A Novel SAR Fast-Locking Digital PLL: 
Behavioral Modeling and Simulations 
Using VHDL-AMS

Dr. Mahmoud Fawzy Wagdy and Anurag Nannaka

Abstract - A novel successive-approximation fast-locking digital phase-locked loop (SAR DPLL) is presented and behaviorally modeled using VHDL-AMS. The DPLL operation includes two stages: (1) a novel coarse-tuning stage for frequency tracking which employs a successive-approximation algorithm similar to the one employed in SAR A/D converters (ADCs) and (2) a fine-tuning stage for phase tracking which is similar to conventional DPLLs. The coarse-tuning stage includes a frequency comparator, a successive-approximation register, a D/A converter (DAC), and control logic. Design considerations and implementation are presented in this paper. VHDL-AMS and Ansoft Simplorer are used to design and perform simulations. The fast-locking DPLL saves about 50% of the lock time as compared to its conventional DPLL counterpart.

Index Terms – PLL, DPLL, fast-locking, coarse tuning, frequency tracking, SAR algorithm, fine tuning, phase tracking, lock time, VHDL-AMS.

I. INTRODUCTION

PLLs are most common in applications like wireless transceivers, cellular phones, global positioning systems, etc. The PLL lock time is an important characteristic as it is the time a PLL takes to adapt to changes in the input frequency. Conventional PLLs employ phase tracking which takes a long time to lock, so they are misfits for contemporary high-speed high-throughput applications. Fast locking is therefore a necessity for clock/data recovery circuits, frequency-hopping spread-spectrum communications, cellular phones, etc.

The literature on conventional (classical) PLLs contains several thousands of designs and research papers, e.g. [1]. However the literature on fast-locking PLLs is relatively very limited due to their more recent demand. Example research papers include: [2-3]. Examples of US patents include: [4-5]. A number of industrial corporations produce fast-locking PLLs, e.g. Analog Devices Inc. [6], True Circuits Inc. [7], and National Semiconductor Inc. [8].

II. THEORY OF OPERATION

The block diagram of the successive-approximation DPLL is shown in the Fig. 1. The SAR DPLL consists of a phase frequency detector (PFD), a charge pump (CP), a lowpass filter (LPF), and a voltage-controlled oscillator (VCO) for fine tuning along with a frequency comparator (FC), a successive-approximation register (SAR), control logic (CL), and a Digital-to-Analog Converter (DAC) that are added for coarse tuning.

As per Fig. 1, the LPF input is selected using CL. In one switch position, the DAC output voltage is applied to the LPF to accomplish coarse tuning which involves frequency tracking, and in the other switch position, the SAR DPLL behaves like a conventional DPLL and performs phase tracking. A “timeout” signal from the FC or an “Endof5cycles” signal from the SAR triggers a monostable multivibrator which controls the switch.

When an input frequency hop occurs, the FC compares the input frequency with the VCO frequency and, depending upon the outcome, the SAR changes the voltage applied to the LPF and consequently the VCO via the DAC. When the input frequency F_input and the VCO output frequency F_vco become close to each other, coarse tuning is suspended and fine tuning is started using the control block.

III. DESIGN OF THE SAR FAST-LOCKING DPLL

The individual blocks of Fig. 1 are explained in this section. Behavioral modeling for each block is also given below.

A. Phase Frequency Detector (PFD)

The block diagram of the PFD is given in [9]. The inputs for the PFD are the input clock and VCO output. The output UP/DOWN signals depend upon the lead/lag relationship between the two input signals. Ultimately, when the DPLL is locked both the UP and DOWN signals remain low. Fig. 2 describes a part of PFD implementation.

B. Charge Pump (CP) and Lowpass Filter (LPF)

The outputs of the PFD (UP and DOWN pulses) are
combined into single output for driving the LPF using a CP. The block diagram for the CP and LPF is given in [9]. The CP sources current into the LPF during the UP pulse to increase the output frequency and sinks current during the DOWN pulse to decrease the output frequency. The CP current Ipump is chosen to be 400µA and -400µA.

The LPF is described in Fig. 3 and consists of R2, C2, and C1. The voltage across C2 increases/decreases due to the PFD UP/DOWN pulses. The LPF components are given by: R2= 300Ω, C2= 10pF, C1= 1pF.

C. Voltage-Controlled Oscillator (VCO)

The VCO produces a digital pulse train whose frequency is proportional to the voltage across capacitor C1 of the LPF. The VCO block diagram is given in [9] and the behavioral implementation is shown in Fig. 4. The VCO operating frequency is assumed to be from 80MHz to 2GHz.

D. Frequency Comparator (FC)

The FC is the most essential component of the Successive-Approximation DPLL and its block diagram is given in [10]. The FC has three inputs (F_input, F_vco, Reset). “F_input” is the input frequency and “F_vco” is the VCO output frequency. The advantage of this type of FCs, which uses ring counters, is that it always gives correct comparison results regardless of the phase shift between the F_input and F_vco signals. The counters are started and refreshed by using the “Reset” signal, which receives a pulse every 15ns thus starting a new frequency comparison cycle.

The frequency comparator has three outputs (High, Low, and Timeout). Output High =’1’ when F_input > F_vco, thus the voltage to the VCO must be increased so that F_vco catches up with F_input. Similarly, output Low=’1’ when F_input < F_vco, thus the voltage to the VCO must be
decreased so that F_vco catches up with F_input.

In cases when F_input and F_vco are close to each other the FC cannot decide during the allotted time (15ns in this case). To account for these cases, a Time-Out signal is added to the output of the FC. The idea is that ‘Time-Out = 1’ means that both F_input and F_vco are too close to each other, accordingly coarse tuning can be stopped and fine tuning can be started. The FC behavioral implementation is described in Fig. 5.

**E. Successive-Approximation Register (SAR)**

The SAR takes the outputs of the FC and selects an appropriate digital input for the DAC. The SAR can be represented by a state diagram as shown in Fig. 6. A 4-bit DAC is used in this design, so there are 16 states in the SAR; each state represents a corresponding voltage for the DAC. Thus the states 0001 through 1111 are equispaced in the range -2.66V through 2.66V in steps of 0.38V. At the beginning of the coarse tuning, the VCO output frequency (F_vco) must be at 1GHz, which is the center frequency of this SAR DPLL, and so at the beginning of coarse tuning, the output state should always be first state (state="1000"). Once coarse tuning completes and fine tuning starts, the state must be brought back to first state for the next coarse tuning.

Although successive approximation is done in 4 clock cycles (15ns each in this case), an additional clock cycle is needed at the beginning since the frequency hop can occur at any time during the first clock cycle.

After 5 clock cycles, the input and VCO frequencies may be very close and the FC will give the timeout signal. But if the FC gives a decision after 5 clock cycles as to which frequency is larger, an “endof5cycles” signal is enabled which gives the monostable multivibrator. The “timeout” and “endof5cycles” are the OR gate inputs. The OR gate output is called “trigger”. This “trigger” signal cannot be directly used to control the switch because while fine tuning in progress, the FC would still continue comparing and the VCO output changes continuously as a function of the voltage across C1 of the LPF, which may generate false outputs at the FC. For example, while fine tuning is in progress the FC may give a false low signal which pulls the “timeout” signal, which could supposedly be ‘1’, back to ‘0’, which results in pulling the switch back to coarse tuning. To prevent this condition, a monostable multivibrator is used. Instead of applying the “trigger” directly to the control switch, it is applied to the monostable multivibrator. The monostable multivibrator produces an output pulse with fixed duration called “oneshotoutputpulse”, which is used to control the switch. The period of the “oneshotoutput pulse” is chosen to be 75ns.

**F. Digital-to-Analog Converter (DAC)**

The DAC outputs a voltage in the range -2.66V through 2.66V to produce a specific frequency from the VCO in the range 200MHz through 1.81 GHz in steps of 115MHz.

**IV. IMPLEMENTATION OF THE SAR FAST-LOCKING DPLL**

**A. Conventional (Classical) DPLL**

The classical DPLL is composed only of a PFD, CP, LPF,
and a VCO. The DPLL lock time is calculated by performing several positive and negative frequency hops. The DPLL is said to be locked when the control voltage to the VCO becomes constant and ripple-free while the phase difference between F_input and F_vco becomes zero or constant.

B. Fast-Locking SAR DPLL

The SAR DPLL is shown in Fig. 1. This SAR DPLL is the counterpart of the flash fast-locking DPLL in [11].

Coarse tuning starts by selecting the SAR output state as “1000”, i.e. 1GHz at the VCO output. The FC then compares F_input with 1GHz first then proceeds with the rest of the SAR algorithm. Thus, the output frequency successively moves closer to the input frequency. A 4-bit DAC is used so at the most 5 cycles (including one additional clock cycle) of 15ns each are needed to reach the closest frequency to the input frequency. Once both frequencies are very close, the “Timeout” signal from the FC or “Endof5cycles” signal from the SAR will cause the control logic connected to the switch to stop coarse tuning and start fine tuning.

Fig. 7 is a plot of the VCO control voltage (Vin) versus time. For the example frequency hop, coarse tuning starts at 500ns and after 75ns (i.e. at 575ns), fine tuning starts and ends at 610ns. The DPLL is said to be locked at 610ns and the lock time is thus 110ns for a frequency hop of 1.4GHz.

C. Comparison Table

Table 1 compares the lock times of the SAR Fast-Locking DPLL and its conventional (classical) DPLL counterpart for various positive and negative frequency hops. The table reveals that lock times are no more than 150ns.

V. CONCLUSIONS

A novel successive-approximation fast-locking DPLL is presented and behaviorally modeled. It employs two stages: a novel coarse-tuning stage for frequency tracking, which reduces the lock time, followed by a fine-tuning stage for phase tracking similar to the conventional (classical) DPLL. The SAR DPLL additional hardware includes a frequency comparator, a SAR, a DAC, control logic, and a switch.

Coarse tuning might take 60–75ns (depending on the input frequency) then shifts control to fine tuning at the end of which the DPLL is completely locked. Simulations revealed that for various frequency hops the lock time is reduced by about 50% on the average.

The SAR DPLL needs to lock completely before hopping to the next frequency because of the one-shot fixed-duration output pulse controlling the switch. So, a minimum of 150ns is needed between two input frequency hops.

<table>
<thead>
<tr>
<th>Frequency Hop</th>
<th>Fast Locking SAR DPLL (in nanoseconds)</th>
<th>Conventional DPLL (in nanoseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>400MHz to 900MHz</td>
<td>150</td>
<td>240</td>
</tr>
<tr>
<td>900MHz to 1.6GHz</td>
<td>150</td>
<td>280</td>
</tr>
<tr>
<td>800MHz to 1.8GHz</td>
<td>160</td>
<td>250</td>
</tr>
<tr>
<td>450MHz to 1.75GHz</td>
<td>140</td>
<td>300</td>
</tr>
<tr>
<td>900MHz to 400MHz</td>
<td>130</td>
<td>360</td>
</tr>
<tr>
<td>1.5GHz to 800MHz</td>
<td>130</td>
<td>360</td>
</tr>
<tr>
<td>1.75GHz to 450MHz</td>
<td>130</td>
<td>360</td>
</tr>
</tbody>
</table>

VI. REFERENCES