

# Lecture Notes - Introduction to Sequence Detectors and CPLDs

---

## Sequence Detectors

---

Sequence detection is the act of recognizing a predefined series of inputs. In our case, we are looking for very simple things very slowly. We are looking for a series of ones and zeros similar to 1011. The overwhelming beauty to programmable logic is its ability to work with DSP (digital signal processing). The capability of processing a great deal of information quickly can not be overemphasized. Suppose you were working at the VLA in Socorro and you wanted to scan thousands of bits of digital information looking for a series of bits that represents an RF signal from a distant galaxy. Or, suppose you wanted to scan a digital image of a human lung looking for a digital signature that would represent a cancerous cell. These are very broad interpretations but I believe you can see where the benefits come in with being able to recognize a specific sequence.

A sequence detector is sequential circuit which is basically a circuit that can store information between operations. There are two main to model for sequential circuits: Mealy and Moore model. A Mealy model circuits the output depends on the inputs and the state of the system, in a Moore model, the output of the system only depends on its state. Please review your class notes on state diagrams for sequential circuits.

### Design a Sequence Detector

For this lab we'll need to create a sequence detector with the following characteristics:

- Serial input X
- The sequence to be detected is 10010.
- Five bits wide output Z with value 00000 when no sequence is detected, 00001 when the first bit of the sequence is detected, 00010 for the second bit, 00100 for the third bit, 01000 for the fourth and 10000 when the whole sequence is detected.
- If input Y is one, the system is reset to output 00000.

This is a sequential circuit; therefore we will need a clock. Since the XCR board we are working with does not include a clock, we will use the slowest clock we have: your finger!

**First step;** Create the state diagram with all possible sets of inputs. Refer to figure 1.

**Second step;** Draw a block diagram of the system. Refer to figure 2.

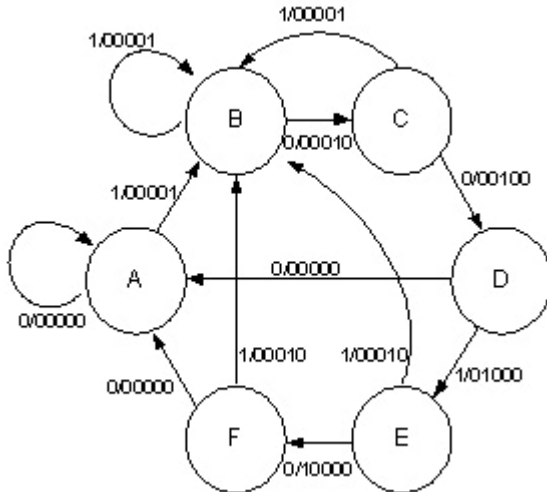


Figure 1. State diagram.

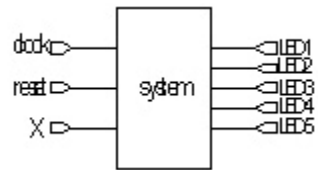


Figure 2. Block diagram

---

## Complex Programmable Logic Device (CPLD)

---

A Complex Programmable Logic Device (CPLD) is a major piece of programmable logic in the electronics industry today. It is basically constructed of interconnected Simple Programmable Logic Devices (SPLDs). The main idea behind CPLDs is that it can be used to replace discrete logic. Since a CPLD doesn't require external devices for configuration, it provides a single chip logic solution.

The CPLD we'll use in this laboratory session is a XCR3064 with 1,500 system gates. A system gate is a standard for representing the number of two input NAND gates that can be emulated using this device. Remembering that the NAND gate is the "universal" gate and that any logic device can be built with enough of them, this is another way of saying that this device can make any project that can be made with 1,500 NAND gates. Imagine having to wire all of them up on a protoboard.

Two views of a CPLD architecture are shown in figure 1. A CPLD is basically a huge interconnect array between SPLDs (figure 1.a). A SPLD block is basically Programmable Array Logic (PAL) or Programmable Logic Arrays (PLA) and Macrocells. PAL and PLA are the predecessors of CPLD. More details on PLA and PAL will be seen in classes.

A Macrocell consists typically of registers and a combinatorial path (it may use one or another). It also provides feedback to the Interconnect Array and I/O cells (buffers for inputs and 3-state outputs when configured as inputs). A block diagram of a Macrocell is shown in figure 2.

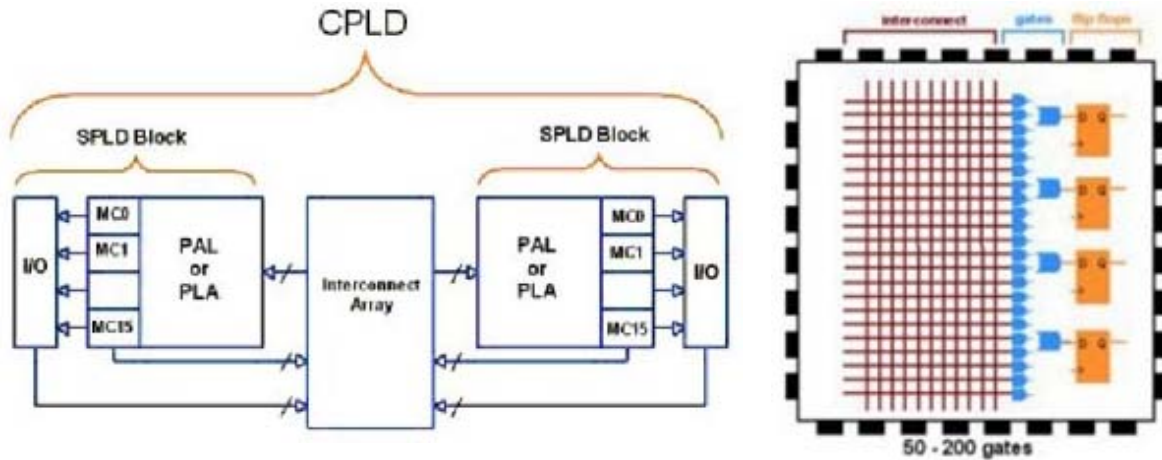


Figure 1; CPLD architecture

A CPLD is greatly praised for its nonvolatile characteristics and its low power consumption. You can remove the power but when you return it, the program will still be there. Here is a small idea of why we are using programmable logic. Suppose you were to design an alarm system. When you did all your final crunching, you determined that you were going to need 34 NAND gates, 22 OR gates, 77 AND gates, and 23 XOR gates. Using the hardware that you have developed projects with up to this point, you can see what an enormous amount of work it would be to wire all of this up. The thought behind programmable logic is that this entire design could be created with just one CPLD. Also, since it is programmable, if you have a minor change from your user, you could easily reconfigure the chip without having to do major rewiring.

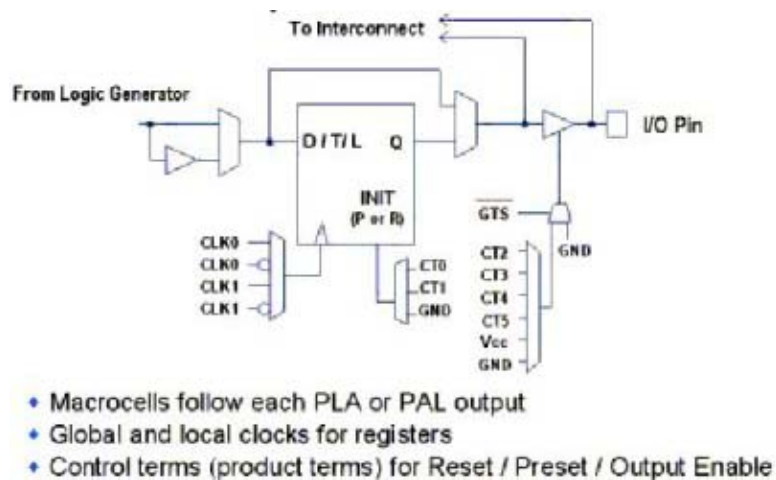


Figure 2; Macrocell block diagram