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

J. Zou, D. Mueller,
H. Graeb, U. Schlichtmann,
E. Hennig, R. Sommer

Institute of Electronic Design Automation, TU Munich,
Infineon Technologies AG
Munich, Germany
Outline

- Analog Hierarchical Design Methodology
- Phase-Locked Loops - Introduction
  - Architecture Selection
  - PLL Building Blocks
  - PLL Specifications
- Automatic Hierarchical Sizing of a Charge Pump PLL
  - Definition & Propagation of Specifications
  - Automatic Behavioral Level Sizing
  - Automatic Circuit Level Sizing
  - Design Verification
- Conclusion & Outlook
Analog Hierarchical Design Levels

Top-Down:
- Circuit Decomposition & Architecture Selection
- Specifications Propagation & Sizing Process

Bottom-Up:
- Verification Process
- Design Space Exploration

Our work focuses on the automatic sizing processes on the both levels: behavioral and circuit levels.
Elements of the Analog Hierarchical Design Methodology

Circuit Decomposition
- f_i -> Phase Detector
- Loop Filter -> A-1
- VCO
- Ctrl. logic
- f_o

Behavioral Modeling
- IN -> VDD -> OUT
- IN- -> VSS

Design Space Exploration
- B=[gain, bandwidth, ...]

Circuit Sizing
- P=[w1, w2, l1, l2, ...]

Specifications Propagation
- automatic

PLL Circuit
time-consuming
Outline

- Analog Hierarchical Design Methodology
- Phase-Locked Loops Introduction
  - Architecture Selection
  - PLL Building Blocks
  - PLL Specifications
- Automatic Hierarchical Sizing of a Charge Pump PLL
  - Definition & Propagation of Specifications
  - Automatic Behavioral Level Sizing
  - Automatic Circuit Level Sizing
  - Design Verification
- Conclusion & Outlook
PLL as Example Circuit for Analog Hierarchical Design

■ PLL Function

A phase-locked loop (PLL) is a control system to generate an output signal which is synchronized in frequency and phase to an input signal.

\[ F_{\text{out}} = F_{\text{ref}} \times N \]

■ PLL Application

Clock Recovery / Frequency Synthesis / ......
Charge-pump PLL can provide zero phase error, an extended frequency range of operation and low cost.

Charge-Pump PLL

Circuit's Decomposition

Architecture Libraries

- Multiplier PD
- EXOR PD
- JK-Flip-flop
- PFD
- Hardware
- Software
- ...
parameter real td = 0;  // minimum divider value
parameter real N_min = 6;  // maximum divider value
parameter real N_max = 20;  // output frequency jump time
integer count, n, M;
analog begin
    @(cross((V(clock_in)-v_th),1)) begin
        if ($abstime>=JumpTime) M=N_max;
        else   M=N_min;
        count=count+1;
        if (count>=M) count=0;
        n = (2*count >= M);
    end
    V(clock_out)<+ transition(n? v_high : v_low, td, tt);
......
module pfd (ref, fb, u, d);
output u, d; input ref, fb;
parameter real v_high=1.5;
parameter real td=10p;
parameter tr=120p;
....
integer state;
analog begin
  @(cross(V(ref)-v_high/2), +1,ttol)
    if (state <1)  state = state +1;
  @(cross(V(fb)-v_high/2), +1,ttol)
    if (state >-1) state = state - 1;
V(u) <+ transition(state==+1) ? v_high : 0.0, td_u, td); V(d) <+ transition((state==-1) ? 0.0 : v_high, td_d, td);
end
endmodule

module ChargePump (Iout, Down, N_Down, N_Up, Up, V_bias_n, V_bias_p);
output Iout; input Down, N_Down, N_Up, Up, V_bias_n, V_bias_p;
electrical Iout, Down, N_Down, N_Up, Up, V_bias_n, V_bias_p;
parameter real v_high = 1.5, v_low = 0; parameter real v_th = (v_high-v_low)/2;
parameter real TransTime=10p from (0:1000000);parameter real Delay=1p from (0:1000000);
parameter real Ip=25.0e-6; // the value of output current
parameter real v_max=1.3, v_min=0.2; // limit the output voltage
parameter real Mis=0.05; // mismatch of charge and discharge current integer state;
analog begin
  @(cross(V(Up)-v_th, 1))        begin
    state = -1;
  end
  @(cross(V(Down)-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 =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

All behavioral models are implemented using Verilog-A.

K. Kundert: Predicting the phase noise and jitter of PLL-based frequency synthesizers
PLL Specifications for Frequency Synthesis Application

- **Nonlinear Acquisition Process**
  The locking time $T_s$ is defined as the time taken by the PLL to synchronize with or to lock the new frequency.
  On circuit level, simulation: time-consuming (hours/days) optimization for $T_s$: impossible

- **Linear Process in the Lock-In State**
  Damping factor, Nature Frequency, Phase Margin, Unity-Gain-Bandwidth, …
  Matlab models, can help analyze the PLL system cannot automatically size the parameters

Our goal: to automatically size the PLL circuit, considering the performances in nonlinear and linear processes at the same time!
Outline

- Analog Hierarchical Design Methodology
- Phase-Locked Loops Introduction
  - Architecture Selection
  - PLL Building Blocks
  - PLL Specifications
- Automatic Hierarchical Sizing of a Charge Pump PLL
  - Definition & Propagation of Specifications
  - Automatic Behavioral Level Sizing
  - Automatic Circuit Level Sizing
  - Design Verification
- Conclusion & Outlook
## Definition of PLL Specifications

<table>
<thead>
<tr>
<th>Type</th>
<th>Specifications</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performances</td>
<td>Output frequency range</td>
<td>150MHz to 500 MHz</td>
</tr>
<tr>
<td></td>
<td>Locking time $T_s$ *</td>
<td>minimal</td>
</tr>
<tr>
<td></td>
<td>Phase margin for all $N$</td>
<td>$\geq 45^\circ$</td>
</tr>
<tr>
<td></td>
<td>RUR* for all $N$</td>
<td>$\geq 20$</td>
</tr>
<tr>
<td>Operation</td>
<td>Reference frequency</td>
<td>25MHz</td>
</tr>
<tr>
<td></td>
<td>Supply voltage</td>
<td>1.5V</td>
</tr>
<tr>
<td>Technology</td>
<td>Infineon</td>
<td>130nm</td>
</tr>
</tbody>
</table>

* We define the locking time in the worst case, i.e. output frequency directly jumps from 150MHz to 500MHz. This determine the divider value $N$ jumping from 6 to 20.

* $\text{RUR} = \frac{\text{reference input frequency}}{\text{the unity-gain-bandwidth of PLL}}$
Performance Propagation on CP PLL

Sizing on behavioral level

Behavioral Level Specifications

Behavioral Level Parameters

Circuit Level Specifications

Sizing on Circuit level

Circuit Level Parameters

CP PLL

\[ T_s \rightarrow \text{Min} \]

\[ PM_{N=6-20} \geq 45^\circ, \quad RUR_{N=6-20} > 20 \]

\[ I_{out} = I_B \]

\[ R_{LP}, C_{LP}, C_{LP2}, K_{VCO} \]

\[ N \]

PFD

CP

LF

VCO

Divider

\[ CMOS \text{ Sizing Constraints} \]

\[ f_{max} > 500 \text{Meg}, \quad f_{min} < 150 \text{Meg} \]

\[ \text{transistor dimensions } \frac{W_i}{L_i} \]

\[ \text{transistor dimensions } \frac{W_i}{L_i} \]
Behavioral Level Parameters Variation Range Definition

Parameters set $\mathbf{p} = [I_B, C_{LP}, C_{LP2}, R_{LP}, K_{VCO}]$

- $5 < I_B$ in $\mu$A $< 100$
- $20 < C_{LP}$ in $\mu$F $< 200$
- $1 < C_{LP2}$ in $\mu$F $< 15$
- $1 < R_{LP}$ in k$\Omega$ $< 100$
- $0.5 < K_{VCO}$ in GHz/V $< 5.5$

Design Space Exploration on VCO Circuit

Feasible region of $K_{VCO}$

s.t. $F_{\text{max}} > 500$MHz & $F_{\text{min}} < 150$MHz

D. Müller, etc.: *Deterministic Approaches to Analog Performance Space Exploration*
Automatic Behavioral Level Sizing

- Optimization process
  \[
  \min \left( T_s \right) \quad \text{p} \\
  \text{s.t. } RUR \geq 20; PM_{N_{\text{max}}} \geq 45^\circ; PM_{N_{\text{min}}} \geq 45^\circ
  \]

- Optimization results

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Performances</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial</strong></td>
<td></td>
</tr>
<tr>
<td>( C_p = 50pF; C_{p_2} = 10pF; I_p = 9\mu A; R_p = 30k\Omega; K_{vco} = 550MHz/V )</td>
<td>( PM_{N_{\text{max}}} = 43.5^\circ; PM_{N_{\text{min}}} = 41.7^\circ; RUR = 56; T_s = 6.4\mu s )</td>
</tr>
<tr>
<td><strong>Optimized</strong></td>
<td></td>
</tr>
<tr>
<td>( C_p = 54.2pF; C_{p_2} = 8.185pF; I_p = 38\mu A; R_p = 9.05k\Omega; K_{vco} = 1.07GHz/V )</td>
<td>( PM_{N_{\text{max}}} = 45.01^\circ; PM_{N_{\text{min}}} = 49.6^\circ; RUR = 20; T_s = 1.3\mu s )</td>
</tr>
</tbody>
</table>

- Optimization time: About 250 simulations about 12min

- Optimization tool: WiCkeD of MunEDA
Automatic Sizing Process for CP

- Test bench
  
  *Two input voltage-pulse signals*

  - $\text{up}$
  - $\text{down}$

  - $I_{\text{out}}$ *output current signal*

  - $I_u$ and $I_d$

- Optimization process

  $$\min_{d_{CP}} \left( I_u - I_B \right)^2 + \left( I_d + I_B \right)^2$$

  s.t. CMOS sizing rules

- Optimization time

  - *about 2min*

Parameters set in CP: $d_{CP} = [w_1/l_1, w_2/l_2, \ldots, w_n/l_n]$
Automatic Sizing Process for VCO

- Test bench
  Two-stage Input voltage signal

\[
\begin{align*}
V_{\text{in}} & \quad V_{\text{max}} \\
V_{\text{min}} & \quad t
\end{align*}
\]

VCO output frequency

\[
\begin{align*}
f_{\text{out}} & \quad f(V_{\text{min}}) \\
& \quad f(V_{\text{max}}) \\
t & \quad t
\end{align*}
\]

- Optimization process

\[
\min_{d_{\text{vco}}} \left( \text{gain} - K_{\text{vco}} \right)^2
\]

s.t. \( f(V_{\text{max}}) > 500\text{MHz} \); \( f(V_{\text{min}}) < 150\text{MHz} \)

where \( \text{gain} = \frac{f(V_{\text{max}}) - f(V_{\text{min}})}{V_{\text{max}} - V_{\text{min}}} \)

Parameters set in VCO: \( d_{\text{vco}} = [w_1, l_1, w_2, l_2, \ldots, w_n, l_n] \)

- Optimization time

about 8 min
Simulation Time on Both Levels and the whole Sizing Time

- Comparison of the time for single transient simulation on both levels

<table>
<thead>
<tr>
<th>Level</th>
<th>simulation time for 20 μs transient simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavioral</td>
<td>38.5sec</td>
</tr>
<tr>
<td>Circuit (except divider)</td>
<td>1h 32min 11sec ≈ 92min</td>
</tr>
</tbody>
</table>

- Time for the whole Hierarchical Design Process

\[ t_{\text{hierarchical-sum}} = t_{\text{behavioral-level}} + t_{\text{circuit-level}} + t_{\text{design-verification}} \]

- CP: about 2min
- VCO: about 8min
- t_{\text{hierarchical-sum}} ≈ 112min
- t_{\text{hierarchical-sizing}} ≈ 20min
- t_{\text{hierarchical-sum}} = about 12min
Design Verification based on Circuit Level Simulation

- Simulation results before & after optimization

Behavioral level initial result: \( T_s = 5.7 \, \mu s \)

Behavioral level optimized result: \( T_s = 1.3 \, \mu s \)

Circuit level verification result: \( T_s = 1.1 \, \mu s \)

- \( F_{\text{min}}(150\text{MHz}) \)
- \( F_{\text{max}}(500\text{MHz}) \)
Design Verification based on VCO Nonlinear Model

Linear and limited VCO model

Nonlinear VCO model polynomial model (4th-order)

Top-Down Phase

Bottom-Up Phase

module vco(V_tune, VCO_out);
    ....
    real freq, phase, kvco;
    analog begin
        kvco=(Fmax-Fmin)/(Vmax-Vmin);
        freq=(V(V_tune)-Vmin)*kvco+Fmin;
        if(freq>Fmax) freq=Fmax;
        if(freq<Fmin) freq=Fmin;
        ....
    end
endmodule

freq = k4*pow(V(V_tune),4)+k3*pow(V(V_tune),3) + k2*pow(V(V_tune),2) +k1*V(V_tune) + k0;
// fitting curve’s coefficients

Design Verification based on VCO Nonlinear Model
Simulation Results of the Nonlinear Model and Circuit Level

Both simulation results are almost identical. Therefore the above used behavioral models are sufficient for verifying the PLL circuit.

\[ t_{\text{hierarchical-sum}} = t_{\text{hierarchical-sizing}} + t_{\text{Verification-with-Nonlinear-Model}} \]

- \( t_{\text{hierarchical-sum}} \approx 30\text{min} \)
- Calibration of VCO Nonlinear Model about 10min
- PLL Verification about 40sec
- About 20min
Conclusion & Outlook

Conclusion

- A CP PLL hierarchical sizing process has been presented.
- Automatic sizing process could be achieved in 20 minutes.
- Sizing constraints on behavioral level are indispensable.
- Design space exploration on circuit level confirms feasibility.

Outlook

- Extension to other specifications, e.g. noise
- Extension to other PLL architectures.