Behavioral Modeling and VHDL Simulation of an All-Digital Phase Locked Loop

Vikas Gaur[1] Mrs. Gayatri Sakya[2]
[2]Guide & Assistant Professor
[1, 2]Department of Electronics Engineering
[1, 2] JSS academy of technical education, Noida

Abstract---In this paper, we have designed a model of an all-digital phase-locked loop (ADPLL) which is discrete in nature. The design is carried out in simulink and then the code of the main blocks i.e. DDS and Hilbert is written and verified in VHDL. The basic components of the ADPLL are the phase detector, the loop filter and voltage controlled oscillator which is realized as the direct digital synthesizer (DDS).The phase detection is realized by using the Hilbert transform and then the phase is calculated using CORDIC algorithm. The linear range of the ADPLL is increased by using the phase-unwrap block. All equations to design this ADPLL are derived by using the relations of time continuous PLL. The time continuous model is simulated first by using matlab-simulink and all equations are analyzed. By using these equations all the relations to design an all-digital PLL are obtained.

I. Introduction

Due to the better predictability and higher accuracy more applications are shifting from analog domain to digital domain. As domain has certain advantages over analog domain such as in digital we can use more complexity.

In all digital PLL's we basically use two types of the phase detector named phase frequency detector (PFD) and time to digital converter (TDC).Phase frequency detector has large frequency range while time to digital converter has very fine resolution. Both of the techniques the phase detection methods can be combined to get the high frequency range of the pfd and better resolution of TDC. But PFD uses a filter due to which digital filtering is possible if large phase errors can be accepted and TDC requires constant delays, which requires manual measurement of the delays. Better predictability cannot be obtained by using the above methods.

To overcome such problems we can use hybrid PLLs where all parts of the PLL are digital but any one part is used as an analog but the main problem here is that it results in reduced flexibility because all the parts of the design are fixed.

In this paper a new PLL is designed where all parts of the PLL like phase detector, loop filter, direct digital synthesizer are digital and hence we gain better lock in range and predictability by using this model. The phase detector is designed by using the Hilbert transform which produces an analytic signal when a sinusoid is applied to its input. An analytic signal is one that does not contain any negative frequency component. The phase is then calculated by using the CORDIC algorithm. To increase the linear range a phase unwrap block has been used. In feedback to the ADPLL we are using the Hilbert transform and the CORDIC algorithm to generate an analytic signal from the output which is further used to generate the error signal. In this ADPLL we are using the PI controller as the loop filter because it has low pass behavior, and this property of the PI controller can be used to retard the higher frequency components.

II. Basics of PLL

The phase-locked loop concept was first developed in the 1930.It has since been used in communications systems, FM demodulators, stereo demodulators, tone detectors, frequency synthesizers etc. PLL's are widely used for clock synchronization in digital circuits and FPGA.

The basic blocks of PLL are the phase detector, loop filter, an amplifier and a voltage controlled oscillator.

The voltage-controlled oscillator is simply an oscillator whose frequency is proportional to an externally applied voltage. When the loop is locked on an incoming periodic signal, the VCO frequency is exactly equal to that of the incoming signal.

The phase detector produces a dc or low-frequency signal proportional to the phase difference between the incoming signal and the VCO output signal.

This phase-sensitive signal is then passed through the loop filter and amplifier and is applied to the control input of the VCO.

If, for example, the frequency of the incoming signal shifts slightly, the phase difference between the VCO signal and the incoming signal will begin to increase with time. This will change the control voltage on the VCO in such a way as to bring the VCO frequency back to the same value as the incoming signal.

Thus the loop can maintain lock when the input signal frequency changes and the VCO input voltage is proportional to the frequency of the incoming signal.

The range of input signal frequencies over which the loop can maintain lock is called the lock range.

An important aspect of PLL performance is the capture process, by which the loop goes from the unlocked, free-running condition to that of being locked on a signal.

In the unlocked condition, the VCO runs at the frequency corresponding to zero applied dc voltage at its control input. This frequency is called the centre frequency, or free-running frequency.

The phase detector is simply an analogue multiplier that multiplies the two sinusoids together. Thus the output of the multiplier-phase detector contains the sum and difference frequency components, We assume that the sum frequency component is sufficiently high in frequency that it is filtered out by the low-pass filter.

Once the system becomes locked, the difference frequency becomes zero and only a dc voltage remains at the loop-filter output.

The capture range of the loop is that range of input frequencies around the centre frequency onto which the loop will become locked from an unlocked condition and the pull-in time is the time required for the loop to capture the signal.

III. LINEAR PLL MODEL

A. *Continuous PLL Model:*

The designed ADPLL can be approximated by using the continuous PLL. Use of analog model to derive equations for discrete model is easier due to the availability of abundant literature on analog design and analysis of PLL. So we start with phase model of continuous PLL.

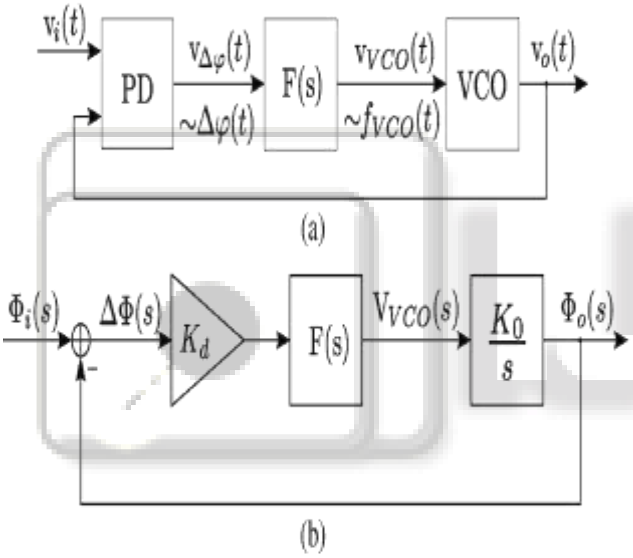

Fig.2 (a): Block diagram (b) phase model.

In fig the block diagram and phase model of the continuous PLL model is shown. All the equations are derived by using the above phase model. The transfer function of the phase error is given as:

$$E(s) = \Delta\Phi(s)/\Phi_i(s) = s / (s + K_dK_0F(s)) \qquad (1)$$

The type of the PLL is defined as the number of integrators in the loop. The designed ADPLL is of type-2 because it uses a PI controller as the loop filter. The transfer function of continuous PI controller is given as:

$$F_{PI}(s) = K_p + K_I/s \qquad (2)$$

The equation above is used in next section to derive transfer function of discrete PI controller.

Putting eqn2 in eqn1 we get:

$$E_{PI}(s) = s^2/(s^2 + 2\zeta\omega_n s + \omega_n^2) \qquad (3)$$

Where the natural frequency and damping factor is given by:

$$\omega_n = (K_dK_0K_I)^{1/2}$$

$$\zeta = (Kp/2)(K_dK_0/K_I)^{1/2}$$

the value of damping factor is taken to be around 0.707 and natural frequency determines the settling time and is a tradeoff between locking time and phase filtering property.

B. *Discrete PLL Model*

The discrete time model of ADPLL has been realized in simulink and the equations of error and transfer functions of the PI controller have been derived on the basis of equations discussed in previous section (eq. 1,2and 3).

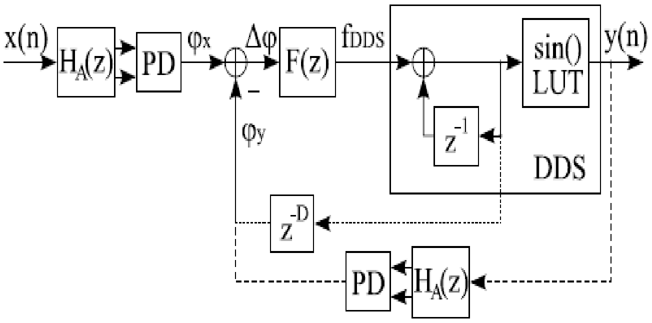

Fig. 2(a)

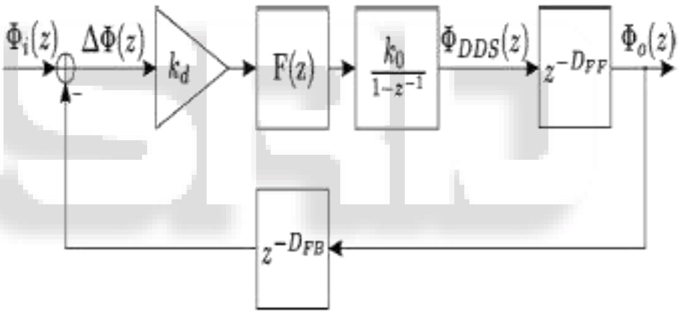

Fig. 2(b)

Fig.2 (a): block diagram(b) phase model.

The block diagram and the phase model of the ADPLL are shown in figure above. As compared to the continuous PLL model the loop filter is replaced by the PI controller and the VCO is replaced by the Direct Digital Synthesizer.

As we can see an analytic signal is generated from the applied input by using Hilbert transform block and then the phase is calculated by using the CORDIC algorithm. The error transfer function for ADPLL model is:

$$E(z) = (1-z^{-1}) / (1-z^{-1} + k_dk_0F(z)z^{-(D_{FB}+D_{FF})}) \qquad (4)$$

Where D_{FB} and D_{FF} are the delays of forward and feedback loop. The transfer function of the loop filter is derived using the equation (2) of continuous time model given in previous section:

Substituting S as $(1-z^{-1})/Ts$in eq.2results in an IIR filter with transfer function:

$$F_{PI}(z) = (b_0+b_1z^{-1})/(1-z^{-1}) \qquad (5)$$

Where $\mathbf{b_0 = K_p + K_i T_s}$

$\mathbf{b_1 = -K_p}$

The relation between coefficients of continuous and discrete model is obtained as:

$$\mathbf{b0 = (2\zeta + \omega_n Ts)\ \omega_n/(\ K_d K_0)}$$
$$\mathbf{b1 = 2\zeta\omega_n/(\ K_d K_0)}$$

C. Implementation and Results of the ADPLL Model

The complete model of the proposed ADPLL is shown in fig. , the blocks used in the design are described one by one in below in the subsections. For this model the input signal should be perfectly sinusoidal in nature. So we have taken a pure sine wave as an input of the ADPLL model which is fed to the Hilbert transform block.

1) HILBERT TRANSFORM

The input signal is converted into an analytic signal by using the Hilbert transform which generates in-phase and quadrature components of the signal. The designed simulink model of the block is shown below with results. For ADPLL model to reduce complexity a multiplier less Hilbert filter which can be found in [1] is used.

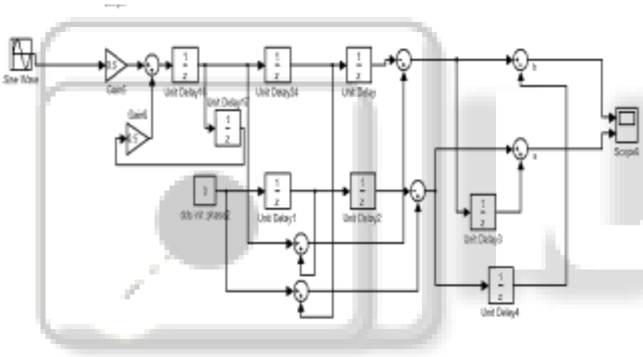

Fig. 3(a): Simulink model of Hilbert Transform

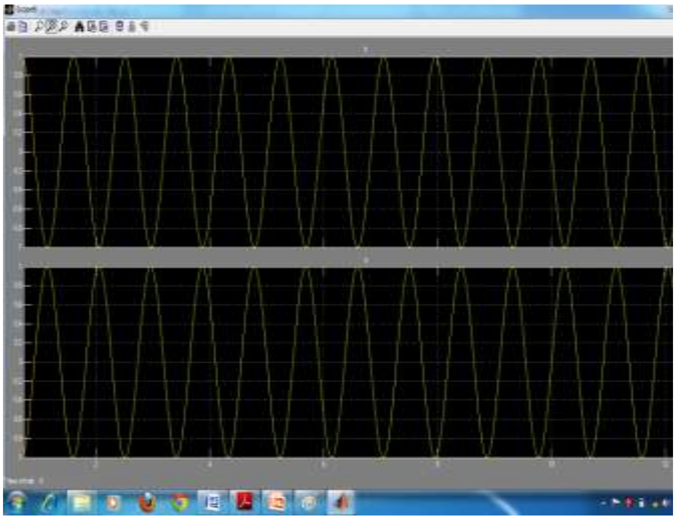

Fig. 3(b): Waveforms from scope at output

Since analytic signal has the form $X = a + iy$ where a is the delayed version of the input signal and y is the Hilbert transform of a. From this analytic signal the phase can be easily obtained by using the formula

Phi= y/x.

This formula can be easily detected by using the CORDIC block. The CORDIC algorithm was proposed by Volder in [2] is used.

2) Phase Unwrap Block:

The lock in range of the ADPLL model is increased by using the phase unwrap block. The simulink model of the PU block is shown below:

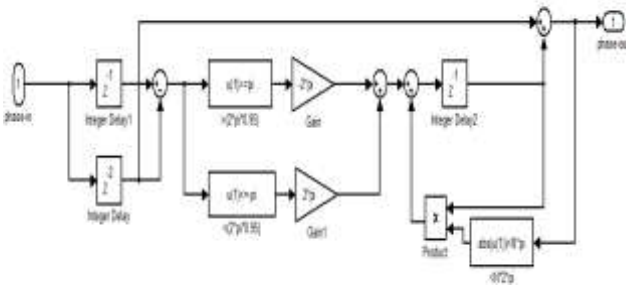

Fig. 4: Model of phase unwrap

3) Direct Digital Synthesizer:

The final output of the ADPLL model is obtained at the DDS output. The simulink model of the block with word length of 16 bit is shown below. And the output of DDS for a particular input is shown in fig 5 (b).The DDS proposed by V. F. Kroupa [3] which is based on a common LUT-based fashion is used.

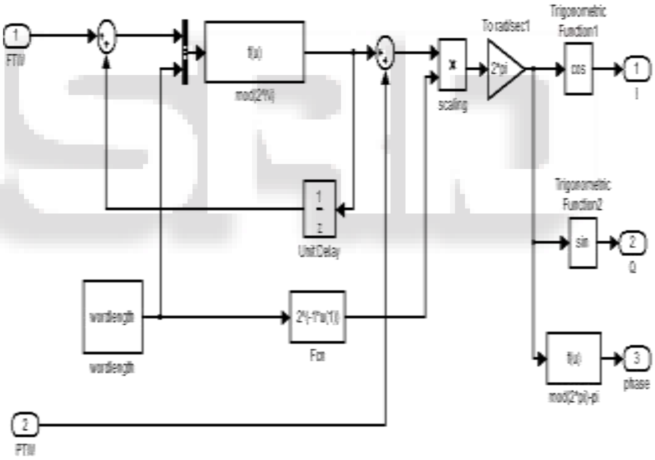

Fig. 5 (a): Model of DDS

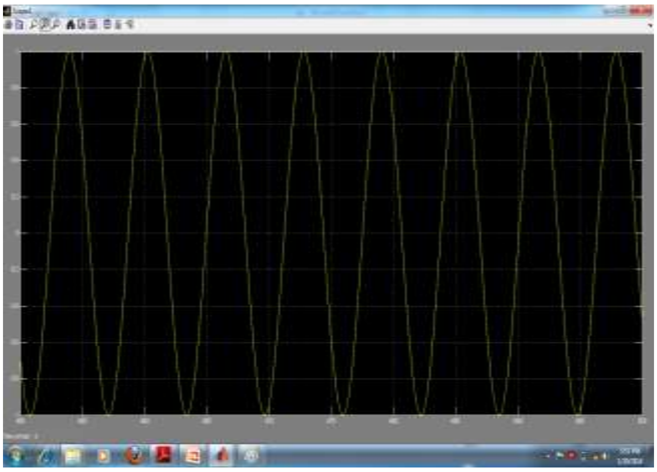

Fig. 5 (b): Output of DDS

4) *Final Output of Complete Model*

The final model was simulated at frequency of 10MHz the capture time of the design is 7.2x10^{-5}secondsthe simulation results are given in fig.6 with input and output waveform.

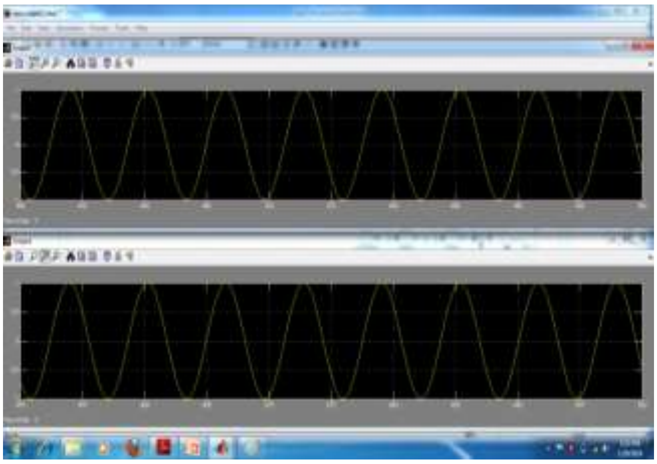

Fig. 6: Final output

5) *Calculations of the Pi Controller Coefficients*

All the calculations of the PI controller coefficients are given below in table. fn is set to 16 kHz and fs=120 MHz where Ts=1/fs.

Parameter	Calculated value
ω_n = 2*pi*fn	100.530 rad/sec.
k0_dds=(2*pi)/Ts	753.982*10^-6.
kd	1
Kp=(2*zeta*ω_n)/(k0*kd)	188.53*10^-6.
Ki=(ω_n)^2 /(k0*kd)	13.404
b1=(-Kp)	-188.53*10^-6
b0=(Kp+Ki)*Ts	188.65*10^-6.

6) *THE ERROR SIGNAL PLOT*

The plot of the error signal is shown below. It can be seen that the error signal tends to zero within no time which means that the efficiency and stability of the ADPLL model is very high. The phase error reduces to zero within 1.3 seconds.

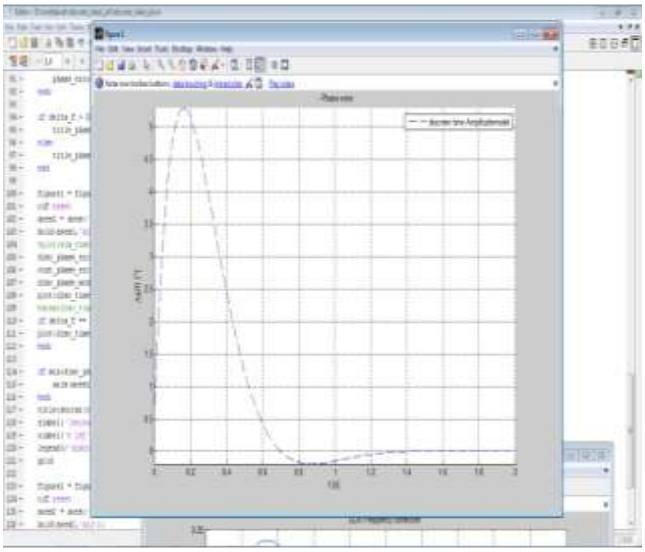

Fig. 7: Error v/s time plot of ADPLL model

7) *VHDL Implementation*

VHDL implementation of DDS and Hilbert transform is carried out in modelsim. The DDS designed for the ADPLL is based on 8-bit LUT and the results of the RTL designs of the blocks are shown below.

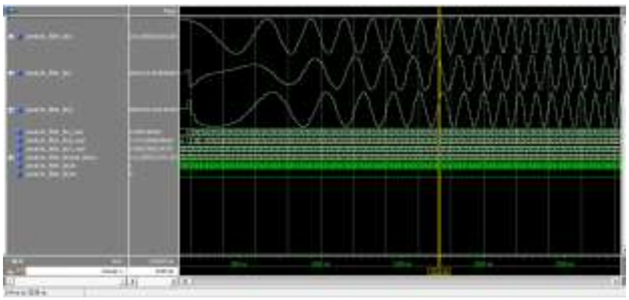

Fig. 8 (a) Simulation of Hilbert core

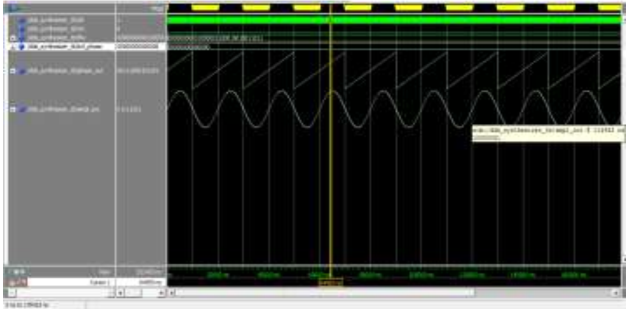

Fig. 8 (b) Simulation of DDS

IV. CONCLUSION

In ADPLL model detailed analysis has been done of the phase detection system which consists of the Hilbert transform and the CORDIC algorithm. If compared with the conventional PLL models with phase detection techniques like the PFDs and TDCs this method of phase detection has certain advantages like easy calibration, better programmability, and high phase accuracy with reduced phase error. The ADPLL designed has better flexibility than the analog PLL models.

REFERENCES

[1] M. Kumm and M. S. Sanjari, “Digital Hilbert transformers for FPGA based phase-locked loops,” in Proc. Int. Conf. FPL, 2008, pp. 251–256.

[2] J. E. Volder, “The CORDIC trigonometric computing technique,” IRE Trans. Electron. Comput, vol. 8, no. 3, pp. 330–334, Sep. 1959.

[3] F. Kroupa, Direct Digital Frequency Synthesizers. Piscataway, NJ: IEEE Press, 1998.

[4] F. Kroupa, Phase Lock Loops. New York: Wiley, 2003.

[5] F. M. Gardner, Phase locks Techniques, 3rd Ed. New York: Wiley, 2005.

[6] J. Wilson, A. Nelson, and B. Farhang- Boroujeny, “Parameter derivation of type-2 discrete-time phase-locked loops containing feedback delays,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 56, no. 12, pp.886–890, Dec. 2009.

[7] R. E. Best, Phase-locked loops: theory, design, and applications, 2nd ed. NY: McGraw-Hill, 1993.

[8] D. H. Wolaver, Phase-locked loop circuit design. NJ: Prentice Hall, 1991.

[9] B. Lathi, Linear systems and signals. Oxford University Press, USA, 2002.