

Lab # 7
Sequential Circuit Design using PLDs

A. **Objective**

The objective of this laboratory is to introduce the student to the use of sequential circuit design using Programmable Logic Devices (PLDs). Sequential circuits will be designed by the student using state equations and D flip-flops and the circuits will be implemented and tested using a PLD.

B. **Materials**

Breadboard	GAL22V10 Programmable Logic Device (PLD)
5V Power Supply	Common-anode 7-segment display
ChipMax Universal Programmer	7447 BCD-to-7-segment decoder/driver
Aldec Active-HDL Software	Seven 220 or 330 ohm resistors
555 Timer	Various resistors and capacitors

C. **Reference**

Refer to the following items (available on the instructor's web page):

- *“Combinational Logic Circuits using Aldec Active-HDL”*
- *“Sequential Logic Circuits Using Aldec Active-HDL”*
- *“ChipMax Universal Programmer Instructions”*
- GAL22V20 Data Sheet

D. **Introduction**

Using Aldec Active-HDL will eliminate the need for a detailed design of a synchronous sequential circuit as the user will only need to specify the state table (as well as simulate the design and implement the design into a PLD). However, in order to give the student an appreciation of the design work that would be required by hand and the complexity of the circuit to be implemented into the PLD, students will be required to perform a hand design as well.

Sequential circuit design using state equations

Several design methods are available for designing synchronous sequential circuits, including the excitation table method, design by state equations, and design using the “one-hot” method. Many PLDs only support D flip-flop designs (including the GAL22V10) so this lab will focus on the state equation method which is well suited for D flip-flops.

The general form of the state equation for a D flip-flop is:

$$\boxed{Q(t + 1) = D}$$

So the input for each D flip-flop is simply determined by finding an expression for the next state for that flip-flop. An example using this method is shown on the following pages.

Example: Design a 4-bit counter using D flip-flops and the state equation method.

A 4-bit counter, also called a modulo-16 counter, counts in the sequence 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 and repeats. The state diagram is shown in Figure 1 below.

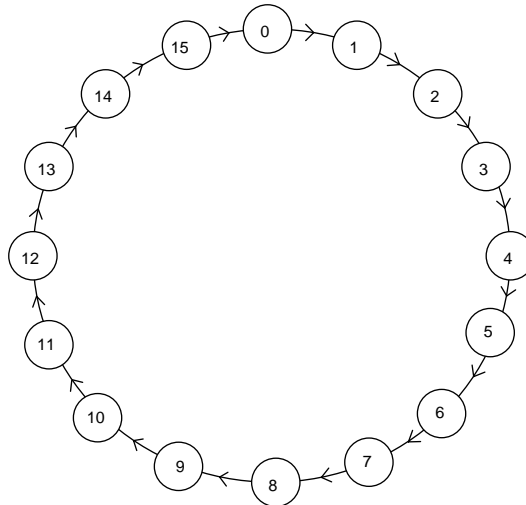


Figure 1: State diagram for a 4-bit counter

The corresponding state table is shown in Figure 2 below. Note that the state is shown in decimal form in the state diagram, whereas it is shown in binary form in the state table with bit D as the MSB.

Present State				Next State			
D	C	B	A	D	C	B	A
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

Figure 2: State table for a 4-bit counter

The characteristic equation for a D flip-flop is very simple: $Q(t + 1) = D$.

So the expression for the next state is simply connected to the D input on the flip-flop.

Expressions for the next state for each of the four flip-flops is determined using Karnaugh maps shown in Figure 3 below.

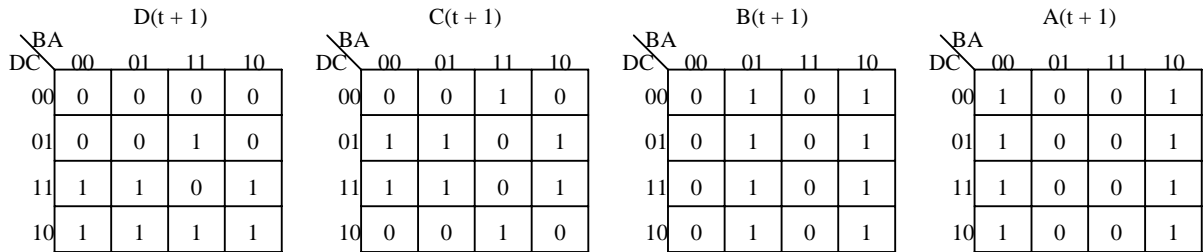


Figure 3: Karnaugh maps for the state equations for the 4-bit counter

Minimal SOP expressions for each output yield the state equations shown below in Figure 4:

$$\begin{aligned}
 D(t + 1) &= D \cdot \bar{C} + D \cdot \bar{B} + D \cdot \bar{A} + \bar{D} \cdot C \cdot B \cdot A \\
 C(t + 1) &= C \cdot \bar{B} + C \cdot \bar{A} + \bar{C} \cdot B \cdot A \\
 B(t + 1) &= B \cdot \bar{A} + \bar{B} \cdot A \\
 A(t + 1) &= \bar{A}
 \end{aligned}$$

Figure 4: State equations for the 4-bit counter

The state equations above are implemented in the circuit shown below in Figure 5.

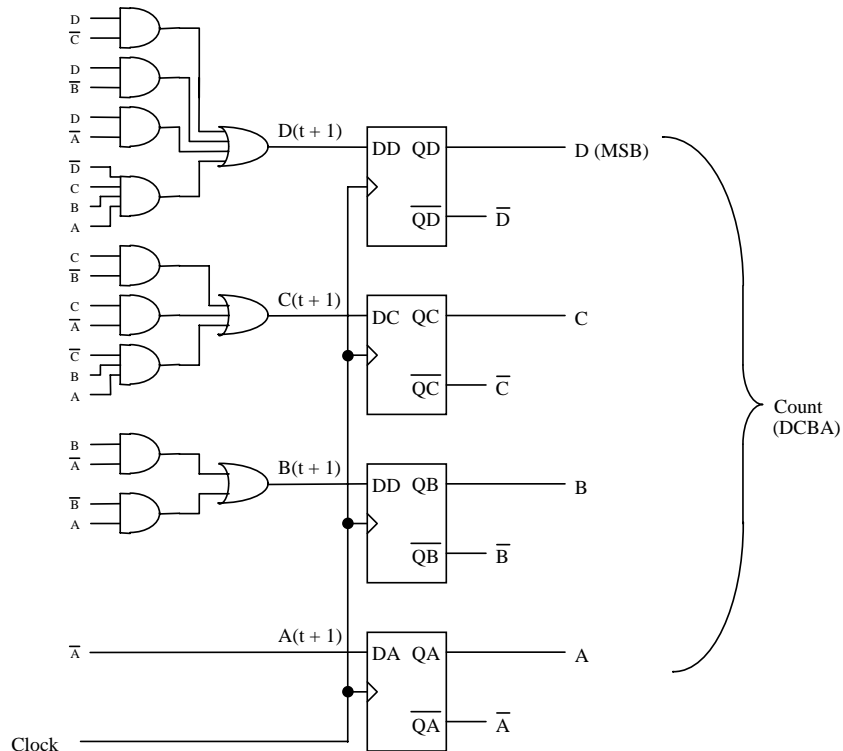




Figure 5: Logic Diagram for the 4-bit counter

E. Preliminary Work

1. Draw the state diagram for the counter described below: Each student should design an up/down counter that will count out each of the unique digits (7 or below) of his or her EmplID in the order in which they occur, followed by the all digits from 0 to 7 that do not occur in the EmplID from highest to lowest. Note that all counting sequences will have all eight digits. The counter should include a count direction control, X, such that the counter will count in the manner described above when $X = 1$ and in reverse order when $X = 0$. An example is shown below.

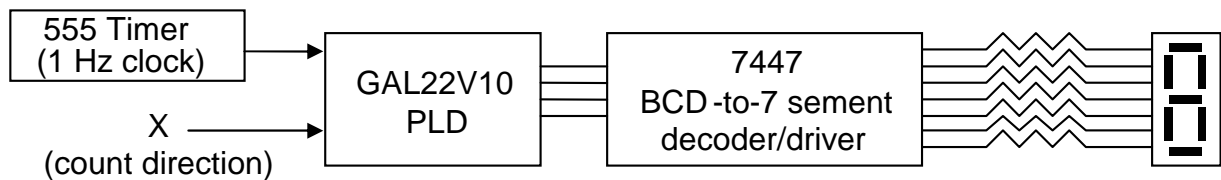
Example: EmplID = 1468443

Counting sequence when $X = 1$: 1, 4, 6, 3, 7, 5, 2, 0

	
Digits in your EmplID (7 or less) in the order in which they occur	Digits from 0 to 7 that are not in your EmplID from highest to lowest

Counting sequence when $X = 0$: 0, 2, 5, 7, 3, 6, 4, 1 (i.e., reverse order)

2. Draw a detailed logic diagram for the counter to be implemented. Note that the same 555 timer circuit used in Lab 6 can be used in this lab also.



3. Design the counter described above using D flip-flops and the state equation method. Include the state table and Karnaugh maps used to determine the D flip-flop inputs. Draw the final circuit (assume that gates with any number of inputs are available). What is the total number of gates required (including the 3 flip-flops)? Note that only one PLD will be used to replace this circuit. Note that this step is not necessary to perform the lab, but is used to give the student a better idea of what is actually being programmed into the GAL22V10.

F. Laboratory Work

1. Ask the instructor to check your state diagram before proceeding to specify the design using the Aldec Active-HDL software.
2. Specify and simulate your design using Aldec Active-HDL. Refer to the handout “*Sequential Logic Circuits Using Aldec Active-HDL.*” Be sure that the design simulates correctly before implementing it into a GAL22V10.
3. Implement your design into a GAL22V10. Refer to the handout “*Combinational Logic Circuits Using Aldec Active-HDL.*” Program the PLD using the ChipMax Universal Programmer. Be sure to specify a design based on “encoded states” rather than a “one-hot” design.

4. Build and test your circuit. Clock the circuit using a 555 timer circuit configured as a 1 Hz clock. Display the output count on a 7-segment display. Demonstrate proper circuit operation to the instructor.
5. Print out the following items from Aldec Active-HDL to include with your lab report:
 - A) State diagram
 - B) VHDL file (.vhd)
 - C) A waveform file – verify that it is correct (write the count next to the waveforms by hand)
 - D) The Chip Report