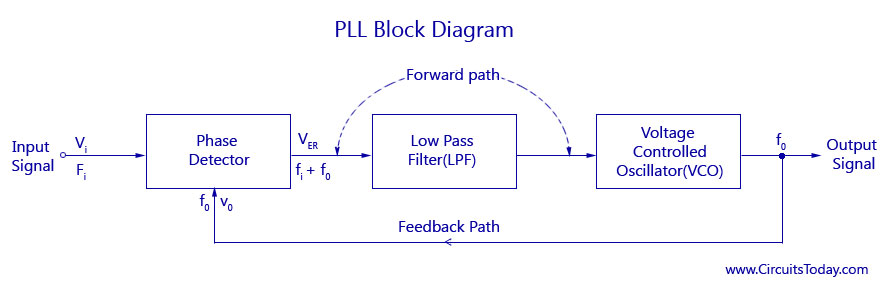
# **Homework 7**

## **What is the Phase Locked Loop?**

Is a useful circuit block which is widely used in radio frequency or wireless application. The way it works is that it takes in a signal to lock in and then can output that signal from its own internal VCO.

## **Diagram of the PLL**



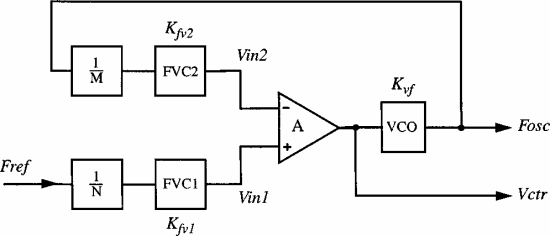
## **How does the Phase Locked Loop work?**

First, the input signal Vi with an input frequency get into a phase detector, which compared the input frequency with the feedback frequency. The output of the Phase Detector is an error voltage Ver which is a DC voltage. This voltage is the input for the next stage name Low Pass Filter (LPF). In this stage the high frequency noise is removed and produced a steady DC level. The final stage is the Voltage Controlled Oscillator (VCO) where the DC level is passed. The output frequency of the VCO is directly proportional to the input signal.

## **What is the Frequency Locked Loop?**

It is similar to a PLL, where it generates an output signal which tracks an input reference signal. However, the output, in this case, is synchronized only in frequency with the input signal and not in phase, therefore, the locking time of this FLL would be very short. The principle operation is bases on a frequency comparison instead of phase comparison.

## **Diagram of the FLL**



## **How does the FLL work?**

First, the input frequency of the reference signal is divided by N and converted to a voltage by the FVC1. Also, the VCO oscillating frequency is divided by M and converted to a voltage. The difference between these two voltages is then amplified by the high gain opamp and the result is Vctr which is used to control the output frequency of VCO.

## **What is button debouncing?**

It is a way to control bouncing which is the tendency of any two metal contacts in an electronic device to generate multiple signals as the contacts close or open. These transitions are due to mechanic and physical issues. Basically, means that when you pressed a button you may read multiple presses in a very short time fooling the program you are using. Debouncing checks twice in a short period of time to make sure the button is pressed.

## **What are some techniques in hardware and software to debounce a button?**

Hardware:

* Most microcontrollers have internal pullup resistors that can be activated for any given GPIO pin. This lets you establish a definite logic state on a button when it’s not pressed by pulling the voltage back up from ground.
* The simplest which works most of the time is debounce with a RC filter. If you select your resistor and your capacitor for the product to be large enough, it will work most of the time.
* To get it right all the time is using a logic chip like: 74HC14 or 40106.

Software:

* Attempts to wait until after the bouncing has stopped before declaring a button press or release. If the switch is still bouncing after a delay time, it delays again until it stables.
* There is the counter-or integrator-based debouncers, which is the equivalent to the RC filter.
* Pattern-based debouncer: takes the overall pattern of the switch’s voltage output over a relatively long time period into a account.

**References:**

*Debounce.* (2015). Arduino CC. Available at: <https://www.arduino.cc/en/Tutorial/Debounce>

Djemouai, A. (2001). *New frequency-locked loop based on CMOS frequency-to-voltage converter: design and implementation.* Available at: <https://ieeexplore.ieee.org/document/938354>

*PLL Phase Locked Loop Tutorial & Primer.* (2019). Electronics Notes. Available at: <https://www.electronics-notes.com/articles/radio/pll-phase-locked-loop/tutorial-primer-basics.php>

*PLL-Phase Locked Loops.* (2018). Circuits Today. Available at: <http://www.circuitstoday.com/pll-phase-locked-loops>

Rouse, M. (2005). *debouncing.* Available at: <https://whatis.techtarget.com/definition/debouncing>

Williams, E. (2015). *EMBED WITH ELLIOT: DEBOUNCE YOUR NOISY BUTTONS, PART I.* Available at: <https://hackaday.com/2015/12/09/embed-with-elliot-debounce-your-noisy-buttons-part-i/>