A Pipeline ROM-less DDFS Using Equal-Division Interpolation

Tsung-Yi Tsai, Hsiang-Yu Shih, and Chua-Chin Wang†, Senior Member, IEEE

Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan 80424
Email: ccwang@ee.nsysu.edu.tw

Abstract—A pipeline ROM-less direct digital frequency synthesizer (DDFS) with equal division interpolation is proposed in this investigation. In order to get higher SFDR and faster clock rate, different segments with various interpolation equations are analyzed. 2nd-order parabolic equations with proper selection of coefficients based on hardware cost is utilized to transcend the limitation of SFDR. Thus, a 4-stage pipeline architecture is realized to achieve better clock speed. This work demonstrates the maximum SFDR for 102 dBc and the output frequency for 50 MHz using TSMC 0.18 μm CMOS technology cell library.

Index Terms—frequency synthesizer, DDFS, parabolic polynomial interpolation, SFDR, FCW

I. INTRODUCTION

Frequency synthesizers play an important role in many communication systems, e.g., wireless communications, mobile phones, or global positioning system (GPS). It can be used to generate signals with various frequencies. Conventionally, phase-locked loop (PLL) [1] is widely used to generate a signal with a selected frequency, where disadvantages, such as poor phase noise performance, high power consumption per frequency, slow switching speed, etc., have been found. To transcend the limitation of PLL, the first direct digital frequency synthesizer (DDFS) was proposed in 1971’s [2]. The amplitude data is stored in ROM-based look-up table, which later becomes the major drawback of this architecture.

Recently, ROM-less DDFS [3] - [5] has been developed to avoid power, area, and speed problems caused by the large ROM table. High-order (more than three) polynomials are utilized to replace the large ROM table and realize the phase-to-sin mapper. However, according to the prior work [6], the order of the polynomial can’t larger than three. Otherwise, the spurious free dynamic range (SFDR) of sine wave will be limited. In order to increase SFDR and maximum output frequency than the prior works, e.g., [7], 2nd order polynomial with equal division in a pipeline architecture is proposed in this design.

†: Prof. C.-C. Wang is the contact author.

Acknowledgement: This proposed design was partially supported by National Science Council, Taiwan, under grant MOST 104-2622-E-006-040-CC2, MOST 105-2218-E-110-006-, MOST 105-2221-E-110-058-, and the authors would like to express our deepest appreciation to CIC (Chip Implementation Center) in NARL (Nation Applied Research Laboratories), Taiwan, for the assistance of thoughtful chip fabrication.

II. PIPELINE DDFS DESIGN WITH EQUAL-DIVISION INTERPOLATION

A. Interpolation scheme selection

Usually, a quarter of the full sine wave is equally divided into several parts, where each part is approximated by equations. Then, by taking advantage of the symmetry of the sine wave, the other three quarters can be synthesized into a complete full cycle waveform. Therefore, the selectivity of interpolation scheme will directly affect the maximum error and SFDR. Three common interpolation methods, e.g., linear interpolation, quasi-linear interpolation, and parabolic interpolation, with different segmentation scenarios is compared by MATLAB simulation and summarized in Table I. The SFDR and maximum error results of parabolic interpolation with 8 segments is better than that of quasi-linear interpolation with 32 segments. Besides, if the design complexity, area overhead, and switching speed are taken into consideration, the parabolic interpolation with 8 segments is a better option.

<table>
<thead>
<tr>
<th>Segment</th>
<th>Parabolic interpolation</th>
<th>Quasi-linear interpolation</th>
<th>Linear interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>(4.97 \times 10^{-4})</td>
<td>86</td>
<td>106</td>
</tr>
<tr>
<td>8</td>
<td>(6.30 \times 10^{-5})</td>
<td>97</td>
<td>38</td>
</tr>
<tr>
<td>16</td>
<td>7.92 \times 10^{-6}</td>
<td>123</td>
<td>96</td>
</tr>
<tr>
<td>32</td>
<td>9.97 \times 10^{-7}</td>
<td>142</td>
<td>105</td>
</tr>
</tbody>
</table>

B. Characteristic equation and pipeline selection

As mentioned in the previous subsection, the complexity can be reduced by selecting proper 2nd-order characteristic equation even though the parabolic interpolation scheme is used. Table II summarized the result of 4 different 2nd-order parabolic equations by MATLAB simulations. From the viewpoint for SFDR, the missing coefficient "b" in equation #2 makes coefficient "a" hard to fit both curvature and slope at the same time. On the other hand, equation #2 has the least complexity, while equation #1 is the most complicated one because one more multiplexer is needed. Last but not least, according to the specification of 32-bit frequency control word.
(FCW) and 24-bit output resolution, we choose equation #4 to realize our design.

According to the previous analysis, the schematic of our 4-stage pipeline DDFS is shown in Fig. 1. If only Register 1 is inserted after the adder, this pipeline 2-stage system achieves only 62 MHz of the maximum clock rate. The reason is that the data processing after Register 1 takes a long time to complete. Besides, the multiplexer is found to be the bottleneck of the speed such that Register 2 and 3 are inserted before and after the multiplexer, respectively, to boost the clock rate up to 100 MHz. The speed has been increased 60% for only 12% area overhead by this design.

III. Simulation Results

This work is realized by TSMC 0.18μm Mixed signal CMOS process. Fig. 2 shows the layout of this work, where the core area is only 0.209×0.209 mm², and the overall chip size is 1.15×1.15 mm². Table III is the performance comparison with prior ROM-less DDFS designs. Our design attains the highest SFDR. Table III. Table II

**DIFFERENT PARABOLIC EQUATION COMPARISON**

<table>
<thead>
<tr>
<th>Equation</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a x^2 + b x + c)</td>
<td>24-bit ADD</td>
<td>24-bit ADD</td>
<td>21-bit ADD</td>
<td>11-bit ADD</td>
</tr>
<tr>
<td>(a(x + b)^2 + c)</td>
<td>14×14 MUL</td>
<td>17×17 MUL</td>
<td>11×14 MUL</td>
<td>11×14 MUL</td>
</tr>
</tbody>
</table>

*Control factor is 32-bit FCW and 24-bit output resolution
*ADD=add, MUL=multiplexer

**TABLE III PERFORMANCE COMPARISON OF DDFS**

<table>
<thead>
<tr>
<th>CFW</th>
<th>Output Resolution</th>
<th>Power (mW/MHz)</th>
<th>Normalized Area (x10^6 mm²)</th>
<th>SFDR (dBc)</th>
<th>Verification</th>
<th>Clock (MHz)</th>
<th>Max. Output Freq. (MHz)</th>
<th>FOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 bits</td>
<td>24 bits</td>
<td>0.31</td>
<td>11.85</td>
<td>68</td>
<td>Sim.</td>
<td>50</td>
<td>25</td>
<td>8324</td>
</tr>
<tr>
<td>9 bits</td>
<td>12 bits</td>
<td>1.07</td>
<td>18.93</td>
<td>60</td>
<td>Meas.</td>
<td>650</td>
<td>N/A</td>
<td>957</td>
</tr>
<tr>
<td>17 bits</td>
<td>8 bits</td>
<td>0.186</td>
<td>1.20</td>
<td>55</td>
<td>Sim.</td>
<td>1000</td>
<td>N/A</td>
<td>1150</td>
</tr>
<tr>
<td>32 bits</td>
<td>10 bits</td>
<td>0.027</td>
<td>0.52</td>
<td>24</td>
<td>Sim.</td>
<td>100</td>
<td>N/A</td>
<td>15891</td>
</tr>
</tbody>
</table>

*Sim=Simulation, Meas.=Measurement

**REFERENCES**


