A DCO Compiler for All-Digital PLL Design

Ching-Che Chung, *Member IEEE*, Chen-Han Chen, and Chi-Kuang Lo
Department of Computer Science and Information Engineering
National Chung Cheng University
No. 168 University Rd., Min-Hsiung, Chia-Yi, Taiwan
Email: wildwolf@cs.ccu.edu.tw

***Abstract*—In this paper, we present a digitally controlled oscillator (DCO) compiler for reducing design turn around time for an all-digital phase-locked loop (ADPLL). According to user input specifications, the DCO compiler can generate the DCO netlist implemented with standard cells and the command scripts for automatically placement and routing. The proposed DCO compiler uses liberty timing files (.lib) to compute the cell delay, and the proposed DCO regular placement approach makes it easy to predict the wire capacitance of the DCO. As a result, the frequency range of the DCO can be estimated accurately. Furthermore, a frequency estimation algorithm is presented to reduce the lock-in time of the ADPLL. The proposed ADPLL and the DCO compiler are simulated in 90nm CMOS process. In post-layout simulation, the maximum frequency range estimation error of the proposed DCO compiler is 6.65% with process, voltage, and temperature variations. In addition, the frequency error of the proposed fast settling ADPLL can be smaller than 2% within four clock cycles with different input multiplication ratio.**

Keywords—Digitally controlled oscillator, all-digital phase-locked loop (ADPLL), fast lock-in.

I. INTRODUCTION

As the explosive growth in complexity of integrated circuits, system-on-a-chip (SoC) had become a design trend for increasing the operating frequency of the integrated system. Phase-locked loops (PLLs) are widely used in the SoCs for generating the on-chip high speed clock which frequency and phase are related to the input reference clock. Traditionally, PLLs are designed by analog charge-pump based architectures. However, in advanced CMOS process, the leakage current of transistors becomes dramatically increasing which causes ripples on the control voltage of the voltage controlled oscillator (VCO). In addition, at low supply voltage, the limited voltage headroom also makes it difficult to design a VCO. In contrast to analog charge pump-based PLLs, all-digital phase-locked loops (ADPLLs) [1]-[8] consists of all digital building blocks which have no passive components. The VCO is replaced by a digitally controlled oscillator (DCO), and the digital loop filter is applied in the ADPLL. The ADPLLs implemented with standard cells are easily migrated to different process and are more suitable for low-voltage SoC design than analog charge pump-based PLLs.

A DCO occupies most portions of the ADPLL in terms of area and power consumption than the other blocks. For the ADPLL, DCO is the most critical component which determines the jitter performance and the output frequency range. In various applications require different frequency ranges, an automatic design flow for the DCO is demanded for easily migration and reducing the design turn around time [4],[5]. In [5], the DCO is transformed into the equivalent model. Thus, the DCO timing analyzer can compute the frequency range of the DCO using the timing information of liberty timing files (.lib). However, the non-linearity of the DCO [5] causes a large frequency range estimation error. Therefore, many SPICE circuit simulations are required to perform the calibration on the DCO timing analyzer. Moreover, the wire capacitance between the delay cells is not included in the DCO timing analyzer [5].

As the process technology is scaling down to the deep submicron CMOS process, the interconnection delay has great impact on the total delay calculation. Thus, prior DCO compilers [4],[5] without considering the wire capacitance will result in large frequency range estimation errors as compared to the post-layout SPICE circuit simulation results. In this paper, the relative locations between delay cells during the automatically placement and routing (APR) are controlled by the proposed DCO regular placement approach. Moreover, the proposed DCO compiler uses a linear DCO architecture and we also uses liberty timing files (.lib) to calculate the cell delay with wire capacitance. As a result, the maximum frequency range estimation error of the proposed DCO compiler is 6.65% as compared to the DCO post-layout simulation results under process, voltage, and temperature (PVT) variations.

Dynamic voltage and frequency scaling (DVFS) is often used in the battery-powered system to increase the system operating time. To quickly switch between the low-power modes and the normal operation mode, the on-chip clock generators which designed by PLLs should have a fast settling time. Therefore, a frequency estimation algorithm is proposed in ADPLL [1] to quickly find out the target DCO control code. However, the period ratios between the DCO and the reference clock are computed by frequency counters. As a result, the integer values of the period ratios cause considerable quantization errors during the calculation of the target DCO control code. To reduce the frequency error, the ADPLL [1] needs to maintain a frequency multiplication ratio larger than 45.

The frequency estimation algorithm which uses fixed-point values of the period ratio is proposed in [3]. It can reduce the quantization error during the calculation of the target DCO control code. However, in order to calculate the target DCO control code, the DCO needs to be oscillated at two extreme frequencies (i.e. maximum frequency and minimum frequency). When the maximum output frequency is over the speed

This work was supported in part by the Ministry of Science and Technology of Taiwan, under Grant NSC102-2221-E-194-063-MY3.

limitation of the DFFs due to PVT variations, the frequency finder [3] may not work correctly. Thus, in this paper, we proposed the modified frequency estimation algorithm to overcome this problem. In addition, the lock-in time is four clock cycles in the proposed ADPLL.

The rest of the paper is organized as follows: Section II describes the proposed DCO regular placement approach. Section III describes the fast lock-in ADPLL with the DCO compiler. Section IV shows the experimental results. Finally, Section V concludes with a summary.

II. DCO Regular Placement Approach

A. Wire Load Delay

TABLE I. WIRE LOAD MODEL IN LIBERTY TIMING FILES

Wire-loads	Annotation

Wire_load("wl001"){
resistance: 8.8e-8 ;
capacitance: 1.3e-4 ;
area: 0.6 ;
slope: 67.678 ;
fanout_length (1, 67.678);
fanout_length (2, 90.237);
fanout_length (3, 135.356);
}
	The wire load model name
	Resistance per unit length
	Capacitance per unit length
	Area per unit length
	Parameter of slope
	Fanout length model

The interconnection delay between delay cells is caused by the wire capacitance and wire resistance. To calculate the period of the DCO output clock accurately, the wire load delay should be included in the delay calculation. The liberty timing files (.lib) provides the wire load delay model for the synthesis tools to roughly estimate the wire load delay. Table I shows an example wire load model. According to the attributes in the wire load model, the fanout number of the wire can be used to predict the wire length of the wire. For example, if the fanout number of a wire is 2, and then, the average wire length looked up from Table I is 90.237. Subsequently, the wire capacitance is 90.237×1.3e-4, and the resistance of the wire is 90.237×8.8e-8. Thus, the estimated wire load can be added to the delay calculation. However, the wire capacitance looked up from the wire load model for pre-layout DCO circuit design is not accurate. As a result, the conventional wire load model is not suitable for the DCO compiler which needs to estimate the DCO frequency range accurately.

The precise wire length can be obtained after automatically placement and routing (APR). The gate-level RC extraction tools can extract the wire load capacitance and resistance of the routed design into a standard parasitic exchange format (SPEF) file. In this paper, the relative locations between delay cells of the DCO during APR are controlled by the proposed DCO regular placement approach. Thus, the proposed DCO compiler will use the post-layout wire load information listed in the SPEF file for the DCO frequency range estimation.

B. Regular Placement Approach

The DCO [3] that has high resolution and good linearity is adopted in the proposed DCO compiler. It is composed of a coarse-tuning stage and a fine-tuning stage, as shown in Fig. 1. The coarse-tuning stage is composed of n coarse-tuning delay cells (CDCs). In addition, the fine-tuning stage [3] is composed of two parallel connected tri-state buffers which are operating as an interpolating circuit to enhance the resolution of the DCO.

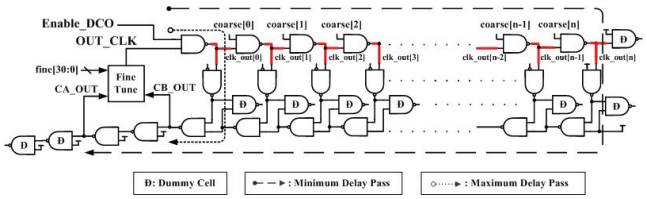

Fig. 1. The proposed DCO architecture.

The wire lengths between the delay cells of the DCO are determined by the APR tools. Once the area constraints and the cell placement results are changed, the routing results will be changed accordingly. Hence, the DCO regular placement approach is presented to avoid the large wire length variations in different DCO placement cases.

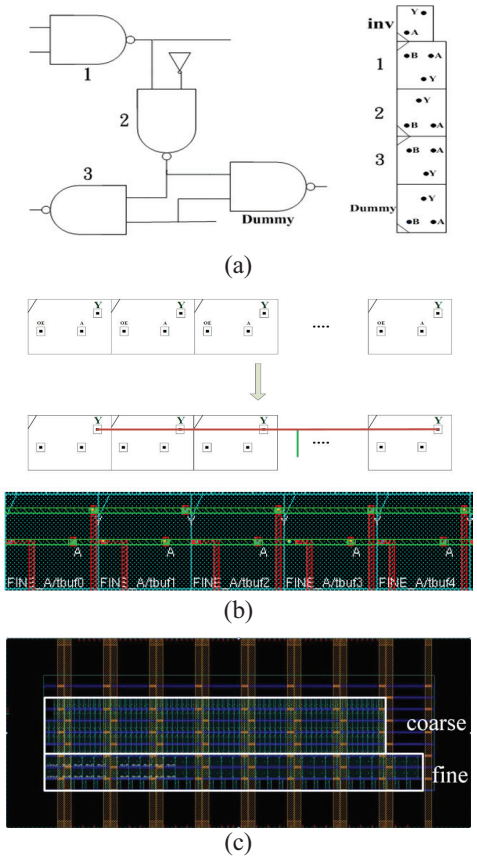

Fig. 2. (a) The regular placement of CDCs. (b) The regular placement of fine-tuning delay cells and the routed result. (c) Overall DCO placement result.

Fig. 2 (a) shows the regular placement of the coarse-tuning delay cells. In the proposed approach, each CDC occupies five standard cell's rows vertically. When CDCs are placed closer, the wire length between the pins of the delay cells becomes shorter. Besides, the wire length between two neighboring CDCs can be controlled almost the same which also improves the differential nonlinearity (DNL) of the coarse-tuning stage.

Moreover, the wire length between two neighboring CDCs will be kept the same when the number of the CDCs is increased or decreased. Therefore, the wire load between two neighboring CDC becomes more predictable. The tri-state buffers of the fine-tuning stage are placed horizontally in two standard cell's rows, as shown in Fig. 2 (b). Thus the output Y pins of tri-state buffers can be routed more regularly because all Y pins at the same standard cell row has the same y coordinate which makes the routed output net of the fine-tuning stage can be a simple straight line.

III. Fast Lock-in ADPLL with DCO Compiler

A. Automatic DCO Generation

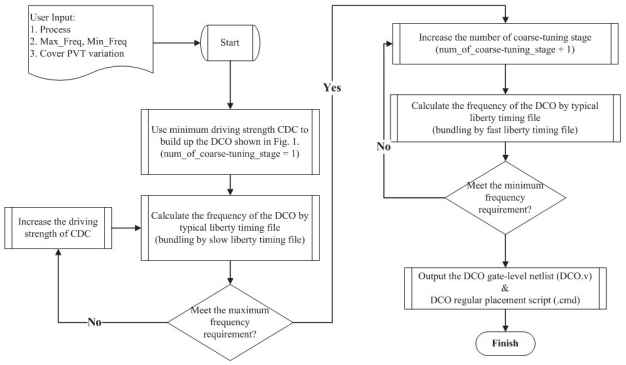

Fig. 3. The flowchart of the proposed DCO compiler.

The flowchart of the proposed DCO compiler is shown in Fig. 3. In different process, we need to perform a DCO regular placement once to obtain the wire load between CDCs and the output net of the fine-tuning delay cells, so that the calculation for the generated DCO output frequency range can be more accurately. After that, when the user inputs the specifications for the DCO (i.e. the frequency range), The proposed DCO compiler will use the non-linear delay model of the liberty timing files (.lib) and the wire load information obtained from the initial DCO regular placement result to calculate the frequency range of the target DCO. The frequency range estimation of the DCO with liberty timing files (.lib) is described in [4]. In addition, the best case liberty timing file and the worst case liberty timing file are applied for the calculation of the minimum output frequency and maximum output frequency, respectively. Thus, the PVT variations can be included in the proposed DCO compiler. Finally, the DCO gate-level netlist and the DCO regular placement scripts for the APR tools are generated.

B. Frequency estimation algorithm

The proposed frequency estimation algorithm (modified from [3]) uses a linear equation to calculate the target DCO control code for speeding up the lock-in time of the ADPLL. For a 12-bit DCO, when the DCO control code is set to zero, the period of the DCO is at its maximum value (P_{max}). Similarly, when the DCO control code is set to 1023, the period of the DCO is at its median value (P_{mid}). R_{min} and R_{mid} mean that the period ratios between the period of the DCO output and the period of the reference clock when the DCO operates at P_{max} and P_{mid}, respectively. Moreover, W_{max} and W_{mid} are the reciprocal of R_{min} and R_{mid}, respectively. If the frequency multiplication factor is M, the target period ratio (R_T) should be M after ADPLL is locked. Thus, the target W value (W_T) is the reciprocal of R_T and is 1/M. W curve is close to a straight line, as shown in Fig. 4. Thus, we can obtain the linear equation for the W curve from W_{max} and W_{min}. Subsequently, the initial DCO control code to produce the target W value (W_T) can be computed within four clock cycles.

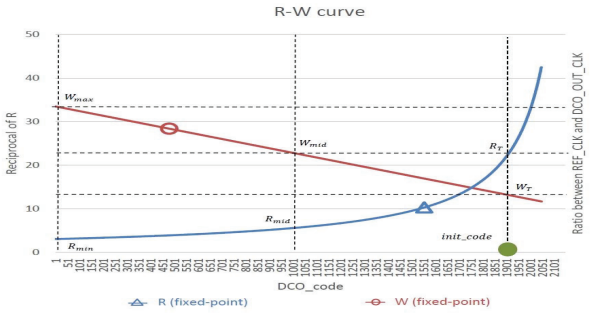

Fig. 4. The relationship between the period ratio and the DCO control code.

IV. Experimental Results

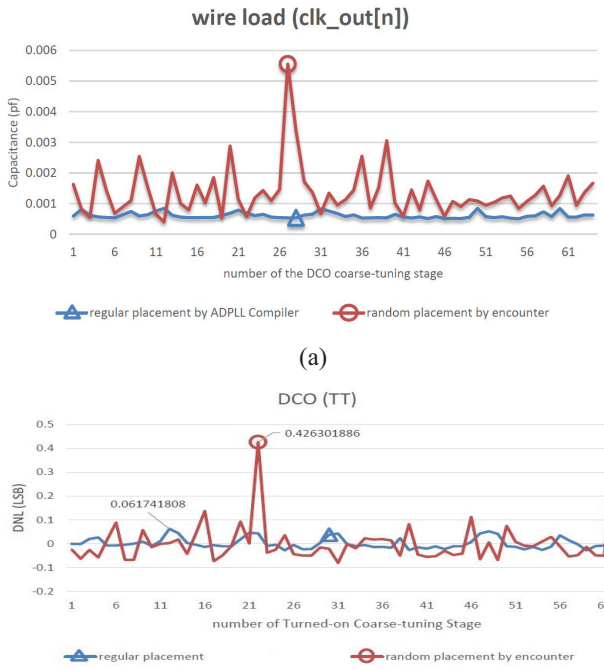

Fig. 5. (a) Wire load capacitance between the coarse-tuning delay cells. (b) DNL of the coarse-tuning stage.

The proposed DCO compiler and the fast lock-in ADPLL are implemented in 90nm CMOS process. Fig. 5(a) shows the wire load capacitance of clk_out[n] nets, as indicated in Fig. 1. The chip area of the DCO is 85×35μm^2. As compared to the random DCO cell placement by the APR tools, the proposed DCO regular placement approach makes the wire load capacitance in each clk_out[n] net very similar. Thus, in the

proposed DCO compiler, we can use the average wire capacitance value obtained from the initial DCO regular placement result to calculate the frequency range of the generated DCO.

Fig. 5(b) shows the DNL of the coarse-tuning stage. As compared to the random DCO cell placement by the APR tools, the proposed DCO regular placement approach reduces the maximum DNL from 0.426 LSB to 0.062 LSB. In addition, the non-linearity of the fine-tuning cells rather than the wire load dominates the linearity of the fine-tuning stage. As the result, the maximum DNL of the fine-tuning stage are both 0.18 LSB in the proposed DCO regular placement approach and the random DCO cell placement by the APR tools.

TABLE II. FREQUENCY RANGE ESTIMATION ERROR IN 90NM CMOS

Type of CDC	Frequency Estimation Error	
CLKNAND2X4	FF	-4.01% ~ +6.65%
	TT	-4.14% ~ +4.97%
	SS	-2.88% ~ +5.01%
CLKNAND2X8	FF	-4.33% ~ +5.55%
	TT	-4.54% ~ +4.10%
	SS	-2.89% ~ +5.23%
CLKNAND2X12	FF	-5.47% ~ +4.57%
	TT	-5.68% ~ +3.41%
	SS	-4.56% ~ +3.90%

Table II shows the frequency range estimation error of the proposed DCO compiler as compared to the DCO post-layout simulation results under PVT variations. As shown in Table II, the maximum frequency error with different type of CDC is 6.65%. Besides the wire load estimation, the accuracy of the proposed DCO compiler also depends on the accuracy of the liberty timing files (.lib).

TABLE III. PERFORMANCE COMPARISONS

	[7] ASSCC'12	**[6] TCAS2'13**	**[2] ISCAS'14**	**Proposed**
Process	180-nm	28-nm	90-nm	90-nm
Supply Voltage	1.8 V	1.0 V	1.2 V	1.0 V
Output Frequency (MHz)	2490	83 ~2000	460 ~6117	80 ~1600
Area (mm^2)	1.78 (chip size)	0.00234	0.065	0.04
Multiplication Factor	498	40	50~2000	1~63
Power Consumption	18.63mW @2.49GHz	0.64mW @2GHz	55.4mW @6GHz	0.30mW @80MHz 1.73mW @1600MHz
Lock-in Time	25 cycles	50 cycles	8 cycles	4 cycles

Table III shows the comparisons of the proposed ADPLL with recent ADPLLs. The DCO compiler generates the DCO circuit of the ADPLL which reduces the design turn around time, and the proposed frequency estimation algorithm can shorten the lock-in time of the ADPLL as compared with [2],[6],[7].

V. CONCLUSION

In this paper, we present a DCO compiler for reducing design turn around time for an ADPLL. The proposed DCO compiler uses a linear DCO architecture with the proposed DCO regular placement approach to estimate the wire capacitance between delay cells. Therefore, the proposed DCO compiler can compute the frequency range of the generated DCO with liberty timing files (.lib) accurately. The simulation results show that the proposed DCO regular placement approach reduces the maximum DNL of the coarse-tuning stage. In addition, the proposed frequency estimation algorithm shortens the lock-in time of the ADPLL within four clock cycles.

ACKNOWLEDGMENT

The authors would like to thank their colleagues in the Silicon Sensor and System (S3) Laboratory of National Chung Cheng University for many fruitful discussions. The EDA tools supported by National Chip Implementation Center (CIC) are acknowledged as well.

REFERENCES

[1] Chia-Tsun Wu, Wen-Chung Shen, Wei Wang, and An-Yeu Wu, "A Two-Cycle Lock-In Time ADPLL Design Based on a Frequency Estimation Algorithm," *IEEE Transcation on Circuits and System II: Express Briefs*, vol. 57, no. 6, pp.430 - 434, Jun. 2010.

[2] Yi-Wei Chen and Hao-Chiao Hong, "A Fast-Locking All-Digital Phase Locked Loop in 90nm CMOS for Gigascale Systems," *in Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, Jun. 2014, pp. 1134-1137.

[3] Chin-Che Chung, Duo Sheng, and Wei-Siang Su, "A 0.5V/1.0V fast lock-in ADPLL for DVFS battery-powered devices," *in Proceedings of International Symposium on VLSI Design, Automation, and Test (VLSI-DAT)*, Apr. 2013.

[4] Chin-Che Chung, Duo Sheng, and Chen-Han Chen, "An All-Digital Phase-Locked Loop Compiler with Liberty Timing Files," *in Proceedings of International Symposium on VLSI Design, Automation, and Test (VLSI-DAT)*, Apr. 2014.

[5] Chao-Wen Tzeng, Shi-Yu Huang, and Pei-Ying Chao, "Parameterized all-digital PLL architecture and its compiler to support easy process migration," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 3, pp. 621-630, Mar. 2014.

[6] Sebastian Höppner, Stefan Haenzsche, Georg Ellguth, Dennis Walter, Holger Eisenreich, and René Schüffny, "A fast-locking ADPLL with instantaneous restart capability in 28-nm CNOS technology," *IEEE Transactions on Circuits and System II: Express Briefs, vol. 59, no. 10, pp. 673-677, Oct*. 2012.

[7] Yun-Chen Chuang, Sung-Lin Tsai, Cheng-En Liu, and Tsung-Hsien Lin, "An all-digital phase-locked loop with dynamic phase control for fast locking, " *in Digest of Technical Papers, IEEE Solid-State Circuits Conference (A-SSCC), Nov 2012, pp. 297-300.*

[8] Chin-Che Chung and Chen-Yi Lee, "An all-digital phase-locked loop for high-speed clock generation," *IEEE Journal of Solid-State Circuits,* vol.38, no. 2, pp. 347-351, Feb. 2003.

[9] Dou Sheng, Chin-Che Chung, and Jhih-Ci Lan, "A monotonic and low power digitally controlled oscillator using standard cells for SoC applications," *in Proceedings of International Asia Symposium on Wuality Electronic Design (ASQED)*, Jul. 2012, pp.123-127.