An Ultra-Low Power Harmonic-Free Multiphase DLL Using a Frequency-Estimation Selector

Yi-Ming Chang, Ming-Hung Chang, and Wei Hwang
Department of Electronics Engineering & Institute of Electronics
National Chiao-Tung University, Hsin-Chu 300, Taiwan
amed.ee96g@g2.nctu.edu.tw

Abstract—This paper presents an all-digital multiphase delay-locked loop (ADMDLL) for wide-locking range and micro-power applications. To enhance locking range and locking speed of the ADMDLL, we proposed the adaptive successive approximation register-controlled (ASAR) algorithm, which uses the frequency-estimation selector (FES) to avoid harmonic lock issue. In addition, the FES can reuse the delay line to reduce circuit area and power dissipation significantly. By using the stack effect, the proposed leakage-reduced delay unit can save 12% leakage power consumption and provide good linearity versus digital control words. After locking, the dynamic frequency monitor circuit is proposed to compensate phase error caused by PVT variations. The proposed ADMDLL is capable of operating in wide supply voltage range from 0.3V to 1.0V. The power dissipation is only 520μW at 1.25GHz/1.0V, and 2.1μW at 13MHz/0.3V, respectively. This work is based on UMC 90nm standard CMOS technology.

I. INTRODUCTION

Phase-locked loop (PLL) and delay-locked loop (DLL) are widely used as de-skew buffers in microprocessors, memory interface, and communication products. Generally, in several high performance applications, such as double data rate (DDR) SDRAM [1], time-interleaved clock generator [2], clock data recovery (CDR) [3], and multi-core processors [4], multiphase DLLs (MDLLs) are often preferred due to their better performances of jitter, stability, and simple design effort than PLLs. In these applications, delay line of MDLLs has to lock a delay within one reference clock cycle.

As the need of wide operating frequency range increasing, conventional MDLLs may suffer from harmonic lock issue. Several solutions are presented in [5]-[7] to eliminate this problem. In [5], an all–analog DLL uses the replica delay cell to extend the operating frequency; however, the analog DLL is easily affected by process variations. The digital solution and the mixed-mode type DLL are presented in [6] and [7]. However, both require extra delay line as time-to-digital (TDC) circuit to select proper frequency range, for which occupies large area and dissipates more power consumption.

In this paper, a novel all-digital multiphase delay-locked loop (ADMDLL) is proposed to achieve wide-locking range, low power, and fast-lock abilities. With the aid of the proposed frequency-estimation selector, the ADMDLL can extend the locking range and accelerate the locking step. After the DLL is locked, the proposed dynamic frequency monitor circuit is enabled and calibrates static phase error caused by PVT variations. Moreover, the ADMDLL can function properly from sub-threshold to super-threshold region (0.3V-1.0V) without additional design.

An overview of the proposed ADMDLL and the novel adaptive SAR algorithm are discussed in Section II. The circuit descriptions are presented in Section III. The post-layout simulation results based on UMC 90nm CMOS technology are shown in Section IV. Finally, conclusions of this work are given in Section V.

II. ARCHITECTURE AND OPERATION PRINCIPLE OF PROPOSED DLL

The architecture of the proposed ADMDLL is shown in Fig. 1. It consists of five major blocks: they are phase detector (PD), lock-in unit (LU), digitally controlled delay line (DCDL), frequency-estimation selector (FES), and dynamic frequency monitor (DFM) circuit. Firstly, the LU activates the FES and sets the DCDL in the middle point of
delay range. Then, the FES utilizes multiphase outputs of DCDL to estimate approximate input frequency range and to generate digital codes (S1, S0). Secondly, the LU uses digital codes (S1, S0) and signals from PD to do the locking procedure of the ADMDLL. Once the DCDL is aligned with the reference clock and delay in one reference clock cycle, the ADMDLL is in the lock state. Finally, after the ADMDLL enters the lock state, the LU enables the DFM circuit to track dynamic phase error caused by PVT variations. When the phase error between ref_Clk and out_Clk is out of the locking window, the DFM generates the internal reset signal to the LU and adjusts the phase error. When the ADMDLL does the locking procedure, the DFM could turn off to save power dissipation.

Conventional DLLs locking range is limited by the following inequality [5]:

\[
\max (T_{d_{\text{min}}}, \frac{2}{3} \times T_{d_{\text{max}}}) < T_{\text{ref}} < \min (T_{d_{\text{max}}}, 2 \times T_{d_{\text{min}}}) \quad (1)
\]

where \( T_{\text{ref}} \) means the reference clock period, \( T_{d_{\text{min}}} \) is the minimum delay of DCDL, and \( T_{d_{\text{max}}} \) is the maximum delay of DCDL. In order to achieve the fast-lock ability, the proposed LU is based on successive approximation register-controlled (SAR) algorithm [8]. Conventional SAR algorithm initially sets DCDL in the half of delay range as \((T_{d_{\text{min}}}+T_{d_{\text{max}}})/2\), performing binary search. Therefore, from equation (1), the locking range of SAR-type DLLs can be expressed as

\[
\frac{1}{3} (T_{d_{\text{min}}}+T_{d_{\text{max}}}) < T_{\text{ref}} < T_{d_{\text{max}}} \quad (2)
\]

Equation (2) shows that conventional SAR-type DLLs have maximum operating range when \( T_{d_{\text{max}}} = 2 \times T_{d_{\text{min}}} \); in other word, the locking range is restricted by 2:1.

The proposed adaptive SAR strategy is using the FES to pre-obtain approximate input frequency range and generates digital codes (S1, S0) to LU. Meanwhile, the LU utilizes the codes, giving the DLL proper initial delay to avoid false locking. For example, if input frequency is high, the LU starts with shorter binary-weighted initial delay, like one-quarter delay time of the DCDL instead of always choosing in the half. In our DLL, we provide three different initial delay of DCDL: one-half, one-quarter, and one-eighth as Fig. 2 shown. Consequently, the locking range of the ADMDLL can be extended as

\[
\frac{1}{12} (T_{d_{\text{min}}}+T_{d_{\text{max}}}) < T_{\text{ref}} < T_{d_{\text{max}}} \quad (3)
\]

From equation (3), the proposed ADMDLL theoretically has maximum locking range when \( T_{d_{\text{max}}} = 11 \times T_{d_{\text{min}}} \). Fig. 2 depicts the locking range principle of proposed ADMDLL. To prevent the false locking caused by PVT variations, the locking range of the ADMDLL should be overlapped pairwise.

### III. Circuits Implementation

#### A. Frequency–Estimation Selector

The proposed frequency-estimation selector (FES) is shown in Fig. 3. Initially, 7-bit binary-weighted control words C[6:0] are set to “1000000”; therefore, the DCDL is in the center of delay range. Among the four phases of the DCDL, the FES utilizes phase 2 and 3 to sample ref_Clk/2 period to obtain approximately reference clock frequency; meanwhile, the FES generates codes (S1, S0) to the lock-in unit (LU), giving the DCDL proper initial delay. Fig. 4 shows the timing diagram of the FES.

In this work, three different frequency bands are controlled by 2-bit codes (S1, S0). For example, when the reference clock frequency is 1.25GHz, the output of the
codes (S1, S0) are (0, 0) the FES is in the “fast” mode, as shown in Fig. 4. According to Table 1, when the 2-bit control codes (S1, S0) are (0, 0), the LU starts with binary-weighted initial state “0010000”. Consequently, the DCDL initial delay is set as one-eighth of delay time to prevent harmonic locking issue.

Compared with previous research in [5]-[7], which require replica delay line as time interval measurement circuit, the proposed FES has two advantages: (1) it reuses the delay line, using only two phases clock to perform time measurement; therefore, the circuit area and power consumption are significantly reduced; (2) the FES is input duty cycle immunity since it samples two times period of reference clock as frequency measurement.

B. Lock-In Unit

The proposed 7-bit lock-in unit (LU) is shown in Fig. 5. It consists of the conventional SAR controller [8] and the adaptive decision block (ADB). The ADB receives control codes (S1, S0) from the FES and generates appropriate digital signals for D6 and D5. If the codes (S1, S0) are (0, 1) or (0, 0), the comparison of SAR search algorithm can further be reduced; hence, the locking steps accelerate as well. The rest of operation steps are basically followed by the SAR search algorithm in [8].

To ensure the LU can operate accurately in wide-locking range, the LU clock signal “clk_sar” is the reference clock cycle divided-by-4. In this work, the LU provides 7-bit resolutions to perform binary search; therefore, the total locking time of the proposed DLL is no more than 28 (7x4) reference clock cycles.

Table I. Digital Codes Versus Initial State

<table>
<thead>
<tr>
<th>Region</th>
<th>P[3:0]</th>
<th>S1, S0</th>
<th>Initial State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range 1</td>
<td>1110</td>
<td>11</td>
<td>1000000</td>
</tr>
<tr>
<td>Range 2</td>
<td>1100</td>
<td>01</td>
<td>0100000</td>
</tr>
<tr>
<td>Range 3</td>
<td>1000</td>
<td>00</td>
<td>0010000</td>
</tr>
</tbody>
</table>

C. Leakage-Reduced Delay Unit

The proposed leakage-reduced delay unit (LRDU) is shown in Fig. 6. It consists of two different types of NAND gates. One is a conventional NAND gate; the other is a half-stack NAND gate.

The binary-weighted control words are converted into the thermometer codes T0-T2. According to different thermometer codes, the LRDU have various delay time. Both intrinsic delay and a delay step are equal to two NAND gates delay time. When operating frequency increases, the number of activated NAND gates are reduced; thus, the active power can be reduced; moreover, the leakage power mitigates 12% due to stack effect [9] of the proposed half-stack NAND gate. To increase the delay resolution, the fine tune delay unit [1] is added in front of the LRDU.

D. Dynamic Frequency Monitor Circuit

Conventional SAR-type DLLs may suffer from a deadlock problem due to environmental variations after DLLs are locked. It may require external reset signal to reset clock. To detect frequency variation automatically after lock, the dynamic frequency monitor (DFM) circuit is proposed.

Fig. 7 shows the DFM circuit; it operates as an additional phase detector (PD). After the DLL achieves lock, the lock-in unit (LU) enables DFM circuit and receives the lock signal from PD. When the lock signal goes from high to low, the DLL is out of lock. The DFM generates internal pulse to reset the clock. To avoid the pulse disappearing, the pulse width should be properly designed; therefore, the delay Td is added to the DFM reset path. The total compensation procedure is no more than 28 reference clock cycles. Fig. 8 shows the timing diagram of the DFM circuit.
IV. SIMULATION RESULTS

The post-layout simulation results are based on UMC 90nm standard CMOS process. When operating in 1.0V, the locking range of the proposed DLL is 200MHz – 1.25GHz. The static phase error is 26.4ps at 200MHz, and the power consumption is 520μW at 1.25GHz; when operating in 0.3V, the locking range of the ADMDLL is 13MHz – 67MHz. The static phase error error and power consumption are 224.3ps and 2.1μW at 13MHz. Fig. 9 shows the layout view of the proposed ADMDLL, and the active area is 0.043x0.14mm². The performance summary of the proposed ADMDLL is given in Table 2.

V. CONCLUSIONS

A novel 200MHz-1.25GHz all-digital multiphase delay-locked loop (ADMDLL) is presented in this paper. With the aid of the proposed frequency-estimation selector, the ADMDLL can switch to appropriate initial delay of the DCDL to eliminate harmonic lock issue; in addition, it can reuse the delay line as time interval measurement circuit, which reduces circuit area and power consumption significantly. The novel leakage-reduced delay unit satisfies good linearity versus digital control words and can save 12% leakage power due to the stack effect. The dynamic frequency monitor circuit is implemented to track phase error after ADMDLL is locked, and the phase error can be compensated within 28 reference cycles. The power consumption is 520μW/1.25GHz at 1.0V. Also, the supply voltage can be lowered till 0.3V for the proposed DLL working correctly. The minimum power consumption is only 2.1μW/13MHz. In conclusion, the proposed ADMDLL can achieve wide-range operation, micro-power, small area, which is suitable for portable or medical device and multi-core SoC applications.

ACKNOWLEDGMENT
This research is supported by National Science Council, R.O.C., under project NSC 97-2220-E-009-027. This work is also supported by Ministry of Education, R.O.C., under the project MOEA 97-EC-17-A-03-S1-005. The authors would like to thank ITRI and Ministry of Economic Affairs for their support.

TABLE II. PERFORMANCE SUMMARY

<table>
<thead>
<tr>
<th>All-Digital Multiphase Delay-Locked Loop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
</tr>
<tr>
<td>UMC 90nm CMOS</td>
</tr>
<tr>
<td>Active Area</td>
</tr>
<tr>
<td>0.043x0.140mm²</td>
</tr>
<tr>
<td>Operating frequency range</td>
</tr>
<tr>
<td>200MHz-1.25GHz@1V</td>
</tr>
<tr>
<td><a href="mailto:13MHz-67MHz@0.3V">13MHz-67MHz@0.3V</a></td>
</tr>
<tr>
<td>Static phase error</td>
</tr>
<tr>
<td>26.4ps @ 200MHz/1V</td>
</tr>
<tr>
<td>224.3ps @ 13MHz/0.3V</td>
</tr>
<tr>
<td>Power consumption</td>
</tr>
<tr>
<td>520μW @ 1.25GHz, 1.0V</td>
</tr>
<tr>
<td>2.1μW @ 13MHz, 0.3V</td>
</tr>
</tbody>
</table>

REFERENCES


Fig. 9 Layout view of the proposed DLL

Fig. 8 Timing diagram of the DFM