Fast Automatic Sizing of a Charge Pump Phase-Locked Loop based on Behavioral Models

Jun Zou[1,2],Daniel Mueller[1],Helmut Graeb[1],Ulf Schlichtmann[1],Eckhard Hennig[2],Ralf Sommer[2]

1. Technische Universitaet Muenchen 2. Infineon Technologies AG
Munich, Germany

ABSTRACT

In this paper, we present an analog hierarchical sizing methodology applied to a third-order charge pump phase-locked loop (CPPLL). The key idea is to propagate the specifications from the requirements of the behavioral level to the circuit level. At the behavioral level, the performance is optimized while considering the potential capacity of the underlying circuits. Critical advantage of the illustrated methodology is a shortened PLL sizing process due to the use of fast-simulating models at behavioral level. The simulation results show the availability of this method on the CPPLL which makes an automatic sizing process actually feasible in terms of computation time.

Keywords

Hierarchical Sizing, Top-Down, Bottom-Up, Behavioral Modeling, Locking Time, Phase Margin.

1. INTRODUCTION

Phase-Locked Loops (PLLs) play important roles in many applications ranging from frequency synthesis to clock recovery in wireless receivers. As the competition in electronic market becomes stronger and stronger, the design of PLLs becomes a crucial part of the time-to-market for many products. Circuit level simulation of PLLs is quite time-consuming. Therefore a hierarchical sizing methodology is needed to accelerate the design process for PLLs.

1.1 Analog Hierarchical Design

To design a complex mixed-signal block such as a PLL, the block is typically decomposed into smaller subblocks (e.g., a charge pump or a filter), after which each of the subblocks can be designed separately. A hierarchical design methodology [1, 2] consists of two complementary design processes: top-down design and bottom-up verification, as shown Figure 1. In the top-down process, two design aspects are included:

- Circuit decomposition and architecture selection.
- Specifications propagation and sizing process.

Starting from the specifications of the system on top level, behavioral model simulation and optimization can be used to evaluate different architectures, which allows an earlier detection of design faults and provides a means by which the PLL system can be tested for its overall loop performance characteristics at a higher level prior to generating circuit-level realizations for the individual subsystem [4]. Once the architecture is selected, the following process is circuit sizing and final layout. In this paper, we will concentrate on the sizing process and therefore on the behavioral and circuit level.

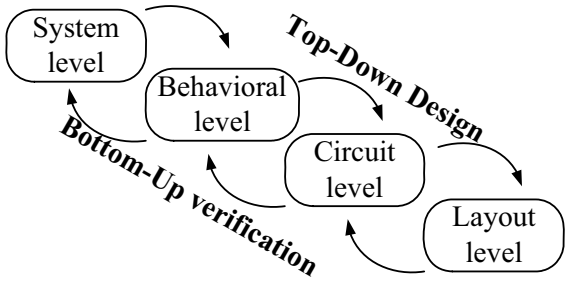

Figure 1 Analog hierarchical design levels

1.2 Circuit Sizing

Circuit sizing aims at sizing circuit parameters like transistor lengths and widths values such that performances like gain are optimized. For a given set of parameter values, the performances of the circuit are determined uniquely by simulation. The inverse map from specifications to parameter values is usually not unique. To eliminate the ambiguity in the inverse map, mathematical methods are used to realize an automatic circuit sizing, based on either deterministic or statistical optimization methods [19-21]. During the automatic sizing process, the performances have to be evaluated for a large number of different circuit parameter vectors by simulation (*simulator-in-the-loop*). Therefore for more complex mixed-signal circuits such as PLLs, simulation time on circuit level is quite long and therefore the automatic sizing, needing a large number of simulations, becomes infeasible in terms of computation time. Nowadays a partitioning of the PLL is performed and each subblock is sized separately on circuit level. On behavioral level, the design is verified, and if the specification is not met, a redesign of the subblocks is performed. We suggest doing a sizing on behavioral level under consideration of the subblock capabilities in a single top-down sizing run avoiding redesigns of the subblocks.

Such hierarchical sizing methods have been already applied in some practical designs [1, 3], in which the specification propagation are clearly described. Different behavioral models for PLL systems are presented for purely PLL simulation and verification [4-7, 22]. A recent approach for

hierarchical PLL design [17] focuses mainly on the jitter-power trade-off in the VCO design. In this paper, a new automatic hierarchical sizing process based on fast-simulating behavioral models is presented for a third-order Charge Pump PLL (CPPLL). It features:

- Automatic sizing in feasible time for a frequency synthesis application focusing on the locking time as critical performance.
- Behavioral level constraints, which guarantee the stability of the PLL for the complete specified range of output frequencies.

This paper is organized as follows. In section 2, a description of the CPPLL and its subblocks with corresponding behavioral models are presented. Then the details of the hierarchical design steps are described in section 3. After that the simulation results and the optimization time are given in section 4. Finally conclusions are presented in section 5.

2. CHARGE PUMP PLL

In this paper, we focus on the CPPLL architecture, because it can provide zero phase error, and an extended frequency range of operation. This architecture is considered as one of the simplest and most effective design platforms and is widely adopted in many PLL systems. A block diagram of a CPPLL is shown in Figure 2. It consists of five blocks, namely phase frequency detector (PFD), charge pump (CP), loop filter (LF), voltage controlled oscillator (VCO) and divider. We will describe the blocks one by one in the remainder of this section. The PLL system is typically mixed analog-digital in nature. The behavioral models will be presented in the Verilog-A language [18]. The *@ statements* are making processes sensitive to events and are well suited to describe the digital event characteristic of the PFD and divider blocks of a PLL. More information about behavioral models of PLLs can be found in [7].

One application of CPPLLs is frequency synthesis. In this case the goal is to generate a signal of changing frequency for clock generation. The output frequency can be set to multiples of the reference input frequency F_{ref} by changing the ratio N of the divider: $F_{out} = NF_{ref}$.

As the value of the divider is changed, the CPPLL is at first oscillating in an unlocked frequency which is very different of the desired frequency. The locking time is defined as the time taken by the CPPLL to synchronize with or to lock onto the new frequency [10]. In this application the fast dynamic is more critical than the noise rejection in other applications, e.g. clock recovery. Therefore our goal in designing the CPPLL is to minimize the locking time. Additionally the stability of the PLL system must be guaranteed. Therefore the phase margin and reference frequency to unity-gain-bandwidth ratio (RUR) must be considered.

Usually the PLL locking time and stability are analyzed in the *s*-domain [11]. But the *s*-domain analysis is based on a continuous time approximation of the CPPLL and can not accurately estimate the locking time. Besides *s*-domain analysis, two analyses using an event-driven non-linear model for a 2nd-order CPPLL in [5] and state-space equations for a 3rd-order CPPLL in [12] are proposed, which can provide exact models of the PLL dynamics. But it is difficult to set up or adapt these equations to other architectures especially for higher-order PLLs. In contrast behavioral models can be set up and modified for other applications or higher-order PLLs rather easily relative to [5,12] and provide a more accurate estimation of the locking time than *s*-domain analysis.

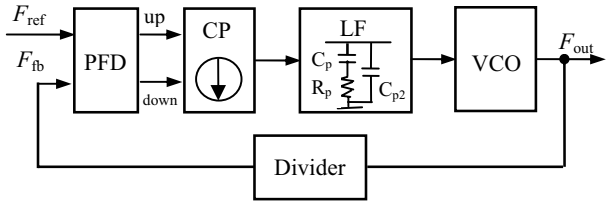

Figure 2 Block diagram of charge-pump PLL system

2.1 Phase Frequency Detector

A PLL that uses a phase frequency detector (PFD) will lock under any condition, if the PLL system itself is stable. Therefore, PFD is the preferred phase detector type compared to other phase detectors as Multiplier PDs or JK-Flip-flops.

A well-known ideal state machine diagram of a digital PFD is depicted in [8,9]. The behavioral model for PFD, based on the state diagram, is presented in List 1. If either the reference input signal F_{ref} or feedback signal F_{fb} rises across the threshold voltage*, an event process would be produced, similar to the event process known in digital behavioral languages, e.g. VHDL or Verilog.

List 1 Behavioral representation of PFD in Verilog-A

```
module PFD (ref,fb,u,d)
.....
 integer state;
 analog begin
   @(cross(V(ref)-v_th), +1,ttol)          //Event Block
      if (state <1)  state = state +1;
   @(cross(V(fb)-v_th), +1,ttol)           //Event Block
      if (state >-1)  state = state - 1;
    V(u) <+ transition((state==+1) ? VDD : 0.0, td_u, tt);
    V(d) <+ transition((state==-1) ? 0.0 : VDD, td_d, tt);
 end
endmodule
```

2.2 Charge Pump

The ideal charge pump (CP) consists of two current sources: source and sink currents. In Fig. 3 the current steering configuration schematic of the CP is depicted. When the *up (down)* signal is active, the source current flows into (out of) the loop filter, and so the output voltage of the loop filter rises up (drops down), which forces a higher (lower)

* Defined here as half the supply voltage.

oscillation frequency. Note that the *up* and *down* signals can not be active at the same time. Additionally, there are the two complementary signals, $\overline{up}$ and $\overline{down}$, as shown in Figure 3. The dummy switch is used to reduce the charge injection during a change of the switch state.

Normally, the CP and the PFD could be written together in a single behavioral model, but in order to conveniently do mixed level/model simulations using *Spectre*, a separate behavioral model for the CP is written, as shown in List 2. In a practical CPPLL, a CMOS implemented CP current source will suffer from various non-idealities, e.g. channel length modulation. In the behavioral model, these non-idealities are represented by mismatch of the source and sink currents, *Mis,* and the maximum and minimum output voltages, *v_max* and *v_min*.

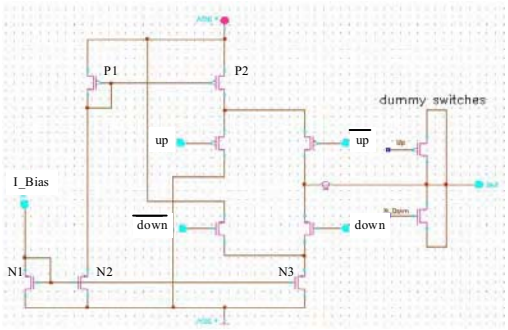

Figure 3 Charge pump schematic

List 2 Behavioral representation of CP in Verilog-A

```
module CP (Iout, Down, Up, N_Down, N_Up, );
  ......
  parameter real Ip=25.0e-6;              // the value of bias current
  parameter real v_max=1.3, v_min=0.2  // limit the output voltage
  parameter real Mis=0.05; // mismatch of charge & discharge current
  integer state;
    analog begin
        @(cross(V(Up)-v_th, 1))       begin   state = -1;   end
        @(cross(V(Up)-v_th, -1))      begin   state =0;     end
        @(cross(V(Down)-v_th, 1))   begin   state = 1;    end
        @(cross(V(Down)-v_th, -1))  begin   state =0;     end
        @(cross(V(Iout)-v_max, 1))   begin   state =0;     end
        @(cross(V(Iout)-v_min, -1))   begin   state =0;     end
      I(Iout)<+transition(Ip*state*(1+state*Mis), Delay, TransTime);
    end
endmodule
```

2.3 Loop Filter

On behavioral level we use a structural model for the loop filter (LF), which is composed of a resistor R_p in series with a capacitor C_p as shown in Figure 2. The CP current sources and the capacitor form an integrator and the resistor R_p introduces a zero point to improve the phase margin and the transient response of the whole CPPLL. At the beginning or in the end of each PFD pulse, a ripple of value I_pR_p arises on the control voltage, which modulates the VCO output frequency and introduces excessive jitter at the output. A small capacitor C_{p2} is added in parallel with R_p and C_p in order to suppress this ripple. Therefore the whole system is a third-order CP PLL.[Δ]

2.4 Voltage Controlled Oscillator

Voltage controlled oscillators (VCOs) can be realized in different architectures depending on the requirements, e.g. ring oscillators or LC oscillators. Here a five-stage single-end ring oscillator is selected, as shown in Figure 4.

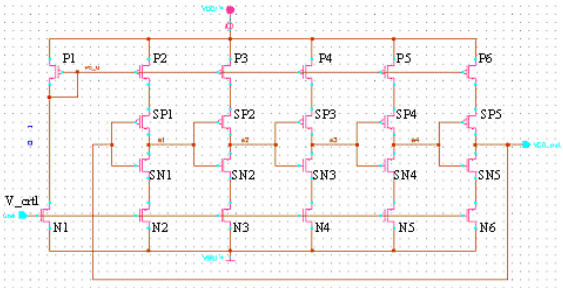

Figure 4 Five-Stage ring VCO Schematic

In this kind of VCO, the input voltage controls the current through the delay elements, thus determines the delay time of each stage and thus determines the output oscillation frequency. An ideal VCO generates a periodic signal whose frequency is a linear and limited function of the controlling voltage, as shown in Figure 5. The output frequency f_{out} can be expressed as:

$$f_{out} = f_{min} + K_{VCO} \cdot (V_{in} - V_{min}) \qquad (1)$$

$$K_{VCO} = (f_{max} - f_{min}) \Big/ (V_{in} - V_{min}) \qquad (2)$$

where f_{min} is the minimal and f_{max} is the maximum output frequency, V_{max} and V_{min} correspond to minimal and maximum input voltages and V_{in} is the output voltage of the loop filter. The behavioral model for the VCO implementing an ideal linear transformation from voltage to frequency is presented in List 3.

List 3 Behavioral representation of VCO in Verilog-A

```
module vco(V_tune, VCO_out);
  ......
  analog begin
    freq=(V(V_tune)-Vmin)*(Fmax-Fmin)/(Vmax-Vmin)+Fmin;
    if(freq>Fmax) freq=Fmax;
    if(freq<Fmin) freq=Fmin;
    phase=idtmod(freq, 0.0, 1.0, -0.5);
    @(cross(phase-0.25,1,ttol)) begin
        Vout=vhi;
    end
    @(cross(phase+0.25,1,ttol)) begin
        Vout=vlo;
    end
    V(VCO_out)<+transition(Vout,0,tt);
  end
endmodule
```

[Δ] A practical CPPLL system is at least third-order due to the parasitic capacitance at the input of VCO, with or without the explicit C_{p2}.

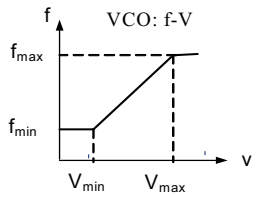

Figure 5 *f* vs. *v* for a ideal VCO

2.5 Divider

Nowadays programmable digital dividers are widely used in frequency synthesizers. The behavioral model in Verilog-A of this digital block is presented in list 4. We will not use a circuit realization during the whole sizing process, as it would significantly increase simulation time without increasing accuracy of the results. The sizing of the divider is not critical as it is a digital block and can be conducted independently from the CPPLL sizing. The model can generate a 50% duty cycle square signal. The maximum value N_{max} and minimum values N_{min} of the divider depend on the desired output frequency range of the PLL as shown in the next section in detail. Additionally, a point of time T_{jump} is set where the divider setting is changed from N_{min} to N_{max} in order to measure the locking time of the output frequency jumping from f_{min} to f_{max}, which is the worst-case condition for PLL acquisition.

List 4 Behavioral representation of divider in Verilog-A

```
module clk_div (clock_out, clock_in);
  ……
  parameter real N_max= 20 ;       // maximum value of divider
  parameter real N_min= 6 ;        // minimum value of divider
  parameter T_jump=8e-6;           // the jump time
 analog begin
   @(cross((V(clock_in)-v_th),1)) begin
       count=count+1;
       if ($realtime>T_jump) M=N_max;
       else      M=N_min;
       if (count>=M) count=0;
       n = (2*count >= M);
    end
    V(clock_out)<+ transition(n? v_high : v_low, td, tt)
  end
endmodule
```

3. HIERARCHICAL SIZING PROCESS

For the sizing of the CPPLL, a reference input frequency of 25 MHz was chosen. The CPPPL was realized in a 130 nm technology with a supply voltage of 1.5 V. The automatic sizing algorithm is based on a gradient-based optimization algorithm supplied with the commercial sizing tool *WiCkeD* [23]. To run an optimization, the corresponding testbenches, simulators and parameter ranges need to be set up. The sizing is a two-stage process.

First an automatic sizing of the CPPLL is conducted at behavioral level. A sizing of the behavioral models is conducted using the model parameters CP current I_p and VCO gain K_{VCO} as well as the loop filter elements, so that the locking time T_s is minimized and the specification given in table 1 are met for the output frequency range and stability determined by the phase margin and the reference frequency to unity-gain-bandwidth ratio (RUR) for all divider values.

Table 1 CPPLL specifications.

Type	*Specifications*	*Values*
Performance	Output frequency range	150MHz to 500 MHz
	Phase margin for all N	$\geq 45°$
	RUR for all N	≥ 20
	Locking time	minimal

In a second step a sizing run is conducted for the VCO and CP individually on circuit level whereby the found optimal model parameters I_p and K_{VCO} are propagated down as specification. Fig. 7 illustrates the hierarchy in the sizing process. In the following the two sizing steps are explained in detail.

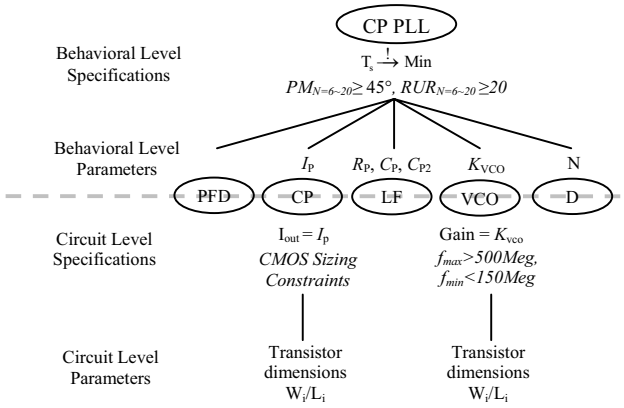

Figure 6 CPPLL hierarchical sizing overview

3.1 Behavioral Level Sizing

In order to achieve the requirement of the output frequency range the maximal and minimal divider ratios must be set to N_{min}=6 and N_{max}=20. The output frequency range of the CPPLL can be directly propagated down to the performance requirement for the VCO circuit level design, because the output frequency range of the VCO determines the output frequency range of the complete CPPLL. The locking time T_s depends on the change in the divider ratio. Here we define the locking time in the worst case, i.e. the output frequency directly jumps from f_{min} (=150 MHz) to f_{max} (=500 MHz), which means that the value of divider N changes from 6 to 20 at a certain point in the simulation.

The PLL system should be stable in any case, otherwise the locking time T_s cannot be defined. From the research results in [12], the ratio between the input frequency F_{ref} and the unity-gain-bandwidth F_{UGB} of the PLL has to be bounded below in order to guarantee the stability of the 3rd-order CPPLL system during the optimization process:

$$RUR = {F_{ref}}/{F_{UGB}} \geq RUR_{min}, \qquad (3)$$

Moreover, the realistic ranges of the capacitances and the CP current must be defined under consideration of area and power consumption: the capacitor can vary from 1pF to 100pF, the resistance from the 1 kΩ to 100kΩ and the CP current from 5μA to 100μA. Through using design space exploration [13-15] on VCO circuit level, we obtain that the gain K_{vco} of VCO can vary from about 500MHz/V to 2.8GHz/V. It is very important for the top down sizing process to restrict the model parameters to a range which can be implemented by the underlying circuits in order to avoid that the sizing process at behavioral level lead to unrealistic model parameters, which actually can not be implemented.

A transient simulation with *Spectre* by Cadence [24] is used to get the specification on the locking time T_s. Based on *s*-domain analysis the phase margin *PM* and unity-gain -bandwidth ω_{UGB} are calculated with *Matlab* by Mathworks [25].

The conceptual open loop bode plot of the CPPLL is shown in Figure 7a, the phase margin *PM* is expressed by:

$$PM = \arctan\left(\omega_{UGB}/\omega_Z\right) - \arctan\left(\omega_{UGB}/\omega_{p3}\right)$$
$$where \quad \omega_Z = \frac{1}{R_P C_P} \quad \omega_{p3} = 1\Big/\left[R_p \frac{C_P C_{P2}}{C_P + C_{P2}}\right] \tag{4}$$

Here ω_z is the zero point and ω_{p3} is the third pole. Since the *PM* is a concave function of *N*, as shown in Figure 7b, the following conditions are sufficient to guarantee the phase margin at each divider value to be above the lower bound of 45°.

$$PM_{N\max} \geq 45° \text{ when } N = 20$$
$$\& \quad PM_{N\min} \geq 45° \text{ when } N = 6 \tag{5}$$

In Figure 7c,d, the unit gain bandwidth ω_{UGB} decreases while the corresponding RUR increases with increasing *N*. Hence it is sufficient to only consider the RUR at the minimum divider value.

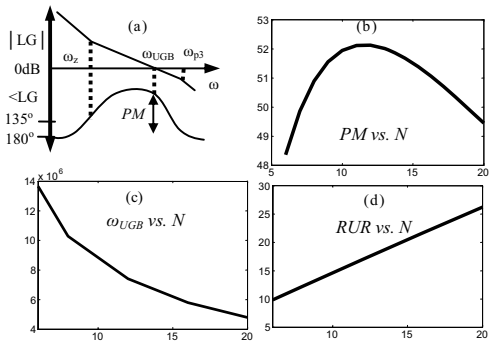

Figure 7 (a) Bode plot of third-order loop (b) PM vs. Divider (c)Unity-gain-bandwidth vs. Divider (d) Ratio vs. Divider

Depending on the above discussion, a proper behavioral level parameter set $\mathbf{p} = [I_p, C_p, C_{p2}, R_p, K_{vco}]$ is computed so that the locking time is minimized considering the constraints for the stability. We set $RUR_{\min}$ in (3) to 20 based on test experience. That is:

$$\min_{\mathbf{p}} T_s$$
$$\text{s.t. } RUR \geq 20;\ PM_{N\max} \geq 45°;\ PM_{N\min} \geq 45° \tag{6}$$

The results of the behavioral level optimization and the initial values are listed in Table 2. The VCO controlling voltage, which is approximately the PLL output frequency, is shown in Fig. 8 for the initial PLL and the optimized PLL.

Table 2 Behavioral level sizing

	Parameters	Performances
Initial	C_p=50pF; C_{p2}=10pF I_p=9μA; R_p=30kΩ K_{vco}=550MHz/V	PM_{Nmax}=43.5° PM_{Nmin}=41.7° RUR=56; T_s=6.4μs
Optimized	C_p=54.2pF; C_{p2}=8.19pF I_p=38μA; R_p=9.05kΩ K_{vco}=1.07GHz/V	PM_{Nmax}=45.01° PM_{Nmin}=49.6° RUR=20; T_s=1.3μs

3.2 Circuit Level Sizing

Following the methodology, the behavioral level parameters serve as performance requirements for the circuit level subblocks and are mapped onto specifications on circuit level. Here we focus on the designs of two analog blocks: CP and VCO.

As mentioned above, the PLL output frequency range is the performance requirement for the VCO. Therefore, the output frequency at the maximum input voltage $f(V_{max})$ and at the minimum input voltage $f(V_{min})$ are treated as constraints in the automatic sizing process, where the difference of the K_{vco} of the VCO behavioral model (2) and the actual *gain* of the VCO circuit must be minimized. The design parameters $\mathbf{d_{VCO}}$ are the transistor geometries (widths and lengths) of the VCO:

$$\min_{\mathbf{d}_{VCO}} (gain - K_{vco})^2$$
$$\text{s.t. } f(V_{\max}) \geq 500MHz;\ f(V_{\min}) \leq 150MHz \tag{7}$$

In the CP schematic in Figure 3, the CMOS transistor pairs, N1-N3 and P1-P2, make up current mirrors, which serve as current sources. Therefore these transistors should e.g. work in saturation. According to the sizing rules given in [16], we can setup a set of corresponding constraints for the optimization process (8). The deviation of the source current I_u as well as the sink current I_d from the behavioral model CP current I_p should be minimal. The design parameters $\mathbf{d_{CP}}$ are the transistor widths/lengths of the CP:

$$\min_{\mathbf{d}_{CP}} (|I_{u,d}| - I_p)^2$$
$$\text{s.t. CMOS sizing rules} \tag{8}$$

4. VERIFICATION

In our bottom-up verification process, we use a circuit level simulation of the sized CPPLL.

The simulation result shown in figure 8 verifies that the sized circuit reached an even better locking time T_s=1.16μs than predicted by the behavioral level simulation and that

the output signal is stable. The difference in T_s between the behavioral optimized value and the final circuit level simulation comes mainly from the nonlinear relationship of the *f-v* of the VCO. Comparing the simulation cost, we can conclude that the optimization process based on flat circuit simulation would last some days or weeks, while the whole hierarchical optimization process can be finished in some minutes. The speed-up is remarkable, a sizing of the presented PLL could be achieved in 20 minutes.

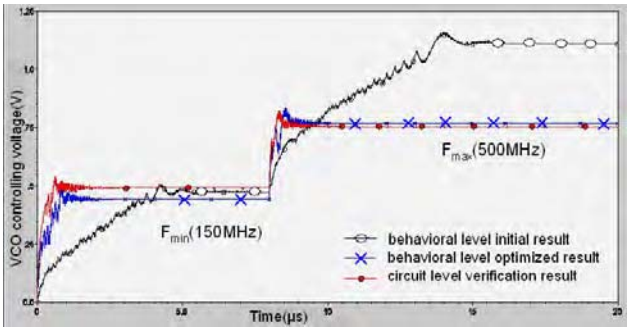

Figure 8 Transient simulation results on both levels

Table 3 Simulation results on each level

Level	T_s	simulation time for 20μs transient simulation
Behavioral	1.3 μs	38.5sec
Circuit	1.16 μs	1h 32min 11sec

5. CONCLUSION

We have shown how the locking time and stability of the CPPLL can be accurately and quickly computed by behavioral level simulation. A CPPLL circuit can be quickly and automatically sized to satisfy the performance requirements in some minutes. The simulation results of the final circuit level simulation have verified the feasibility of our hierarchical design approach on a CPPLL. We can conclude that with the help of the hierarchical sizing methodology, the design process can be remarkably shortened.

ACKNOWLEDGMENTS

The authors would like to thank B. Obermeier, M. Pronath of MunEDA and Prof. Dr. H. L. Zapf of FH Muenchen for many helpful discussions.

REFERENCES

[1] H. Chang, A. Sangiovanni-Vincentelli, F. Balarin, etc, A Top-Down, Constraint-Driven Design Methodology for Analog Integrated Circuits, *IEEE CICC,* pp. 533-536, 1992.

[2] G. G. E. Gielen and R. A. Rutenbar, Computer-Aided Design of Analog and Mixed-Signal Integrated Circuit, *Proceeding of THE IEEE*, VOL. 88 NO.12 Dec. 2000.

[3] I. Vassiliou, H. Chang, A. Demir, A. Sangiovanni-Vincentelli, A Video Driver System Designed Using a Top-Down Constraint-Driven Methodology, *IEEE ICCAD,* pp. 463-468, 1996.

[4] B. A .A. Aanto, E. M. EL-Turky and R. H. Leonowich, Behavioral Modeling Phase-Locked Loops for Mixed-Mode Simulation, *Analog Integrated Circuits and Signal Processing*, No.10, pp. 45-65, 1996.

[5] P. Acco, M. P. Kennedy, C. Mira, B. Morley and B. Frigyik, Behavioral Modeling of Charge Pump Phase Locked Loops, *IEEE ISCASS*, pp. 375-378, 1999.

[6] M. Hinz, I. Koenenkamp and E. H. Horneber, Behavioral Modeling and Simulation of Phase-Locked Loops for RF Front Ends*, Proceeding of the 43rd IEEE MSCS*, 2000.

[7] K. Kundert, Predicting the phase noise and jitter of PLL-based frequency synthesizers*,* Available from *http://www.designers-guide.org*.

[8] R. E. Best, *Phase-Locked Loops Design, Simulation and Application*, McGRAW-Hill, 2003.

[9] B. Razavi, Design of Monolithic Phase-Locked Loops and Clock Recovery Circuits - a Tutorial. *IEEE Press*, 1996.

[10] C. M. Chie and W.C. Lindsey, Phase-Locked Loops: Applications, Performance Measures and Summary of Analytical Results*, IEEE Press*, NY, 1986.

[11] F. M. Gardner, Charge pump phase- locked loops*, IEEE Trans. Communications,* VOL. COM-28, No.11, November 1980.

[12] P. K. Hanumolu, M. Brownlee, K. Mayaram and U.K. Moon, Analysis of charge pump phase locked loops*, IEEE Transactions of Circuit and System,* VOL.51. No.9, Sept. 2004.

[13] D. Müller, G. Stehr, H. Gräb, U. Schlichtman, Deterministic Approaches to Analog Performance Space Exploration*, IEEE DAC*, 2005.

[14] B. D. Smedt, G. E. Gielen, WATSON: Design Space Boundary Exploration and Model Generation for Analog and RF IC Design*, IEEE Trans. CAD*, 2003.

[15] F. De Bernardinis, A. Sangiovanni-Vincentelli, A Methodology for System-Level Analog Design Space Exploration, *IEEE DATE*, 2004

[16] H. Graeb, S. Zizala, J. Eckmueller, K. Antreich, The sizing rules method for analog integrated circuit design, *IEEE/ACM ICCAD*, pp. 343-349, 2001.

[17] S. K. Tiwary, S.Velu, R. A. Rutenbar and T. Mukherjee, Pareto Optimal Modeling for Efficient PLL Optimization*, Nanotech,* Vol. 2 pp. 195 – 198, 2004.

[18] *Verliog-AMS Language Reference Manual: Analog & Mixed-Signal Extensions to Verilog-HDL,* version 2.1. Available from *www.verilog-ams.com.*

[19] W. Nye, D. Riley, and A. Sangiovanni-Vincentelli, An optimization-based system for the design of integrated circuits, *IEEE ICCAD*, 1988.

[20] K. Antreich, J. Eckmueller, H. Graeb, M. Pronath, etc, WiCkeD: Analog circuit synthesis incorporating mismatch, *IEEE CICC,* pp. 511-514, May 2000.

[21] F.Medeiro, F.V. Fernandez, Dominguez-Castro, A. Rodriguez-Vasquez, A statistical optimization-based approach for automated sizing of analog cells, *IEEE/ACM ICCAD*, 1994.

[22] M. H. Perrott, Fast and Accurate Behavioral Simulation of Fractional-N Synthesizers and other PLL/DLL Circuit, *IEEE DAC*, pp. 498-503, June 2002.

[23] *www.muneda.com.*

[24] *www.cadence.com.*

[25] *www.mathworks.com.*