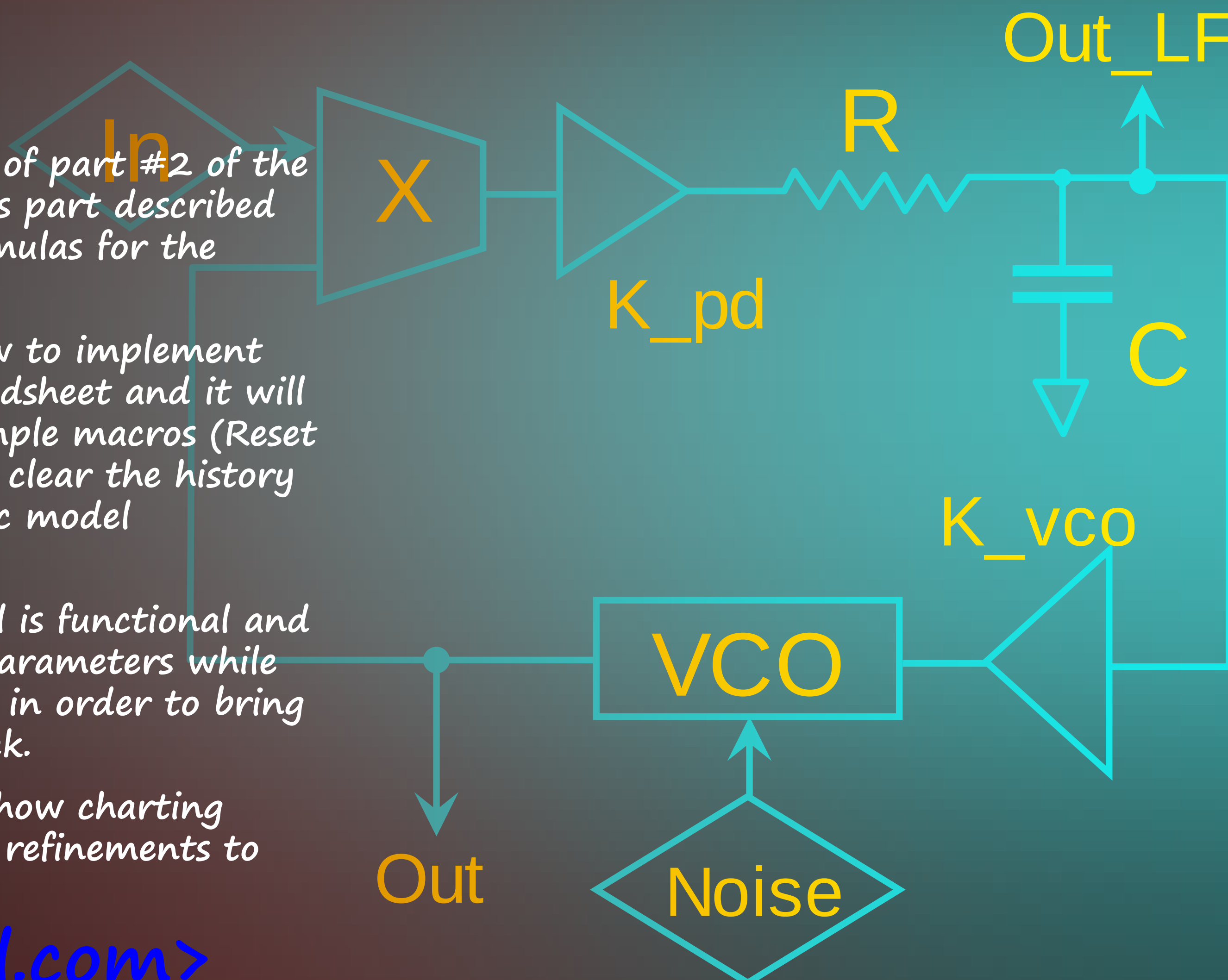
How to Model a Phase-Locked Loop (PLL) in Excel – part #3

-This is the continuation of part #2 of the PLL tutorial. The previous part described the derivation of the formulas for the numerical model.

- This part will show how to implement the formulas in the spreadsheet and it will also demonstrate two simple macros (Reset and Start_Pause) used to clear the history and animate the dynamic model respectively.

- At this point the model is functional and you can start changing parameters while the simulation is running in order to bring the loop in and out of lock.

- The next tutorial will show charting options and will add few refinements to the model.

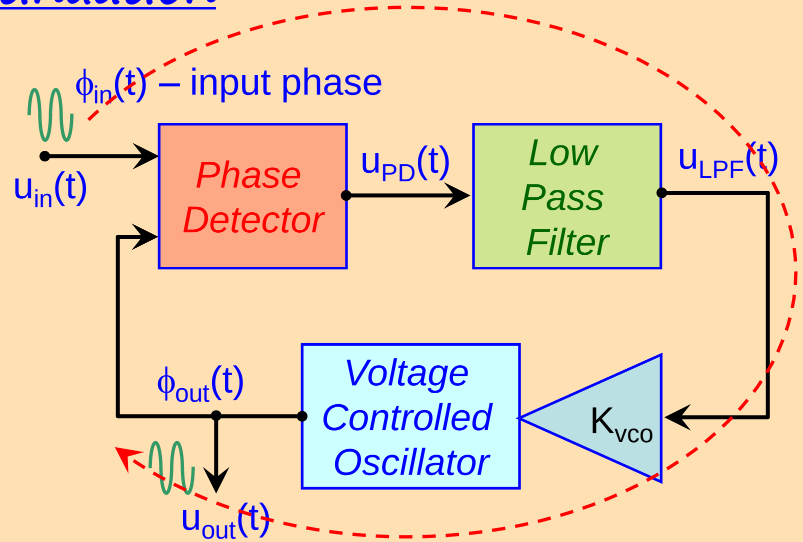


<excelunusual.com>

Implementing the calculation section - continuation

- We need to implement the previously derived formulas in a worksheet:

$$\varphi_{in}(n)=\varphi_{in}(n-1)+2\cdot\pi\cdot f_{in}(n)\cdot h$$
$$u_{in}(n)=\sin(\varphi_{in}(n))$$
$$u_{PD}(n)=u_{in}(n)\cdot u_{out}(n-1)$$
$$u_{LPF}(n)=\frac{h\cdot u_{PD}(n)+R\cdot C\cdot u_{LPF}(n-1)}{R\cdot C+h}$$
$$\varphi_{out}(n)=\varphi_{out}(n-1)+2\cdot\pi\cdot\left(f_{VCO\_free}+u_{LPF}(n)\cdot K_{VCO}\right)\cdot h$$
$$u_{out}(n)=\sin(\varphi_{out}(n))$$



- The input phase: A18: “=A19+2*PI()*B$11*B$1” $\Leftarrow \varphi_{in}(n)=\varphi_{in}(n-1)+2\cdot\pi\cdot f_{in}(n)\cdot h$

- The input voltage: B18: “=SIN(A18)” $\Leftarrow u_{in}(n)=\sin(\varphi_{in}(n))$

- The output PD voltage: C18: “=B18*F19” $\Leftarrow u_{PD}(n)=u_{in}(n)\cdot u_{out}(n-1)$

- The output LPF voltage: D18: “=(B$1*C18+B$5*D19)/(B$5+B$1)” $\Leftarrow u_{LPF}(n)=\frac{h\cdot u_{PD}(n)+R\cdot C\cdot u_{LPF}(n-1)}{R\cdot C+h}$

- The output phase: E18: “=E19+2*PI()*(B$3+B$7*D18)*B$1” $\Leftarrow \varphi_{out}(n)=\varphi_{out}(n-1)+2\cdot\pi\cdot\left(f_{VCO\_free}+u_{LPF}(n)\cdot K_{VCO}\right)\cdot h$

- The input voltage: F18: “=SIN(E18)” $\Leftarrow u_{out}(n)=\sin(\varphi_{out}(n))$

- After entering the above formulas you need to copy range A18:F18 down to row 37 and after that you will have 20 rows (20 time steps) of parallel (fast) calculations.
- 20 steps of static formulas will be combined with a macro which will generate a dynamic model based on a copy-paste loop. Each loop iteration will shift the simulation data 20 steps down (in the past) making room for new calculation data.

The PLL dynamic run macros:

-In addition to the macros associated to the input parameter buttons we also need “Start_Pause” and a “Reset_” macros and buttons.

- The “Reset” macro clears all the simulation history. All the initial conditions (after reset) are left to zero thought you could upgrade the “Reset” macro to insert different initial conditions.

- The Boolean variable “s” helps give the “Start_Pause” button a toggle effect. The value of this variable determines the conditional “Do” loop to either run (for s=True) of stop (for s=False) . Whenever the button is clicked, variable “s” will toggle and so will the macro operation mode.

- The “Start_Pause” macro, once started, will copy 5000 time steps of history information of the model and paste it 20 cells down (in the past). It will do this operation repeatedly in a loop creating continuously time rolling simulation environment in which only the last 5000 time steps of information are saved and can be plotted. The plot is be dynamic and you can see the effect of any change in input parameters on the waveforms.

- At this point the model is functional provided you plot some of the waveforms ($u_{LPF}(t)$ is my favorite). You can run it and while it runs change various input parameters (especially Fin). Try to bring the loop in and out of lock by adjusting Fin and watching u_{LPF}. Charting the loop voltages will be explained in the next tutorial.

```
Public s As Boolean

Sub Start_Pause()
s = Not (s)
    Do Until s = False
            [A38:F5038] = [A18:F5018].Value
        DoEvents
    Loop
End Sub

Sub Reset()
    [A38:F5038].ClearContents
End Sub
```