
1	0	0	X	X

JB = (A')(C)

A\BC	*00*	*01*	*11*	*10*
0	X	X	1	0
1	X	X	1	1

KB = (A + C)

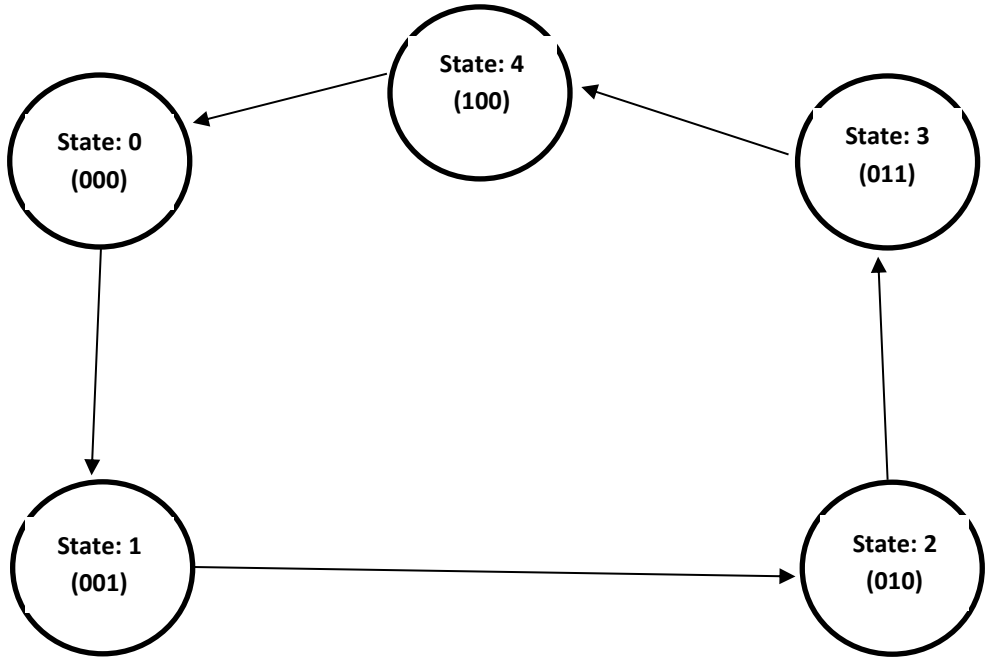
A\BC	*00*	*01*	*11*	*10*
0	1	X	X	1
1	0	X	X	0

JC = (A')

A\BC	*00*	*01*	*11*	*10*
0	X	1	1	X
1	X	1	1	X

KC = (1)

Custom 3-bit synchronous counter state diagram



QUESTIONS

Questions from the Experiment

What is a synchronous counter?

- The term synchronous refers to events that have a fixed time relationship with each other. A synchronous counter is one which all the flip-flops in the counter are clocked at the same time by a common clock pulse.

Applications of the synchronous counter?

A Digital Clock

A common example of a counter application is in timekeeping systems. Figure 9–48 is a simplified logic diagram of a digital clock that displays seconds, minutes, and hours. First, a 60 Hz sinusoidal ac voltage is converted to a 60 Hz pulse waveform and divided down to a 1 Hz pulse waveform by a divide-by-60 counter formed by a divide-by-10 counter followed by a divide-by-6 counter. Both the *seconds* and *minutes* counts are also produced by divide-by-60 counters, the details of which are shown in Figure 9–49. These counters count from 0 to 59 and then recycle to 0; synchronous decade counters are used in this particular implementation. Notice that the divide-by-6 portion is formed with a decade counter with a truncated sequence achieved by using the decoder count 6 to asynchronously clear the counter. The terminal count, 59, is also decoded to enable the next counter in the chain.

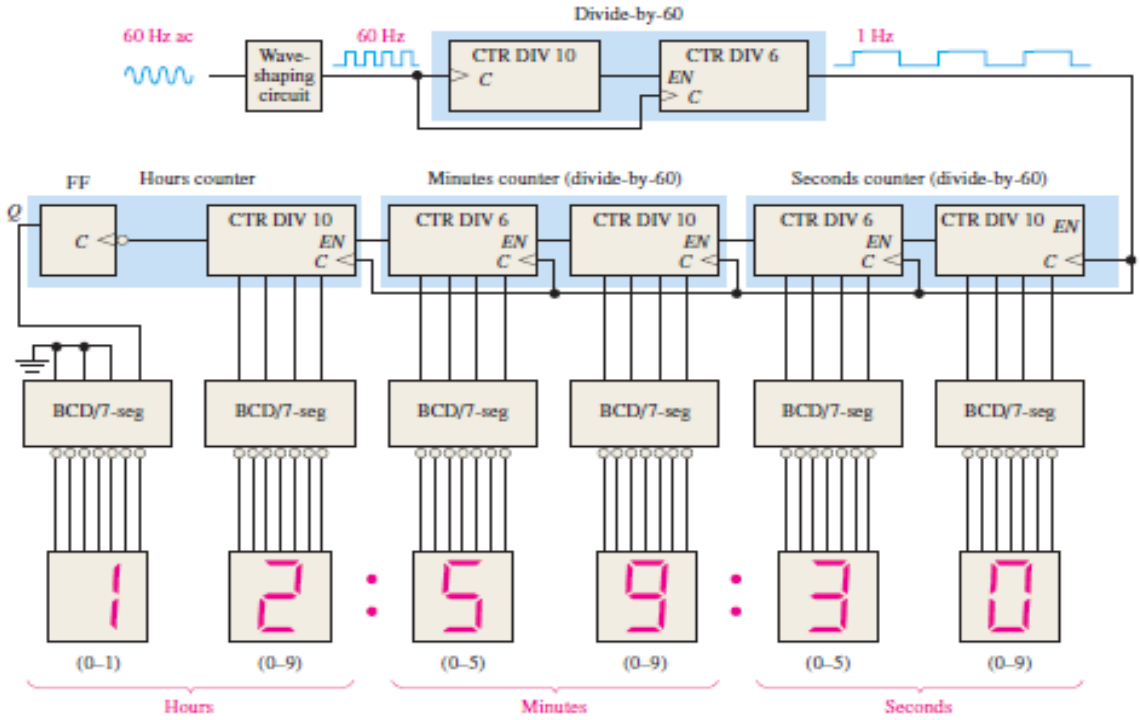


FIGURE 9–48 Simplified logic diagram for a 12-hour digital clock. Logic details using specific devices are shown in Figures 9–49 and 9–50.

The *hours* counter is implemented with a decade counter and a flip-flop as shown in Figure 9–50. Consider that initially both the decade counter and the flip-flop are RESET, and the decode-12 gate and decode-9 gate outputs are HIGH. The decade counter advances through all of its states from zero to nine, and on the clock pulse that recycles it from nine back to zero, the flip-flop goes to the SET state ($J = 1$, $K = 0$). This illuminates a 1 on the tens-of-hours display. The total count is now ten (the decade counter is in the zero state and the flip-flop is SET).

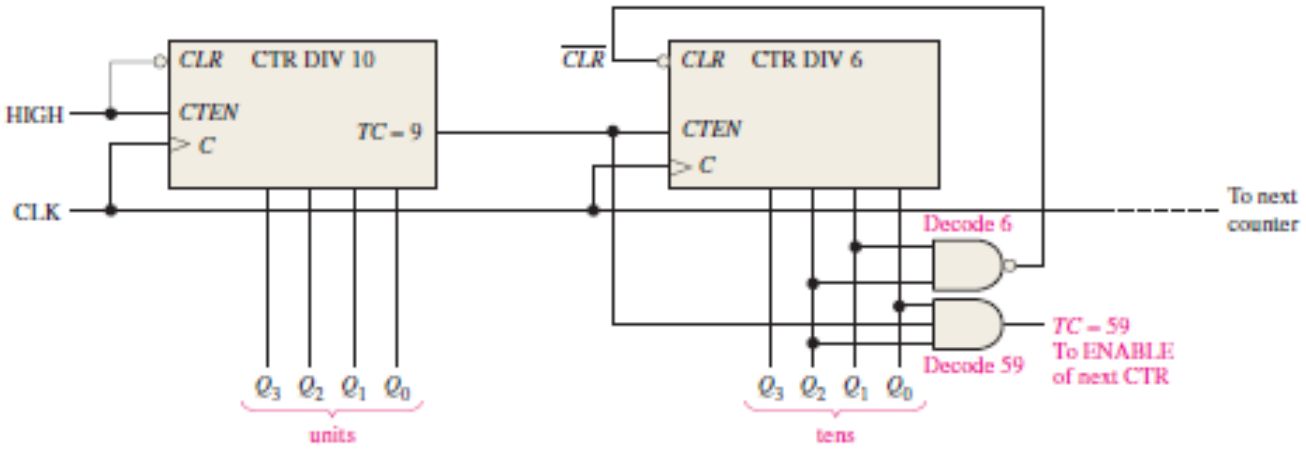


FIGURE 9–49 Logic diagram of typical divide-by-60 counter using synchronous decade counters. Note that the outputs are in binary order (the right-most bit is the LSB).

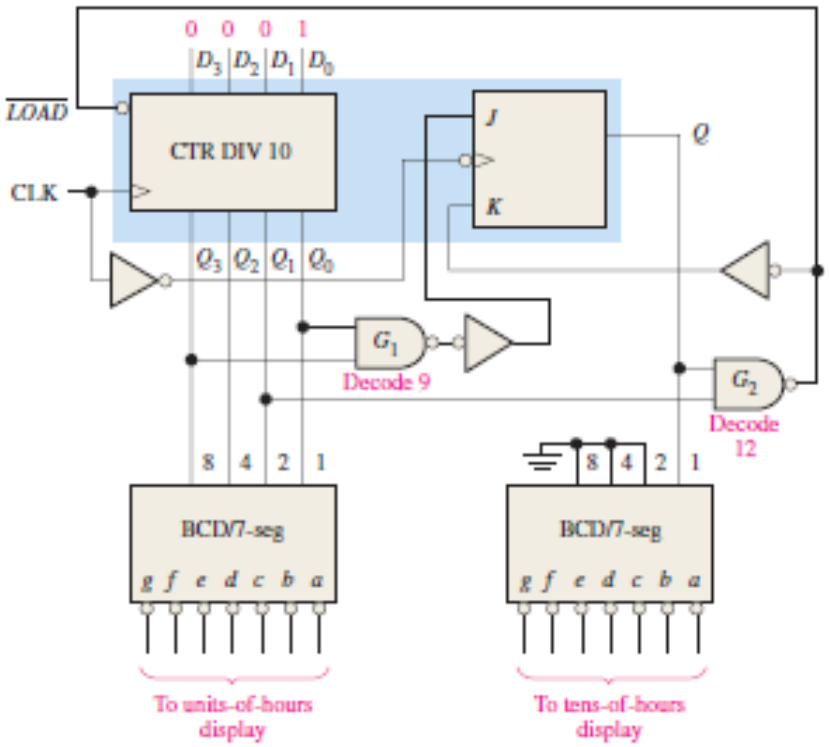


FIGURE 9–50 Logic diagram for hours counter and decoders. Note that on the counter inputs and outputs, the right-most bit is the LSB.

Next, the total count advances to eleven and then to twelve. In state 12 the Q_2 output of the decade counter is HIGH, the flip-flop is still SET, and thus the decode-12 gate output is LOW. This activates the $\overline{LOAD}$ input of the decade counter. On the next clock pulse, the decade counter is preset to 0001 from the data inputs, and the flip-flop is RESET ($J = 0$, $K = 1$). As you can see, this logic always causes the counter to recycle from twelve back to one rather than back to zero.

Source: *Digital Fundamentals, Eleventh Edition, Thomas L. Floyd, 2015*.