A Design of First-Order Delay-Line DPLL in 1.2μm CMOS Technology

Seki, Ikuo
Department of Electrical Enginering, Kyushu University : Graduate Student | Toshiba, Co.

Nakashi, Kenichi
Department of Electronic Device Engineering, Kyushu University

Ushida, Mitsuhiko
Department of Electrical Engineering, Kyushu University : Graduate Student | Kawasaki Heavy Industry, Co.

Taniguchi, Kenji
Department of Electronic Device Engineering, Kyushu,University

https://doi.org/10.15017/1474940

出版情報：九州大学大学院システム情報科学紀要. 1, pp.45-50, 1996-09-27. 九州大学大学院システム情報科学研究院
バージョン：
権利関係：

九州大学大学院
システム情報科学研究科報告
第1巻　第1号　平成8年9月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.1, No.1, September 1996

A Design of First-Order Delay-Line DPLL in 1.2μm CMOS Technology

Ikuo SEKI*, Kenichi NAKASHI**, Mitsuhiko USHIDA***and Kenji TANIGUCHI**

(Received June 21, 1996)

Abstract: This paper describes a CMOS 1st-order delay-line DPLL in 1.2μm technology for clock regeneration. We have employed a parallel-architecture PC (Phase Comparator) to improve the speed and a DCO (Digitally Controlled Oscillator) without timing hazard. And we have also laid it out in 1.2μm CMOS, and simulated its performance by SPICE as well as logic simulation. Results show that the DPLL operates up to 60MHz, and that lock-in ranges are +5/-5% for regular "10" input and +5/-5% for $2^{13}-1$ PRBS (Pseudo-Random Bit Sequence) input, respectively.

Keywords: DPLL, Delay-line, CMOS, Clock regeneration, Parallel architecture

1. Introduction

High-speed and large volume digital data transmission is required to deliver digital information through the network. In that area, precise clock regeneration is strongly required to ensure very low error rate data transmission through the optical link or RF wireless network systems. PLL (Phase Locked Loop) is the most suitable component to regenerate clock from received data. And from the system point of view, since at the inter-chip or intra-chip level, signal communication or clock distribution is one of the most concerned problem, integrated on-chip PLL is implemented widely from microprocessors to memories to solve these problems. And furthermore, personal portable equipment applications require small number of parts and battery operation.

DPLL (Digital PLL) is suitable for these applications, because it requires no external components such as resistor or capacitor needed in the LPF part of analog PLL and because of low-power characteristics of CMOS technology employed. Since DPLL is fully digital circuit, it can be easily integrated by the CMOS technology, achieving high density integration and it offers multi-function, such as multi-mode as in analog PLL's [1].

Fully-digital DPLL is very slow compared to analog one, because digital operations such as addition or subtraction take a long time. A hybrid PLL [2] which employed a parallel sampling PC overcome this problem to some extent using multi-phase clocks generated by a delay-line and showed good characteristics.

This paper describes a design of an improved CMOS 1st-order delay-line DPLL for clock regeneration application. The 1st-order DPLL has fast pull-in characteristics and is easy to design. We have employed a parallel architecture PC and improved its speed by reducing sampling stages and circuit complexity and adopted a DCO which eliminated timing hazard encountered in the design [3]. And we also laid it out in 1.2μm CMOS technology, and simulated its performance by SPICE as well as logic simulation.

2. First-Order DPLL

Figure 1 shows a block diagram of ordinary DPLL. The DPLL contains PC (Phase Comparator), LPF (Low Pass Filter) and DCO (Digitally Controlled Oscillator). Being different from ordinary analog PLL, DPLL has a DCO instead of VCO (Voltage Controlled Oscillator) and PC processes phase comparison in digital domain, namely subtraction of binary coded data of input and DCO signals.

Figure 2 shows a detailed block diagram and signal flow chart of ordinary 1st-order DPLL. The 1st-order DPLL is simple and it has fast pull-in characteristics and is easy to design.

3. Delay-Line DPLL

Fully-digital DPLL is very slow compared to analog one, because digital operations such as addition or subtraction take a long time and the reference clock is divided to supply interleaved signals. A hybrid PLL [2] overcomes this problem using multi-phase sampling by multi-phase clocks generated by a delay-line and shows good characteristics.

3.1 System Description

Figure 3 shows a block diagram of a delay-line DPLL. Delay-line generates multi-phase clock signals periodically separated by the equal to delay time of one delay cell. PC detects the position in which time slot the input signal rises among multi-phase clocks and then encodes the detected signal into binary data, and then

* Department of Electrical Enginering, Graduate Student (At present, Toshiba, Co.)

** Department of Electronic Device Engineering

* * * Department of Electrical Engineering, Graduate Student, (At present, Kawasaki Heavy Industry, Co.)

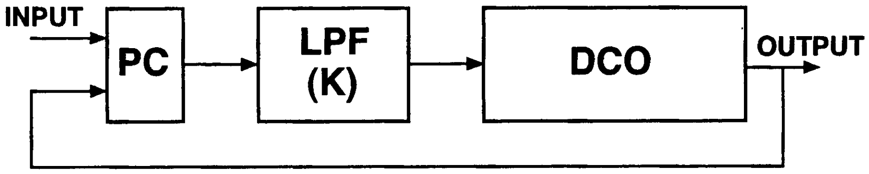


Fig.1 Block Diagram of DPLL.

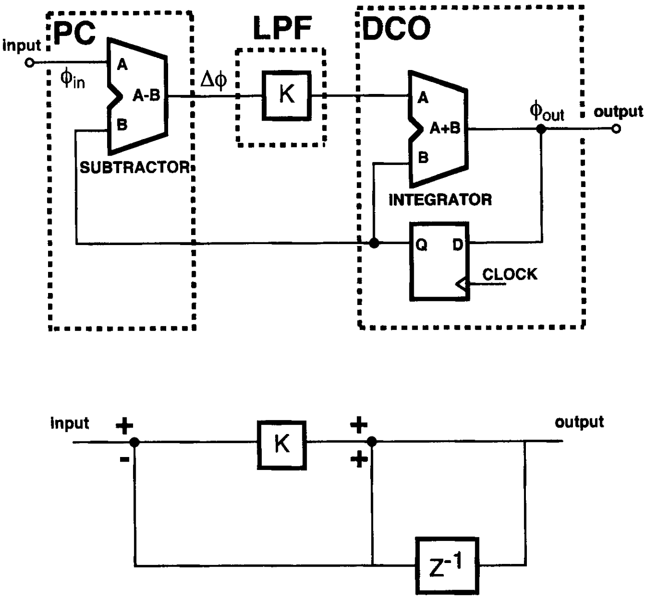


Fig.2 Block Diagram and Signal Flow Chart of 1-st Order DPLL.

subtracts DCO output data to calculate the phase difference. LPF processes filter operation digitally. In the 1-st order DPLL, LPF is a mere gain element, therefore it performs just multiplication of a constant. And DCO integrates LPF output signal and selects the most appropriate clock signal from multi-phase clocks by the integrated result and outputs the selected signal.

3.1.1 Phase Comparator

We employed parallel sampling architecture [2)] in the PC part for high speed operation. Figure 4 shows block diagram and timing chart of the parallel architecture PC. Upper half part of PC detects the position in which time slot the input signal rises among multi-phase clocks and then encodes 4-bit binary position data. The rest of PC subtracts the 4-bit data from DCO output binary data to calculate the phase difference. The PC consists of 3 stages, multi-sampling, encoder and subtraction stages.

The multi-phase sampling stage consists of two parts, multi-phase sampling part and data transfer part. The input signal is sampled by the multi-phase sampling part at each rising edge of delay-line output and latched onto one of two sets of latches, odd- or even-cycle latches, alternatively. At every even or odd clock cycle, latched data are transferred to encoder and phase error is updated. The input data are sampled by 16 D-FF's (D Flip-Flops) clocked by 16-tap delay-line, and the sampling occurs sequentially through 16 D-FF's. The next stage consists of two groups of latches, "odd-cycle" group and "even-cycle" group, which contain 16 latches each, and furthermore, the 16 latches are formed by group A and group B, each consisting of 8 latches, as shown in fig. 4. Each latch group is clocked by four interleaved clock signals, ϕ_a, ϕ_b, ϕ_c, and ϕ_d. Therefore this stage consists of four arrays of eight latches (eight D-FF's), which is half the number of delay-line taps (16). Four-phase clocks ϕ_a, ϕ_b, ϕ_c, and ϕ_d are generated to clock the four set of latches. The first 8 latches, "odd-cycle group A", are clocked by ϕ_a and second 8 latches, "odd-cycle group B", are clocked by

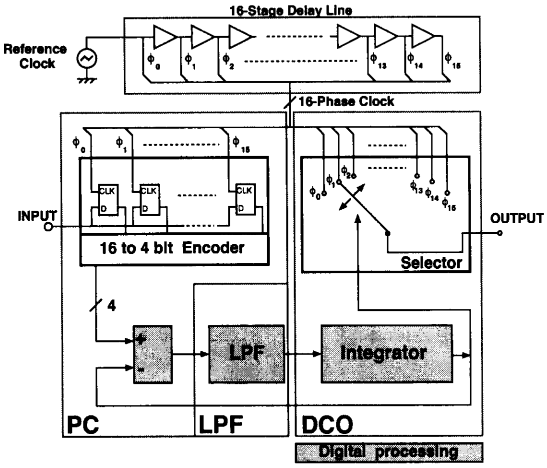
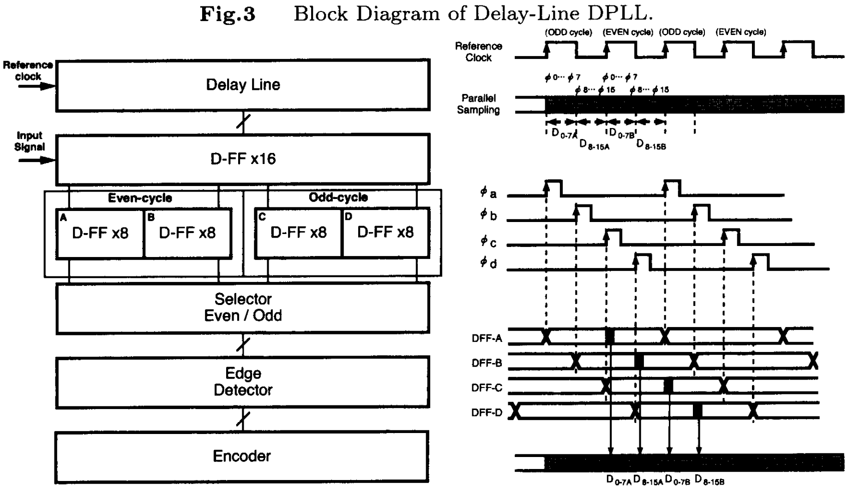


Fig.3 Block Diagram of Delay-Line DPLL.



(a) Block Diagram

(b) Timing Chart

Fig.4 Parallel Architecture PC, (a) Block Diagram and (b) Timing Chart.

ϕ_b which is delayed 8 taps of ϕ_a. And the rests of latch arrays, "even-cycle" latches, are clocked by ϕ_c and ϕ_d delayed 8 taps of ϕ_c as shown in fig. 4. Each latch array can operate at 1/4 of reference clock frequency. Hence, it is possible to overcome the speed-limitation of sequential operation. The selector stage selects the final data from "odd-cycle" group and "even-cycle" group , and reconstruct the data. Finally, the encode stage detects the rising edge position using EX-OR gates and encodes it to binary coded data by gate matrix.

3.1.2 Low-Pass Filter

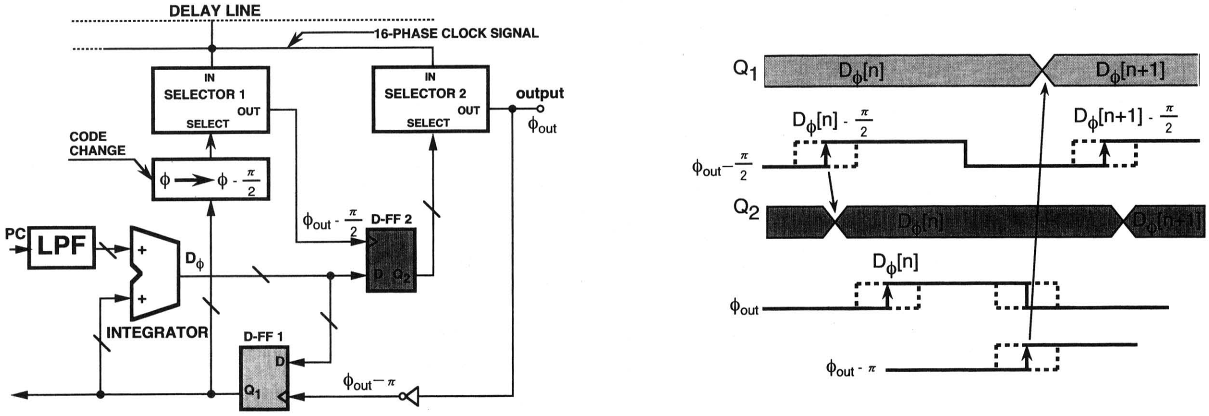
In LPF, data length is internally eight bits. LPF is a merely gain element and Gain K is a power of 2, so it just shifts 4-bit binary data of PC by a few bits to upper or lower bit side.

3.1.3 Digitally Controlled Oscillator

DCO integrates LPF binary data and selects most appropriate clock signal from 16-phase clocks supplied from delay line. DCO consists of adders to integrate LPF output signal and selectors to select the most appropriate clock signal. Since the selected clock signal is asynchronous with system clock, switching of the selector and rising of selected signal might occur simultaneously and therefore glitch might occur before or after switching period. In order to avoid this timing hazard, one more selector is added before the clock selector. Figure 5 shows block diagram and timing chart of the hazard eliminated DCO. SELECTOR 1 generates the clock signal preceding SELECTOR 2 signal by relatively $\pi/2$ in phase, by changing the code of output. D-FF 2 latches the updated selection data D_0 at $\pi/2$ before output signal is asserted. By doing this, tap selection in the SELECTOR 2 finishes before output clock is selected. And at the falling of the output, the new tap position is latched on the D-FF 1.

3.2 Circuit Implementation

It is suitable to choose the number of taps of delay-line of a power of 2. This is because of the fact that it is easy to encode the rising edge position into binary code and



(a) Block Diagram of Timing Hazard Eliminated DCO

(b) Timing Chart of Timing Hazard Eliminated DCO

Fig.5 Timing Hazard Eliminated DCO, (a)Block Diagram and (b) Timing Chart.

that multiply or divide operations of 2^n (n is an integer), are replaced by just n-bit shift operation towards upper or lower bit side. So multiplier or divider is replaced by shifter, and thus then we can reduce circuit complexity and improve the speed and reduce power consumption. Data length of PC, LPF and DCO are 4, 8 and 4 bits integer, respectively and 2's complement number representation is used internally. Therefore delay-line has 16 taps ($2^4 = 16$). In this design, the PC binary data are extended from 4-bit to 8-bit, namely 4-extra bits (all zero) are added in the LSB side to avoid underflow in integral calculation, and shifted to lower bit side. Adder is carry ripple adder to implement easily.

TSPC (True Single Phase Clock) logic style [4] is used to achieve higher speed.

4. Simulation Method

4.1 Layout

We simulated transistor level as well as functional level and gate level. In order to simulate at the transistor level, parasitic and wiring capacitances have to be considered. We used a layout software Magic developed at UCB for LSI implementation [5]. It can handle the MOSIS scalable CMOS technology [6], which is based on λ-rule design and handles single-poly, double-metal technology. And for cell library, we used UCB's "Low-Power Cell Library" [4]. By Magic, we can extract parasitic capacitances of wiring and MOS transistors. We have laid out DPLL in 1.2μm CMOS technology but have not optimized the layout.

We assumed that supply voltage is 5V. For transistor model, MOSIS 1.2μm transistor model parameters are used in SPICE simulation for keeping consistency of Magic layout system. Because of limitation of transistor model employed in the circuit simulator PSpice [7], we used level 2 MOS model. Level 2 MOS model is not suitable for sub-micron devices and shows pessimistic results, but PSpice has only four MOS models, level 1 through 4. Level 4 model is for sub-micron devices, but it is more difficult to extract parameters than in the case of other levels.

4.2 Simulation

We used PSpice [7], an analog/digital mixed circuit simulator, which can simulate digital circuits at logic level as well as analog simulation. It has standard TTL/CMOS digital library (74 series) and digital primitives and gate delay time table for them. It takes too long time to simulate DPLL at transistor level, because transient time, namely pull-in process takes usually several tens of micro-second. Therefore simulation was done by 2-step, i.e., gate delay extraction and logic level simulation.

First, all gate delays are extracted from layout results, and then the gate delay time table of the library are rewritten to simulate at logic level. To obtain gate delays from layout results, actual circuits are simulated part by part by PSpice considering parasitic capacitances. And second, the DPLL are simulated at logic level.

5. Results

Figure 6 shows the layout result of the DPLL. Layout result in 1.2μm design rule has active area of about 1.4mm × 2.0mm, and transistor count is about 5000. Figure 7 shows the simulated result for 1.2μm at 60MHz operating frequency. In the figure, lower four traces are binary coded select signal of the DCO. Since this DPLL is 1-st order one, there is steady phase error to compensate frequency difference between input and DCO signals. And since number of the delay-line taps is 16, phase error detection resolution is 2 π /16. Therefore relatively large phase error remains, and jitter characteristics are not good. To improve this characteristics,

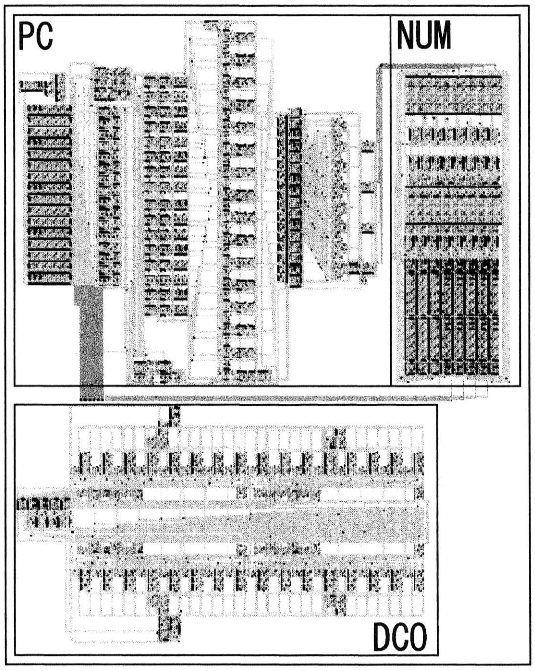


Fig.6 Layout Result of the DPLL in 1.2μm Design Rule. Area is about 1.4mm × 2.0mm.

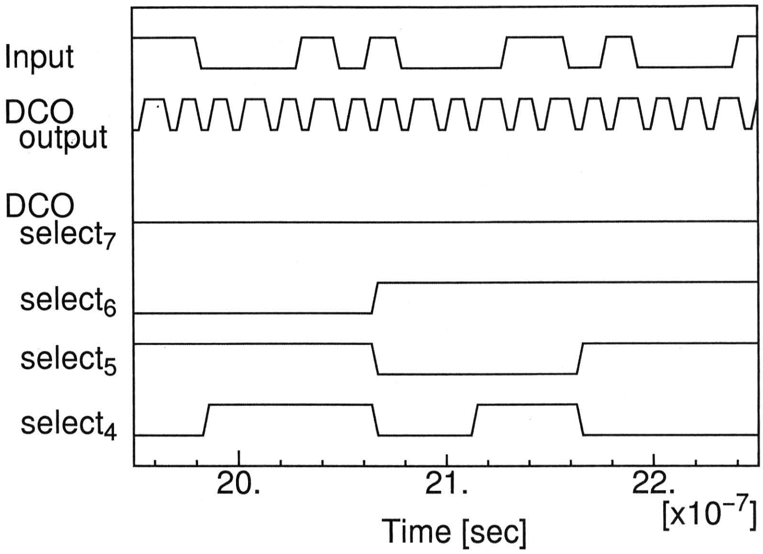


Fig.7 Simulation Result for 1.2μm Design Rule at 60MHz.

we have to increase number of delay-line taps or introduce phase extrapolater. These phase errors can be reduced within one delay-line time slot by changing the output tap of the delay-line, namely by phase shifting. Since phase error of the 1-st order PLL (DPLL) is proportional to the frequency difference between input and VCO (DCO) signals, phase shift can be done by changing output tap, detecting frequency difference or by some arithmetic operations in the DCO. And lock-in ranges are +5/-5% of DCO free-running frequency for $2^{13}-1$ PRBS (Pseudo-Random Bit Sequence) input and +5/-5% for regular pattern "10" input, respectively. When operating frequency is 60MHz and frequency difference between the input and free-running DCO signals is 1%, DPLL locks-in in a few micro-seconds.

We also simulated DPLL by original PSpice library, 74AC series components, and obtained the result showed that maximum frequency was 20MHz. The result shows that 74AC series, corresponding with 1μm technology, is slower than in the 1.2μm technology. This is because the 74AC components are discrete ones and the delay times of PSpice library are the worst case ones including parasitic and wiring capacitances in the 74AC IC's. On the other hand, in the LSI design, all circuitry is integrated on one chip, therefore parasitic and wiring capacitances are relatively smaller than 74AC series. And we also actually constructed DPLL by 74AC components and partly confirmed the operation and its performance.

6. Conclusion

In conclusion, this paper described a CMOS 1st-order delay-line DPLL for clock regeneration application. Parallel-architecture DPLL has been designed and laid-out in 1.2μm CMOS technology. Its performance has been simulated by SPICE as well as logic simulation. The results show that the designed 1.2μm DPLL can operate up to 60MHz at 5V supply and about 20mW of power consumption, and that lock-in ranges of 1.2μm DPLL are +5/-5% of DCO free-running frequency for $2^{13}-1$ PRBS input and +5/-5% for regular pattern input, respectively.

Acknowledgments

We would like to thank K. Hirose for layout support and experiment, and H. Shirahama for fruitful discussion.

References

1) H.Sato, K.Kato, T.Sase, I.Ikushima, and S.Kojima, "A Fast Pull-in PLL IC Using Two-Mode Pull-in Technique", *IEICE Trans.* Vol.J74-B-I, No.10, pp.817-824, Oct. 1991.

2) B.Kim, D.N.Helman, and P.R.Gray, "A 30-MHz Hybrid Analog/Digital Clock Recovery Circuit in 2.0μm CMOS", *IEEE Journal of Solid-State Circuits*, Vol.25, No.6, pp.1385-1394, June, 1990.

3) Ikuo Seki, "A Study on Multi-Function DPLL for Timing Regeneration", Master Thesis, Department of Electrical Engineering, Kyushu University, March, 1995.

4) U.C. Berkeley Low-Power Cell Library, and Tom Burd, "Low-Power CMOS Library Design Methodology", M.S. Thesis, ERL M94/89, University of California at Berkeley, 1994.

5) Magic (Ver.6.3), 1990 DECWRL/Livermore Magic Release, WRL Research Report 90/7, DEC WRL, 1990, and J.Ousterhout, et al., *Proc. 21st Design Automation Conference*, pp.152-159, 1984.

6) MOSIS Scalable CMOS Design Rules, Jen-I Pi, the MOSIS Service, 1995.

7) PSpice (Ver.5.4), User's Manual, MicroSim Corp., 1989.