

Behavioral Modeling and Simulation of Jitter and Phase Noise in Fractional-N PLL Frequency Synthesizer

Xiaojian Mao, Huazhong Yang and Hui Wang

Dept. Electronic Engineering, Tsinghua University, Beijing, China

Oct. 21, 2004

Outline

- Introduction
- Basic Concepts
- Proposed Behavioral Model for Fractional-N PLL Synthesizer
- Simulation Experiments
- Conclusion

Introduction

- Why Fractional-N PLL is used?
 - Settling Time, Frequency Resolution, and Phase Noise
 - Reference Frequency, Bandwidth of Loop
- Why Voltage-Domain Behavioral Modeling
 - Phase Domain Models vs. Voltage-Domain Models

Outline

- Introduction
- **Basic Concepts**
- Proposed Behavioral Model for Fractional-N PLL Synthesizer
- Simulation Experiments
- Conclusion

Jitter

- Jitter, Long-Term Jitter, Period-Jitter

$$v_n(t) = v(t + j(t))$$

$$J_{long-term} = \sigma(t_{i+k} - t_k) = \sqrt{\text{var}(t_{i+k} - t_k)}$$

$$J_{period} = \sigma(t_{k+1} - t_k) = \sqrt{\text{var}(t_{k+1} - t_k)}$$

Phase Noise

- Noise Phase $\phi(t)$

$$v_n(t) = v \left(t + \frac{\phi(t)}{2 \cdot \pi \cdot f_c} \right)$$

- Phase Noise

$$L(f_m) = \frac{S_v(f_c + f_m)}{2 \cdot V_1^2}$$

Jitter In Blocks of PLL

- *PM jitter in driven blocks.*
 - *Synchronous, no memory effect.*
- *FM jitter in autonomous blocks*
 - *Accumulating*
- *Relation between Jitter and Phase Noise*
 - *If we know the jitter performance of PLL, then the phase noise can be derived*

$$J = \sqrt{a \cdot T}$$

$$L(f_m) = \frac{1}{2} \frac{a \cdot f_c^2}{a^2 \cdot \pi^2 \cdot f_c^4 + f_m^2}$$

PM Jitter in Verilog-A

- **Using *transition()* Function**

```
analog begin
```

```
    @(initial_step) seed = 716;
```

```
    @(cross(V(ref), dir, ttol)) begin
```

```
        If (state > -1) state = state - 1;
```

```
        dt = 0.707*jitter*$dist_normal(seed,0,1); // PM jitter
```

```
    end
```

```
    @(cross(V(vco), dir, ttol)) begin
```

```
        if (state < 1) state = state + 1;
```

```
        dt = 0.707*jitter*$dist_normal(seed,0,1);
```

```
    end
```

```
    l(out) <+ transition(lout*state, td + dt, tt); // PM jitter is added.
```

```
end
```


FM Jitter in Verilog-A

- Using *timer()* function

```
analog begin
```

```
    @(initial_step) begin
```

```
        seed = 286;
```

```
        next = 0.5/freq + $realtime;
```

```
    end
```

```
    @(timer(next)) begin
```

```
        n = !n;
```

```
        dT = jitter*$dist_normal(seed,0,1); // FM Jitter
```

```
        next = next + 0.5/freq + 0.707*dT; // Next Transition Updated
```

```
    end
```

```
    V(out) <+ transition(n ? Vhi : Vlo, 0, tt);
```

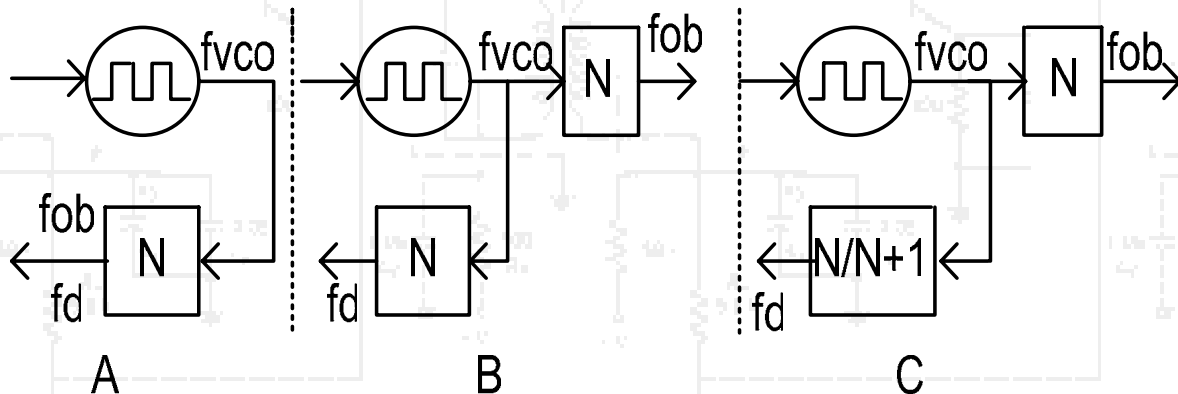
```
end
```

Outline

- Introduction
- Basic Concepts
- **Proposed Behavioral Model for Fractional-N PLL Synthesizer**
- Simulation Experiments
- Conclusion

Efficiency of the Models

- Combine jitter sources to the degree possible
- Combine VCO with Frequency Divider
 - Jitter After N-divider: $J_2 = \sqrt{N} J_1$



Combine VCO and Frequency Divider in Fraction-N PLL Frequency Synthesizer

analog begin

@(initial_step) begin

seed = -561; dT = 0;

prev = \$abstime;

Voutdiv = Vlo;

fp = \$fopen("./Maoxj_vco_fd.txt");

Next1 = (N+floor(V(ctrl)+0.5))/N; Next2 = 1;

delta = jitter * sqrt(2*N);

end

// compute the freq from the input voltage

freq = (V(in) - Vmin)*(Fmax - Fmin) / (Vmax - Vmin) + Fmin;

if (freq > Fmax) freq = Fmax;

if (freq < Fmin) freq = Fmin;

// apply the frequency divider, add the phase noise

freq = freq/N;

freq = freq/(1 + dT*freq);

// phase is the integral of the freq modulo 1

phase = idt(freq,0);

to be continued

Combine VCO and Frequency Divider in Fraction-N PLL Frequency Synthesizer

```
// Cross Up for Frequency Divider Output  
@(cross(phase-Next1, +1, ttol)) begin  
Next1=Next1+(N+floor(V(ctrl)+0.5))/N;  
Voutdiv = Vhi;  
end
```

```
// Cross Up for oscillograph // Update Jitter  
@(cross(phase-Next2, +1, ttol)) begin  
Next2=Next2+1;  
dT = delta * $dist_normal(seed, 0, 1);  
end
```

```
// Cross Down for Frequency Divider Output  
@(cross(phase-Next1+(N+floor(V(ctrl)+0.5))/2/N,+1,ttol)) begin  
Voutdiv = Vlo;  
end
```

to be continued

Combine VCO and Frequency Divider in Fraction-N PLL Frequency Synthesizer

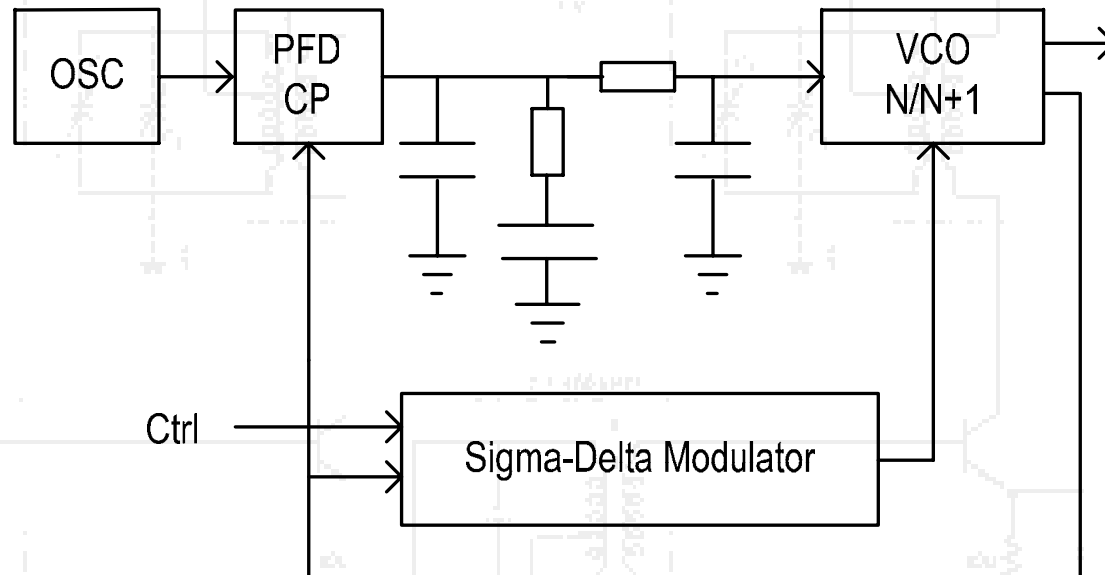
```
// Cross Down for oscillograph // Update Jitter  
@(cross(phase-Next2+0.5, +1, ttol)) begin  
    if ($abstime >= outStart) begin  
        $fstrobe( fp,"%0.20e ", $abstime - prev);  
    end  
    prev = $abstime;  
    dT = delta * $dist_normal(seed, 0, 1);  
end  
V(outdiv) <+ transition(Voutdiv, 0, tt);  
end
```

end-

Outline

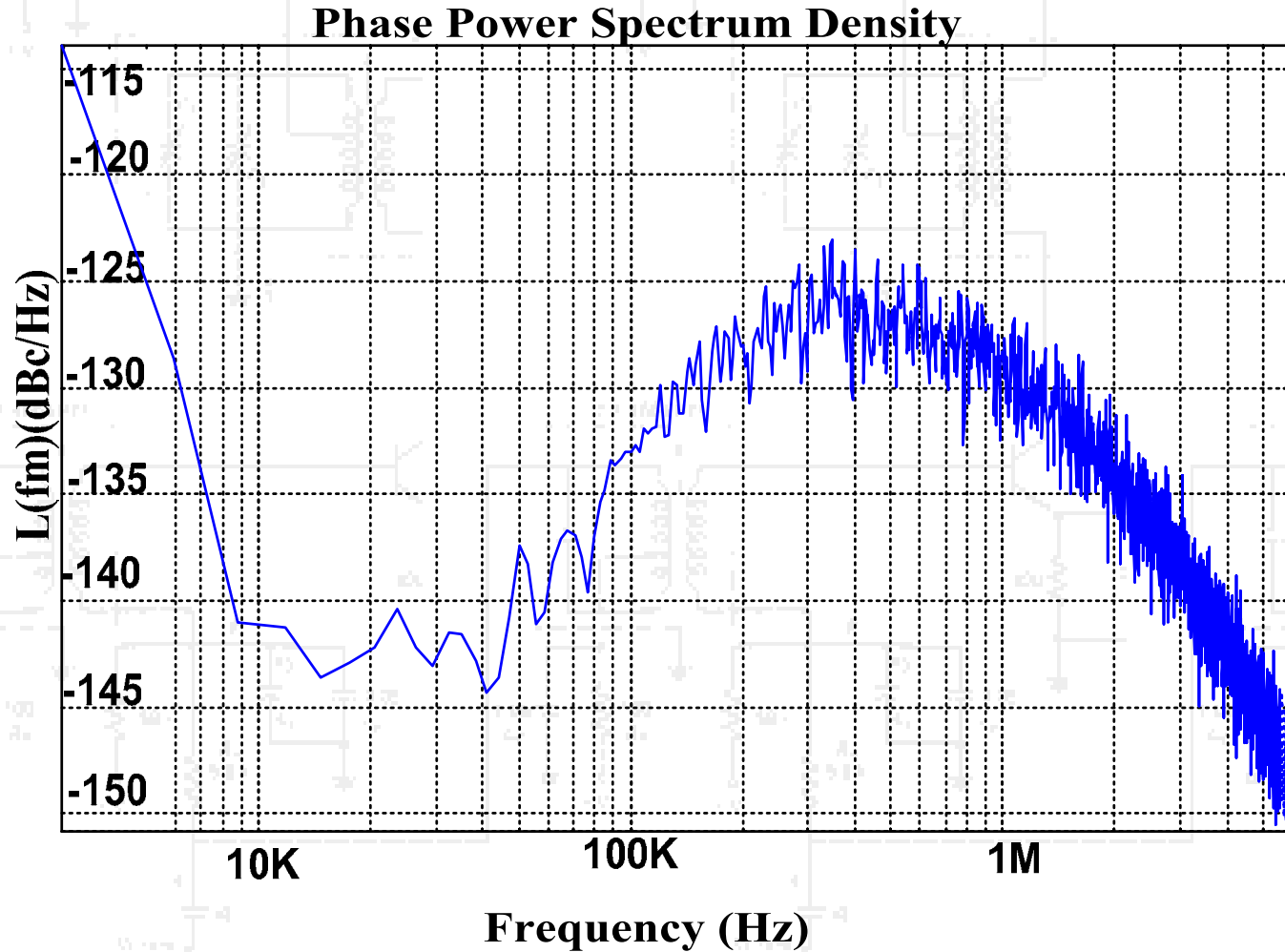
- Introduction
- Basic Concepts
- Proposed Behavioral Model for Fractional-N PLL Synthesizer
- **Simulation Experiments**
- Conclusion

A 4-order PLL Frequency synthesizer for simulation

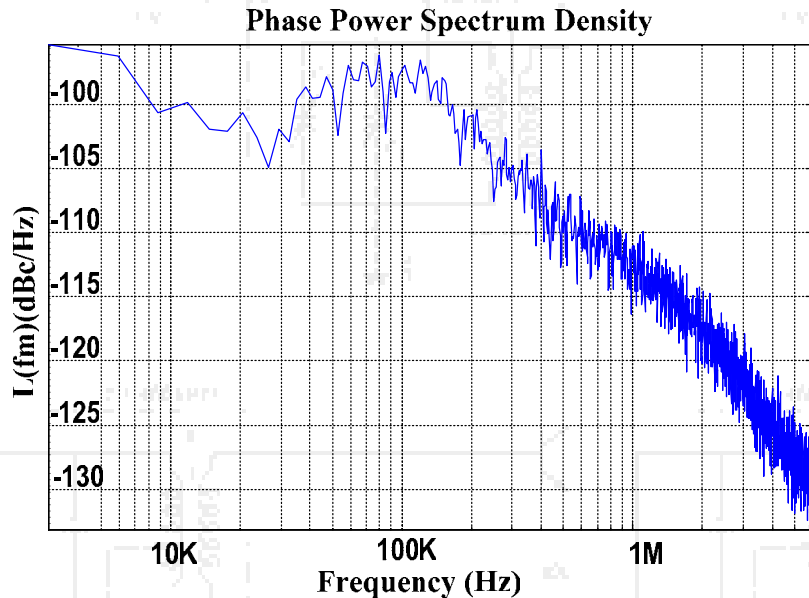


The synthesizer is chosen with a reference frequency of $f_{ref}=12\text{MHz}$ an output frequency 879.6MHz , and a divider rate of 73.3 . The bandwidth of 3rd order loop filter is 100KHz . The fractional-N is realized through a multi-modulus divider controlled by a 3-rd sigma-delta modulator

Output Phase Noise duty to Sigma-delta Modulator



Output Phase Noise of All Kinds of Noise Sources



The noise of OSC is -140dBc/Hz at 10KHz , which corresponds to a jitter of 34fs .
The noise of VCO is -100dBc/Hz at 100KHz with center frequency 878MHz and the period jitter is 55fs .
PM jitters of the other blocks about 200fs .

The simulation has been carried out on a Sun SparcUltra450 workstation with a CPU with frequency 450MHz . It takes 19 minutes to run a simulation without noise for 24 thousands reference cycles. When including all noise sources, the time cost is same because the runtime is determined by the number of activities and the noise sources of our behavioral model do not generate any extra activities. However, the SparcUltra450 always runs into core dump problem if the behavioral model with jitter does not take the method proposed in this paper.

Outline

- Introduction
- Basic Concepts
- Proposed Behavioral Model for Fractional-N PLL Synthesizer
- **Simulation Experiments**
- Conclusion

Conclusion

- *An improved methodology for modeling and simulating the phase noise and jitter performance of fractional-N phase-locked loops was presented.*
- *This method compensates Kundert's realization in [1] to fractional-N frequency synthesizers.*
- *The simulation is done at the behavioral level, and so is efficient enough to be applied in a wide variety of applications.*
- *The behavioral models are calibrated from circuit-level noise simulations, and so the high-level simulations are accurate.*

[1] K. Kundert, "Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers," Available from www.desingers-guide.com, May, 2003.



Thanks