Chapter 7 Objectives

At the conclusion of this chapter, the reader will be able to:

- List the components in a Phase Locked Loop, explaining the purpose of each one.
- Describe the three operating states of a PLL.
- Given the parameters of the loop, calculate the frequencies in each part of a PLL.
- Draw a block diagram of a direct PLL frequency synthesizer.
- Calculate the frequencies and divisors needed in a direct PLL synthesizer.
- Calculate the frequencies in the loop of an indirect PLL synthesizer
- Describe the software events necessary for control of a PLL synthesizer.
- Draw a block diagram of a DDS frequency synthesizer
- Calculate the various parameters for a DDS frequency synthesizer.
- Given block and schematic diagrams of a frequency synthesizer, develop a plan for troubleshooting it.

Chapter 7: Systems for Frequency Generation

All radio transmitters and receivers use *oscillators* to provide needed frequencies. An oscillator is a stage that converts DC from the power supply into an AC output signal at a specified frequency. Up to this point we have studied two ways of controlling the frequency of an oscillator. These two methods are *discrete LC control* and *crystal control*.

An oscillator's frequency can be controlled by a discrete LC tank. This approach is simple, and the oscillator's frequency is easily adjusted by varying either the L or C component values within the tank. However, this approach doesn't provide a very stable output frequency; the Q of the LC tank is too low to keep the frequency steady.

Crystal control of an oscillator's frequency provides rock-stable output. This would be the ultimate choice for all transmitters and receivers, except that the frequency of a crystal oscillator cannot be appreciably changed without replacing the crystal. When many different operating frequencies are required, this approach becomes very expensive.

A *frequency synthesizer* is a circuit that *synthesizes* or "builds" new frequencies. These new frequencies are based on a highly stable frequency source, which is usually a single quartz crystal oscillator. The stable frequency source is called the *reference* or *master* oscillator.

Modern frequency synthesizers are digitally controlled. They make possible all sorts of products and applications, from electronically tuned shirt pocket stereo receivers, to sophisticated commercial communication and navigation equipment. Digital control of frequency allows microprocessor control of radio features. The frequency synthesizer in a typical product is merely an input/output (I/O) device connected to the controlling CPU. Software calls the shots, and analog hardware does the work.

The mix of analog and digital hardware in a frequency synthesizer can be very intimidating to the technician, especially when there is a hidden piece of computer software running the show. No matter how complex, all frequency synthesizers are based on a few basic ideas. By learning these principles, you'll be well prepared for working with and troubleshooting these fundamental communications building blocks.

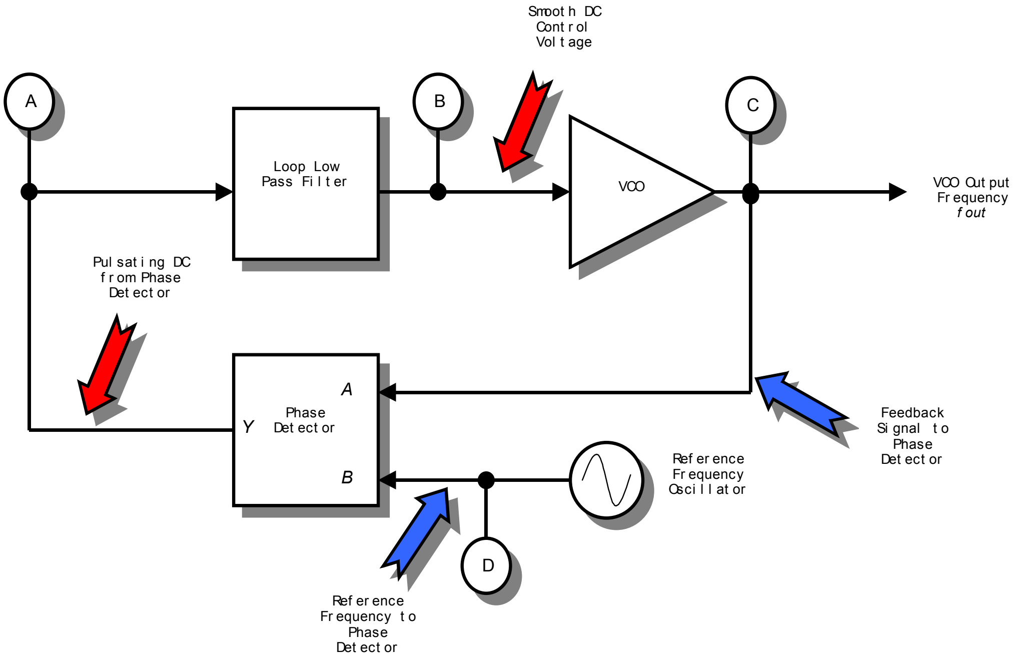
7-1 The Phase Locked Loop

The phase-locked loop, or PLL, is one of the most useful blocks in modern electronic circuits. It is used in many different applications, ranging from communications (FM modulation, demodulation, frequency synthesis, signal correlation), control systems (motor control, tracking controls, and so on), as well as applications such as pulse recovery and frequency multiplication. Knowing PLLs can really boost your "tech IQ!"

PLL Theory

A PLL is a *closed-loop* system, whose purpose is to lock its oscillator onto a provided input frequency (sometimes called the *reference* frequency.) A closed-loop system has feedback from output to input. In a PLL the feedback is negative, meaning that the system is *self-correcting*. When we say that the PLL's oscillator is *locked* onto the reference, we mean that there is *zero frequency difference (error) between the PLL oscillator and the reference frequency*. It might not make sense at this point as to why we would want to "lock" one oscillator onto another's frequency. Can't we just take the output from the reference oscillator and be done?

There are two reasons why we will want to do exactly this. First, the PLL provides *filtering* action. A PLL can lock onto a noisy reference signal, providing a filtered output that is relatively free of noise. Second, by modifying the PLL feedback loop we can derive new frequencies from the reference signal. We can build a “tunable” frequency source based on a rock-solid frequency crystal frequency reference. This is *frequency synthesis*. Figure 7-1 shows the basic elements in a PLL.



Smooth DC Control Voltage
A
B
C
Loop Low Pass Filter
VCO
VCO Output Frequency *f out*
Pulsating DC from Phase Detector
A
Y
Phase Detector
B
Feedback Signal to Phase Detector
Reference Frequency Oscillator
D
Reference Frequency to Phase Detector

Figure 7-1: Phase locked loop block diagram

A PLL consists of a *voltage controlled oscillator*, or *VCO*, a *phase detector*, and a *loop filter*. Each of these components has a special job in keeping the PLL locked onto the reference frequency.

Voltage Controlled Oscillator

A PLL has a special oscillator, a VCO. Previous oscillators we have studied depended on either an LC resonant circuit or a crystal to determine their frequency. The output frequency of a VCO depends on an LC or RC circuit, and a *control voltage*. The LC or RC portion of the circuit determines the approximate frequency range that the VCO will operate in, and the control voltage moves the VCO frequency up or down within that range.

Most technicians think of a VCO as a *voltage to frequency converter*, since the input to a VCO is a DC control voltage, and the output of a VCO is a varying frequency. Figure 7-2 is a graph of the *transfer characteristic* of a simple VCO circuit. The transfer characteristic is the input-output relationship.

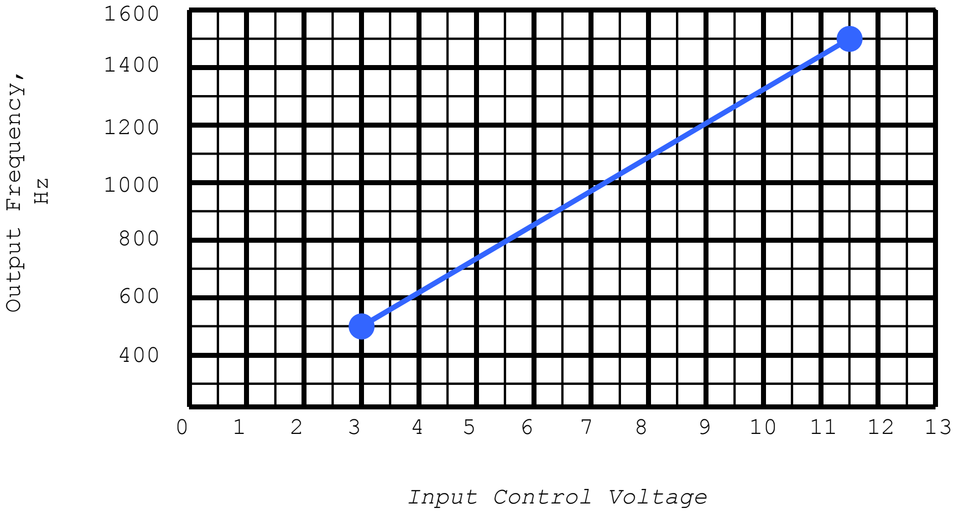


Figure 7-2: Transfer characteristic of a VCO

Note the units on the axes in Figure 7-2. The horizontal axis has units of voltage, and the vertical shows frequency. This shows us that the output frequency of the device depends upon the input control voltage.

There are definite limits on how high and how low the frequency of the VCO can go. These limits are inherent to any VCO, and are determined by the circuit designers. Most practical VCO circuits do not operate over more than about a 3:1 frequency range.

Example 7-1

What will the output frequency of the VCO of Figure 7-2 be if the control voltage is (a) 3 V ; (b) 8 V ; (c) 12 V

Solution

The output frequencies can be read directly from the graph of Figure 7-2:

a) When $Vc = 3$ V, $f_{out} = \underline{\underline{500\text{ Hz}}}$

b) When $Vc = 8$ V, $f_{out} = \underline{\underline{1100\text{ Hz}}}$

c) When $Vc = 12$ V, $f_{out} = \underline{\underline{\text{undefined}}}$. The VCO isn't designed to accept a control voltage above 11.5 volts. We know this because the graph stops at $Vc = 11.5$ V.

Phase Detector

The purpose of a PLL is to lock the VCO frequency onto the reference frequency. In order for this to happen, a decision must be made about the VCO's frequency: Is the VCO frequency too high, too low, or just right?

The decision making process must involve feedback. It's very much like the cruise control in a car. Suppose that you have set the cruise speed in your car to be 70 MPH. Somewhere, a sensing device measures the car's speed. That speed information is fed into the cruise control unit. If the car is moving too slowly (speed < 70 MPH), you know that the cruise control will respond by opening the throttle a little wider, which will bring the vehicle speed up to the desired point. The opposite will happen if the car is moving too quickly; the throttle is closed to slow down. The speed of the car is not expected to always be *exactly* the same as the set point of the cruise control. There is always a small error, usually +/- 2 MPH. This is necessary to prevent stability problems ("hunting").

In a PLL, the VCO control voltage is like the throttle in a car, and the resulting VCO frequency is analogous to the car's speed. The *phase detector* is the decision maker that compares the VCO frequency to that of the reference. In Figure 7-1 you can see that the VCO output signal is fed back into the A input of the phase detector. The B input of the phase detector sees the reference signal. Unlike the cruise control in a car, the phase detector decision-maker will not be satisfied until there is *zero frequency difference (error)* between the VCO and the reference source. In fact, this can be stated as a simple law:

Finley's Law for Phase Detectors

If the two inputs of a phase detector are not at <u>exactly</u> the same frequency, then the phase detector output will be in either positive or negative <u>saturation</u>.

How can the phase detector possibly achieve *zero* error in frequency? The answer becomes evident when we look at the relationship between frequency and phase. Suppose that we stretch the car analogy a little further by imagining two cars traveling in the same direction down a four-lane highway. The "frequency" of each car is indicated by its speedometer. The "phase" of the cars is just their relative position on the highway. In Figure 7-2, although both cars are not perfectly in phase, they are both moving at exactly the same speed. Their "frequencies" are identical. *The phase of two signals does not have to be the same for their frequencies to be identical. However, the phase error must be constant.*

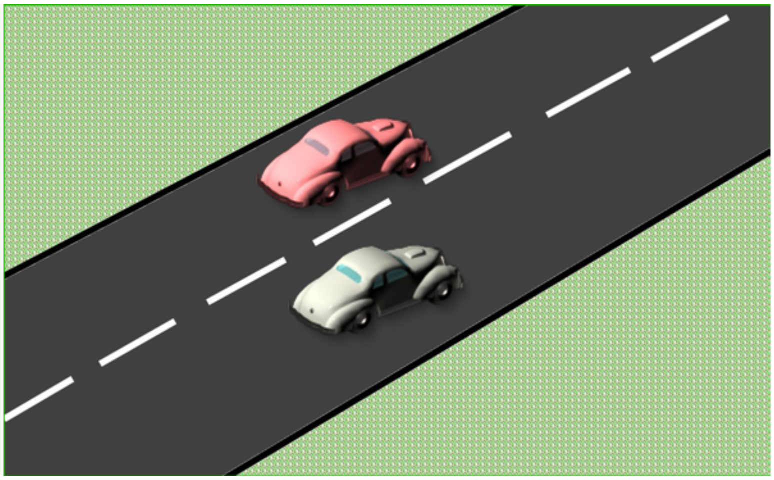


Figure 7-3: The frequencies are identical, the phases are not. The phase error is static.

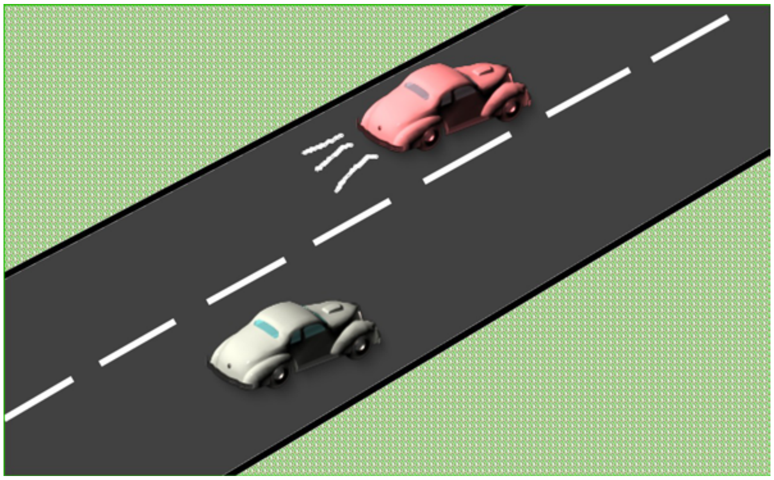
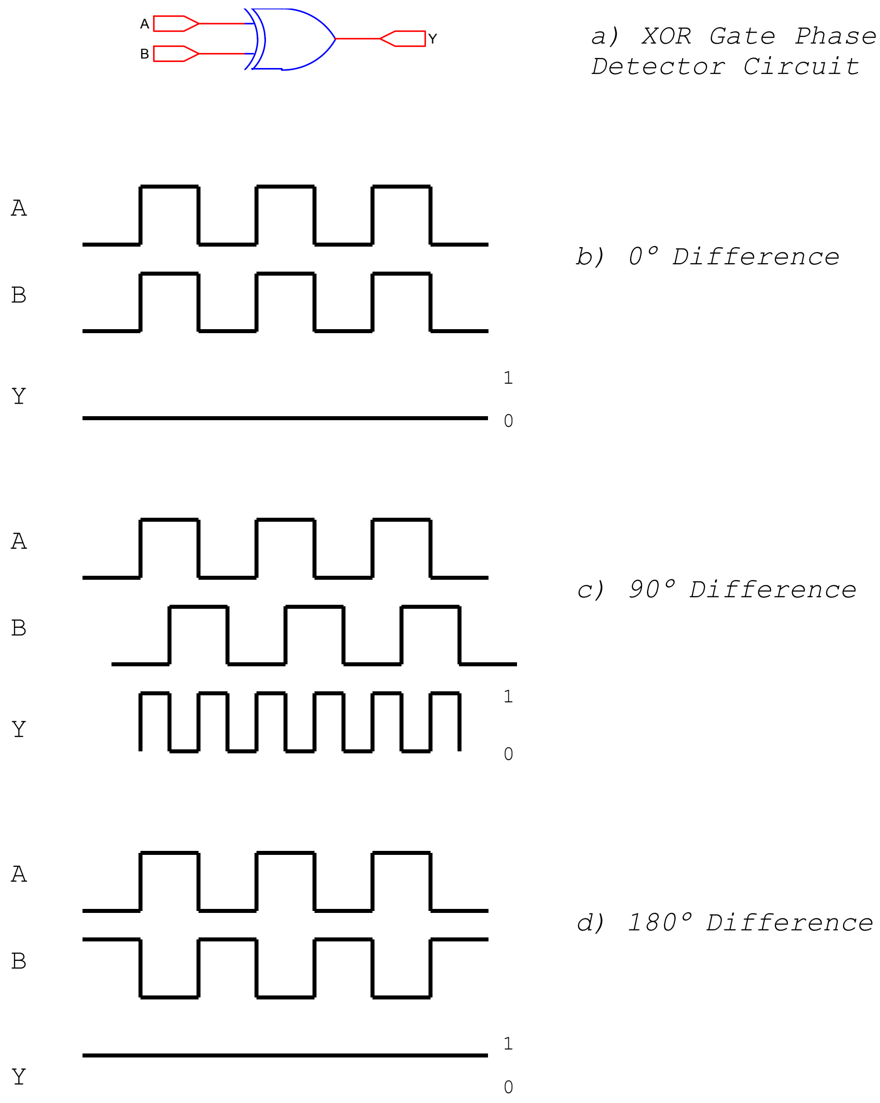


Figure 7-4: The phase error is increasing; the frequencies are unequal

In Figure 7-4, the left car has sped up. The two cars are no longer in a fixed relationship with each other. There is not only a phase difference between them, but also an *increasing* phase difference (error). Their speeds (frequencies) are no longer equal.

A phase detector achieves zero frequency error by comparing phase. When the phase difference between two signals is constant, their frequencies are identical. This is why Finley's law is true for phase detectors. This special property makes the phase detector an excellent frequency "referee" for the PLL. It also explains why there is always zero frequency error in a PLL once it is in lock.

Most phase detectors are digital. The output of a phase detector is pulsating DC with a varying duty cycle. One of the simplest possible phase detectors is an exclusive-OR logic gate, as shown below in Figure 7-5(a).



a) XOR Gate Phase Detector Circuit

b) 0° Difference

c) 90° Difference

d) 180° Difference

Figure 7-5: XOR-type phase detector and circuit waveforms

An exclusive-OR gate gives a "1" output whenever the inputs are opposites, and a "0" output when the inputs are the same. In Figure 7-5(b), the two inputs are precisely in phase, which means they're the same all the time. The gate always produces a "0" output, which corresponds to 0 volts.

In Figure 7-5(c), the "B" input is *leading* the "A" input by 90°. Now the gate inputs are different at parts of the cycle, and consequently, the output "Y" goes high precisely one-half of the time. We would say that its duty cycle is 50%, and that its average voltage is $V_{cc}/2$.

As we increase the phase difference to 180°, the output stays high all the time. The duty cycle is now 100%, and the average voltage is V_{cc}. The XOR gate has converted the input phase difference into an average DC voltage. This DC voltage is pulsating at twice the frequency of the input signals. Figure 7-6 shows the transfer characteristic of this phase detector.

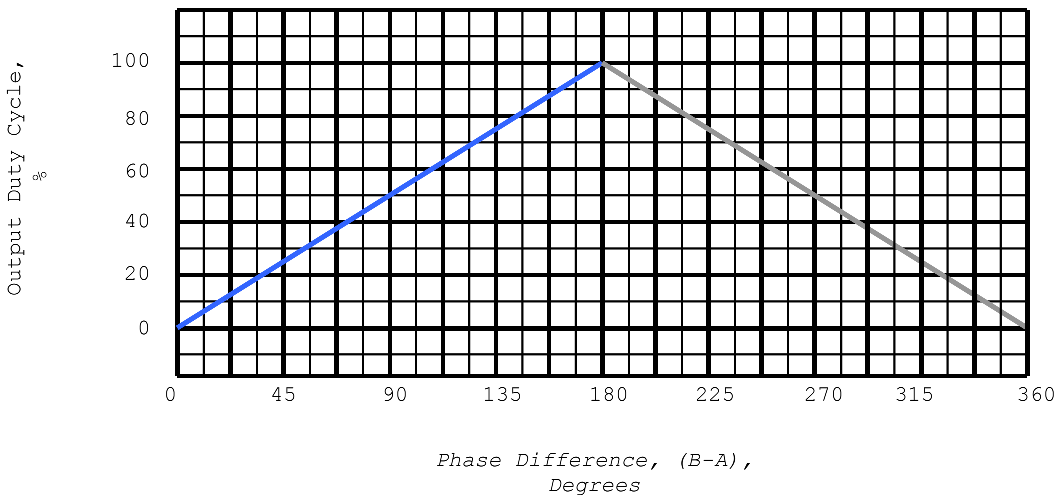


Figure 7-6: Transfer Characteristic of the Exclusive-OR Phase Detector

The blue graph in Figure 7-6 reflects operation from a 0 to 180° difference. As the phase difference increases, the average output voltage increases. But something strange happens as the phase difference passes 180° (gray region of graph) -- the output voltage starts *decreasing*! The XOR phase detector simply cannot operate over more than a 180° range.

In order to overcome this limitation, more complex logic circuits are used in actual PLL chips. The XOR phase detector has another problem. It is sensitive to the duty cycle of the two input signals, which are required to be square waves. If either signal varies in duty cycle, the output will falsely indicate a phase change. A practical phase detector usually includes a Schmitt trigger on each input in order to convert the signals to square waves, and an edge-detector circuit to overcome the duty cycle problem.

Loop Filter

The "Y" output of the phase detector in the PLL of Figure 7-1 is a pulsating DC voltage with a varying duty cycle. The bigger the phase difference becomes (within certain limits, of course), the larger this duty cycle becomes, and the larger the average voltage being fed back into the VCO on top. This voltage will tend to correct the frequency of the VCO, either raising or lowering its frequency. But there's a problem here.

The VCO needs a nice, steady DC voltage at its control voltage input. Can you imagine the effect of the pulsating DC on the VCO? Think of a car that only has two throttle positions, wide open and off. The desired speed is 70 MPH. We're traveling 69 MPH, which is too slow -- so we must choose the *wide open* throttle position. The car lurches forward with this throttle application, *overshooting* the target speed of 70 MPH. Now our only choice is to totally close the throttle and jam on the brakes. The passengers are thrown forward as the car rapidly decelerates. The cycle repeats, over and over. The motion of the car isn't very smooth at all, although its average speed is very close to 70 MPH!

The same thing would happen to the VCO. We'd like its output frequency to be steady, like the reference input. What we need to do is smooth out the pulsating DC from the phase detector into a steady *DC average* voltage. This is the purpose of the *loop filter*. This filter in effect smoothes the rough phase detector output waveform into a steady DC voltage for the VCO. The VCO will then be able to smoothly track the input reference frequency.