VHDL-AMS Modeling of a New PLL with an Inverse Sine Phase Detector (ISPD PLL)

Mohamed KARRAY* , Jong Kug SEON* , Jean-Jacques CHARLOT*
and Nouri MASMOUDI**
* Ecole Nationale Supérieure des Télécommunications
Département COMELEC
46, rue Barrault, 75013 Paris, France
** Ecole Nationale des Ingénieurs de Sfax
3038 Sfax, Tunisie

karray@enst.fr

Keywords
Behavioral modeling, VHDL-AMS, PLL, ADVanceMS.

Abstract
For improving the performance of PLL, a $\sin^{-1}(x)$ *function generator has been substituted for a standard phase detector. The behavior of blocks have been modeled with VHDL-AMS at different abstraction levels and compared with experimental results. We point out that connections between that blocks may need some (impedance) adaptation and physical accurate models may not be absolutely necessary for an accurate simulation of the loop.*

1. Introduction

The Phase Lock Loop (PLL) is essentially a negative feedback control system composed of phase detector, filter, and voltage controlled oscillator (VCO). The properties of the Phase Detector (PD) circuit have strong influence on the dynamic performance of the PLL system [1]. The lock range of the PLL system is also dependent on the loop bandwidth and the design of the PD. In Analog PLLs, the PD has almost always been implemented using a balanced modulator (a four-quadrant multiplier). The acquisition properties of the PLL system, however, are severely limited by the inherent characteristics of the transfer function of the multiplier-type PD such as non-linearity, input dynamic range, and narrow bandwidth.
This paper recalls the mathematical modeling described in [2] and presents the VHDL-AMS modeling of the proposed ISPD-PLL and comparison with experience.

2. ISPD PLL Description

A new phase detector for improving the performance of PLL system by substituting a $\sin^{-1}(x)$ function generator for a standard phase detector has been introduced [2]. The proposed Inverse Sine PD (ISPD) can directly convert the phase difference of two input signals into the linear output voltage (Figure 1).

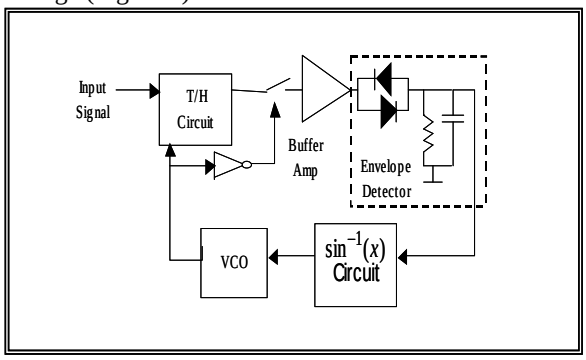

Figure 1 : The proposed PLL with Inverse Sine PD.

3. Mathematical Model

We propose now the mathematical model of ISPD PLL.
Let us consider that the input signal is defined as:

$$Vin(t) = A\sin\alpha(t) = A\sin(\omega_0 t + \theta_i(t)) \quad (1)$$

$$\theta_i(t) = (\omega_i(t) - \omega_0)t + \psi_i(t) \quad (2)$$

In discrete time the preceding equation is written:

$$\alpha(k) = \omega_0 t(k) + \theta_i(k) \quad (3)$$

$$\alpha(k) = \omega_0 \sum_{i=1}^{k} t(i)\theta_i(k) \quad (4)$$

When : $t(k) = T(0) + T(1) + \ldots\ldots T(k)$ For k>0

The output of the ISPD can be expressed then by:

$$\varepsilon_r(k) = K_{is}[\sum_{i=1}^{k}(\omega(i) - \omega_0)T(i) + \theta_i(k) - \frac{\pi}{2}] \quad (5)$$

This enables us to extract a general equation from ISPD PLL [2].

$$\theta_o(k+1) - \theta_o(k) = 2\pi\frac{K_{vco}\varepsilon_r(k)}{f_0 + K_{vco}\varepsilon_r(k)} = 2\pi\frac{K_i\varphi(k)}{f_0 + K_i\varphi(k)} \quad (6)$$

where φ indicates the difference between the input signal and VCO one; what enables us to present the general system as follows:

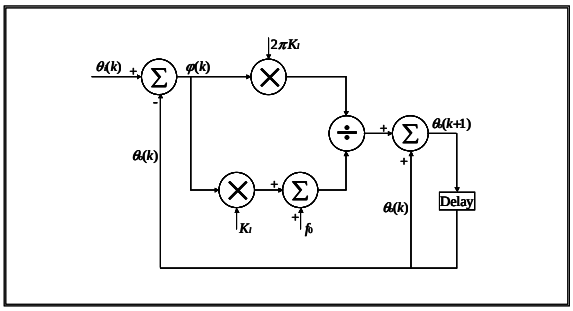

Figure 2 : ISPD PLL block diagram.

3.1 Phase step response

A step of phase input is used to verify how the ISPD PLL is affected by the initial phase error $\varphi(0)$. If the input phase $\theta_i(t) = \theta = cste$, the phase error can be obtained as:

$$\varphi(k) = \theta_i(k) - \theta_o(k) = \theta - \theta_o(k)$$

And $$\varphi(k+1) = \varphi(k) - 2\pi \frac{K_l \varphi(k)}{f_0 + K_l \varphi(k)} \quad (7)$$

Where $K_l = K_{is} * K_{vco}$.

In the steady state both $\varphi(k+1)$ and $\varphi(k)$ equal φ_s. For large enough *k*, the phase error is small enough, the loop equation can be approximated by:

$$\varphi(k+1) = \varphi(k) - 2\pi \frac{K_l \varphi(k)}{f_0} = \varphi(k)(1 - K_s) \quad (8)$$

where $K_s = 2\pi \frac{K_l}{f_0}$.

In is seen from this approximation that the condition for the steady state to exists $|1 - K_s| < 1$. The loop gain should satisfy the condition: $0 < K_l < \frac{f_0}{\pi}$.

3.2 The response for a ramp of frequency

The response is very important, since most of dynamic characteristics of the PLL system can be determined.

The input signal is represented as: $\alpha(t) = \omega_0 t + \theta_i(t)$, where $\theta_i(t) = (\omega_i - \omega_0)t + \psi_i$.

We can finally obtain the equation of the phase error [2]:

$$\varphi(k+1) = \varphi(k) - \frac{\omega}{\omega_0}\left[\frac{2\pi K_l \varphi(k)}{f_0 + K_l \varphi(k)}\right] + 2\pi \frac{\omega - \omega_0}{\omega_0} \quad (10)$$

In the steady state both $\varphi(k+1)$ and $\varphi(k)$ equal φ_s, and it gives:

$$\frac{K_l \varphi_s}{f_0 + K_l \varphi_s} = \frac{\omega - \omega_0}{\omega}$$

For $\varphi_s = -\frac{\pi}{2}$ $\quad \omega_b = \omega_0 \frac{f_0 - K_l \frac{\pi}{2}}{f_0}$: lower lock range frequency.

For $\varphi_s = \frac{\pi}{2}$ $\quad \omega_h = \omega_0 \frac{f_0 + K_l \frac{\pi}{2}}{f_0}$: upper lock range frequency.

The steady state error of the loop for a ramp of phase is: $\varphi_s = \frac{f - f_0}{K_l}$. And the phase error converge into the loock region if *w* lies within the range:

$$\frac{f_0 - K_l \frac{\pi}{2}}{f_0} < \frac{\omega}{\omega_0} < \frac{f_0 + K_l \frac{\pi}{2}}{f_0}.$$

4. VHDL-AMS Modeling

The Phase Detector is the most important block of the proposed PLL, it uses a $sin^{-1}(x)$ generator and a Track/Hold (T/H) block.

The inverse sine function to be approximated, $y(x)$, is an odd function with the normalized angle x. Therefore, as starting point, an odd polynomial is considered and suitable constraints for forcing perfect fit are $y(\pm 1.0) = \pm\pi/2$, $y'(0) = 1$, and $y'(\pm 1.0) = \infty$. To satisfy these three conditions, only the first three terms of an odd polynomial are required and three coefficients, a, b, and c, are selected. An implicit term is also introduced to add a free coefficient, d [2].

The constraints of $y(\pm 1.0)$ and $y'(\pm 1.0)$, when considered together, preclude the choice of a rational approximating functions. One of the simplest forms containing four coefficients that can be chosen is the following odd function:

$$y(x) = sin^{-1}(x) \cong \frac{ax + bx\sqrt{c - x^2}}{1 + dx^2} \quad (11)$$

The suitable constraints are applied to the odd function (11) and three of the coefficients are expressed in terms of one remaining coefficient, say a:

$$sin^{-1}(x) \cong \frac{x\left[a\frac{\pi}{2} + (1 - a\frac{\pi}{2})\sqrt{1 - x^2}\right]}{1 + (a-1)x^2} \quad (12)$$

When the error curves are calculated for different values of a, it is found that $a = 1.64$ yields near-minimax absolute error. The optimized approximation is then:

$$Sin^{-1}(x) \cong \frac{2.01 \cdot x - 1.23 \cdot x \cdot \sqrt{1 - x^2}}{0.78 + 0.5 \cdot x^2}, \quad -1 \le x \le 1 \quad (13)$$

The absolute approximation error between $sin^{-1}(x)$ and the approximation equation (13) is shown in Figure 3, and the $\sin^{-1}(x)$ VHDL-AMS model is shown in Figure 4.

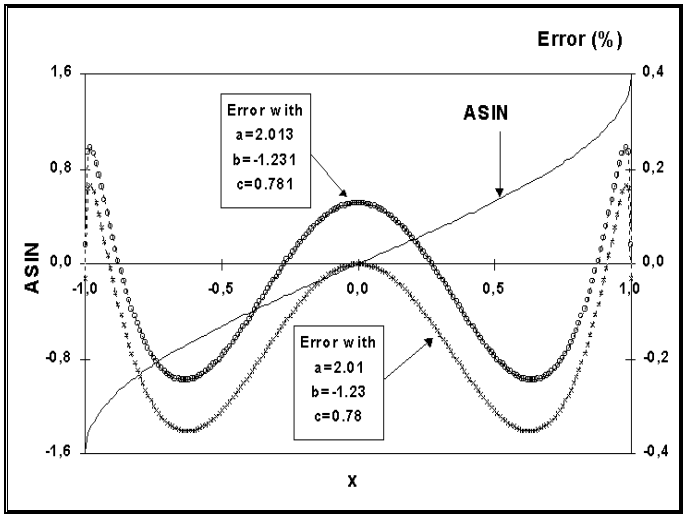

Figure 3 : The absolute errors and simulation result of inverse sine approximation.

The T/H block gives the instantaneous value of the input : it follows the input until the clock reaches high level and holds this value when the clock is at the low level.
Figure 5 shows the VHDL-AMS model of T/H block.
The VCO block converts the output of the $\sin^{-1}(x)$ block into a frequency needed by the clock. Figure 6 shows the VHDL-AMS model of VCO block.
Figure 7 presents some ADVanceMS simulation results. These results are in a good agreement with experiments and SPICE transistor level simulations presented in [2] and reported in figures 8 and 9.
Simulating a complex system with physical models gives accurate results but consumes an important CPU time. Using higher abstraction level models allows a faster simulation, but may give less precise results. A trade-off is needed between fastness and accuracy. The best is to combine different abstraction levels that is possible with VHDL-AMS. Nevertheless connection between blocs modeled with different abstraction levels may induce some problems because some models may take into account conservation laws but the others don't.
In a first step, all blocs of the PLL were modeled with behavioral models. In a second step, the behavioral inverter is replaced by a CMOS (physical) inverter. The interconnection between the inverter and the T/H block is made through a resistor. The resistor value has to be fitted in order to adapt impedances (high impedance for the inverter and infinite impedance for the T/H block). Simulation of second step with ADVanceMS gives the same results than in the first one. But in the same operating conditions, step one CPU time is about 1 minute, step two, 15 minutes. This means that it is not necessary to use a physical model with that condition.

```
ENTITY ARCSIN IS
GENERIC ( kis    : real;
          f0     : real );
PORT ( TERMINAL INP, OUTP, m : electrical);
END ENTITY ARCSIN;
ARCHITECTURE behavior_RCSIN OF ARCSIN IS
--Fonction d'approximation de Arcsinus
FUNCTION ARCSIN_fun (ve :real) return REAL IS
VARIABLE vs :real;
BEGIN
vs:=((2.01*ve-1.23*ve* (1.0-(ve**2.0))**0.5)/(0.78+0.5*(ve**2.0)));
return vs;
END FUNCTION;
QUANTITY ve across iin through INP    to m;
QUANTITY vs across             OUTP   to m;
BEGIN
vs == kis * ( ARCSIN_fun ( ve ) – ARCSIN_fun (1.0) );
END USE ;
END ARCHITECTURE behavior_ARCSIN;
```

Figure 4: VHDL-AMS model of $\sin^{-1}(x)$ generator.

```
ENTITY TH IS
GENERIC ( high   : real );
PORT ( TERMINAL INP, OUTP, CLK, m : electrical);
END ENTITY TH;
ARCHITECTURE behavior_TH OF TH IS
QUANTITY Vin across iin through INP    to m;
QUANTITY Vth across            OUTP    to m;
QUANTITY Vck across            CLK     to m;
BEGIN
IF (Vck'ABOVE ( HIGH) ) USE
  Vth == Vin;
 ELSE
  Vth == Vth;
END USE;
END ARCHITECTURE behavior_TH;
```

Figure 5 : VHDL-AMS model of T/H block.

```
ARCHITECTURE behavior_VCOP OF VCOP IS
   QUANTITY vout  across iout through  OUTP    to m;
   QUANTITY vin   across iin  through  INP     to m;
   QUANTITY  f_tmp , per       : real;
   SIGNAL   pulse_tran   : real := 0.0 ;
  BEGIN
   f_tmp == f0 - kvco * vin;
  IF    f_tmp <= Fmin  USE        per == 1.0 / Fmin ;
  ELSIF f_tmp >= Fmax  USE        per == 1.0 / Fmax ;
                       ELSE       per == 1.0 / f_tmp;
  END USE;
END USE;
vout == pulse_tran'ramp(tr,tf);
  p1 : PROCESS
  BEGIN
   WAIT UNTIL domain = time_domain;
   Break;
     LOOP
      pulse_tran <= high;
      WAIT FOR ((per+tr-tf)/4.0);
      pulse_tran <= low;
      WAIT FOR ((per-tr+tf)/2.0);
      pulse_tran <=high;
      WAIT FOR ((per+tr-tf)/4.0);
    END LOOP;
  END PROCESS;
END ARCHITECTURE behavior_VCOP;
```

Figure 6: VHDL-AMS model of VCO block.

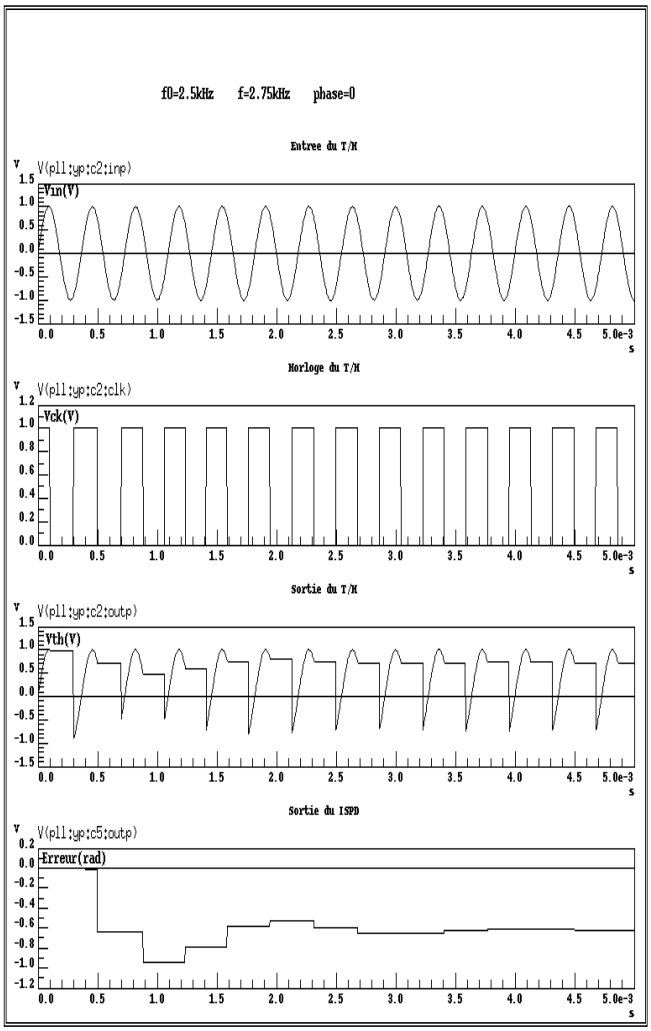

Figure 7: ISPD PLL simulation results.

5. Experimental Results of ISPD PLL

The measured results of input and output signal waves of each block at the sampling instants are shown in Figure 8. The loop of the PLL was locked on 500 kHz sine wave.

The tracking and holding periods of THA are coincided with each half cycle of periods of the VCO, and the clock signals of the VCO with the periods are controlled by the output of the Inverse Sine functional circuit. The frequency of the VCO will be increased or decreased to diminish the phase error. This behavior of the loop will be repeated until the frequency of the VCO equals that of the input signal [2].

Figure 9 shows how the ISPD and VCO outputs change with a 2 MHz input frequency, and the loop of ISPD PLL is locked. Initially, the VCO is at free-running frequency of 2.1 MHz. When the input frequency 2 MHz applies to the input of loop, approximately 2 µs after the initial frequency change, the input voltage of VCO settles at about 1.3 V.

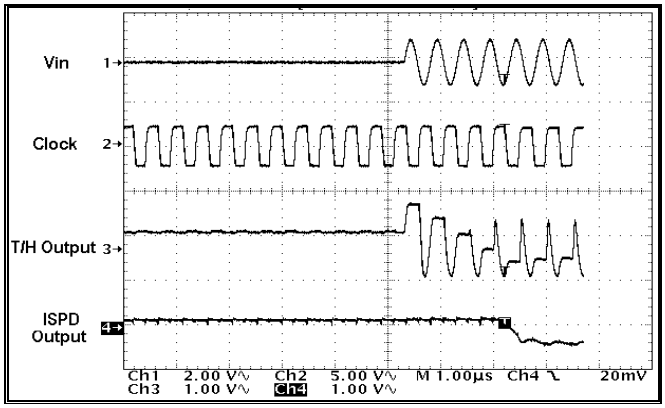

Figure 8 : The measured results of input and output signal waves of each block at the sampling instants.[2]

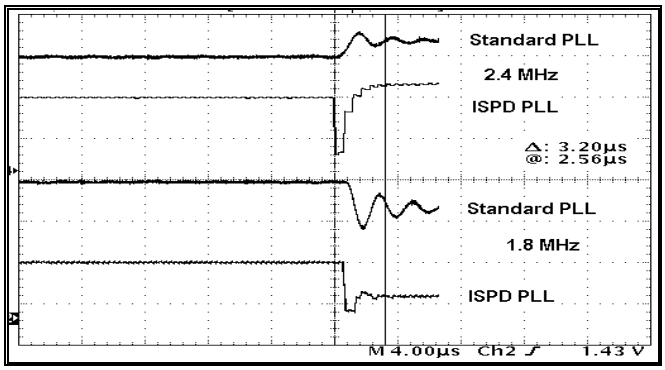

Figure 9 : The measured locking behavior of the loop.[2]

6. Conclusion

The structure and the mathematical model of this new PLL are different from those of the traditional PLL. The ISPD (Inverse Sine Phase detector) clearly improves the performances of the traditional PLL with filter and the generator of Inverse Sine can be used for other applications [3]. Its modeling is presented there with VHDL-AMS and simulation results shows a good agreement with experimental ones. In addition, a discussion has been leaded about the relevance of using different modeling abstraction levels and about the need of adaptation impedance between blocks.

References

[1] B. Razavi, " A2-GHz 1.6mW Phase-Locked Loop" IEEE J. of Solid-Sate Circuits, vol.32, N°5, Mai 19997.

[2] J.K. Seon, "*Analysis, Design and Realization of a New PLL Circuit using an Inverse Sine Phase Detectors*", (in French) PhD dissertation ENST-Paris, 2000.

[3] J.K. Seon et J.J Charlot, French Patent (FR0000786) "*Electronic Circuit for generate trigonometric signal , applications of a such circuit and operating procedure*", (in French) January 2000.