Switching Activity Analysis of Shifters and Multipliers for Application to ROM-less DDFS Architecture Selection for Low Power Performance

Chua-Chin Wang, Senior Member, IEEE
Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan 80424
ccwang@ee.nsysu.edu.tw

Hsiang-Yu Shih
Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan 80424

***Abstract*— A dynamic power estimation method for critical logic cirucits used in ROM-less direct digital frequency synthesizer (DDFS) designs is proposed, including the analysis of switching activity of adders, shifters, and multipliers. Most important of all, the analytic solutions of the logic circuits' bit switching activity are dervied, respectively. As soon as the partitions of the $\frac{\pi}{2}$ and the polynomial interpolation equations are given, the proposed method is able to assess the overall switching activities such that the power profile of different combination of partitions and equations can be predicted before any physical implementation using logic circuits. The accuracy of the power dissipation profile with respect to the estimation using the state of art is increased by 10∼78%.**

***Keywords*—ROM-less DDFS, polynomial interpolation, dynamic power estimation, bit switching activity, spurious free dynamic range (SFDR)**

I. Introduction

Frequency synthesizer is a very critical sub-system to generate signals with a selective frequency in various applications, e.g., portable devices, CDR (clock and data recovery), GPSs (Global Positioning Systems), etc. Phase-locked loop (PLL) has been widely used in the realization of the frequency synthesizer [1]- [2]. PLL-based frequency synthesizers were also been criticized by the intrinsic slow frequency switching speed and poor spectral purity [3]. Therefore, the PLL-based frequency synthesizers become questionable when they are expected to meet the demand of fast frequency switching. DDFS proposed in 1971's [4], where the amplitude data are stored in ROM-based look-up table, has been considered as another possible solution to carry out frequency synthesizers. The samples of sine wave amplitude are stored in the ROM-based look-up table, which are accessed on demand and then converted into digital pattern sequences by the amplitude complementor. Finally, a digital-to-analog converter (DAC) converts the digital patterns into an analog sine wave signal. Apparently, this kind of DDFS is suffered from large chip area, long access time, and huge power dissipation caused by the ROM table.

By contrast, the ROM-less DDFS demonstrates many advantages, including high spurious free dynamic range (SFDR), design felxibilty, and low area cost [5], [6]. The major difference of this approach is to replace the ROM-based look-up tables with high-order polynomial algorithms in the realization of the phase-to-amplitude converters (PAC). However, any algorithm based on high-order polynomials leading to very complicated hardware implementation will not provide high speed output signals. For instance, According to the statement of the prior work [8], any polynomial with order higher than 3 may be inefficient to attain high SFDR. Besides, power consumption has become the major concern for many recent portable applications. Lopelli *et al.* managed to propose a power-consumption estimation method for ROM-based DDFS designs [7]. However, regarding the power estimation for ROM-less DDFS designs, none of any effective method was proposed yet. Notably, the most power-consuming blocks of ROM-less DDFS circuits are multipliers and shifters [9], since these 2 blocks are the most computation intensive units. Therefore, analytic solutiona to estimate the power consumption of these two blocks is disclosed in this work besides that of the adder. The result of the proposed theory is extremely helpful to evaluate the power consumption profile before a ROM-less DDFS design is realized on silicon.

II. Switching Activity Analysis of Shifter and Multiplier in ROM-less DDFS

A. Switching activity assumption

The major power consumption of logic circuits is dominated by dynamic power, which is governed by $P_{dyn} = \alpha \cdot f \cdot C \cdot V^2$, where f is the system clock, C is the area or capacitance, V is the system supply voltage, and α denotes the switching activity of $0 \rightarrow 1$ [9]. Since the system clock and supply voltage can be assumed the same for any DDFS implementation and C is basically a random variable hard to be formulated, α becomes the only factor for dynamic power estimation. Therefore, we

Prof. C.-C. Wang is the contact author. He is with Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan 80424. (e-mail: ccwang@ee.nsysu.edu.tw).

978-1-5386-5251-0/18/$31.00 ©2018 IEEE

take advantage of calculating the number of bit switching, i.e., (bit length) $\times$ (probability of $0 \rightarrow 1$) to assess the power consumption.

B. Switching activity analysis of shifters

As addressed earlier, the shifter is one of the most computation intensive logic circuits in ROM-less DDFS designs. Certainly, it is also deemed as a computation intensive logic block in other digital circuit designs. Fig. 1 is an illustrative example of 4-bit logic right shifter composed of 2 layers of MUXs (multiplexers) to demonstrate how the switching activity is analyzed.

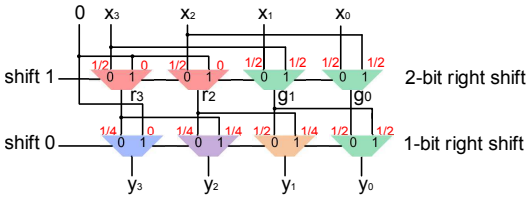

Fig. 1. 4-bit logic right shifter

Assume the probability of every input of the shifter to be "1", namely Prob(1), is 1/2. x_i, $i = 3, 2, 1, 0$, are the data to be shifted, and y_i, $i = 3, 2, 1, 0$, are the data to be output. r_3, r_2, g_1, g_0 are the outputs of the top layer MUX, respectively. By simple mathematical derivation, the Prob(1) of r_3, r_2, g_1, g_0 is found to be 1/4, 1/4, 1/2, 1/2, respectively. Thus, if the 2-to-1 MUXs are classified by the combination of Prob(1) of the inputs, there are a total of 5 types.

1). red MUXs : Prob(1) of 2 inputs are (1/2, 0)
2). green MUXs : Prob(1) of 2 inputs are (1/2, 1/2)
3). blue MUXs : Prob(1) of 2 inputs are (1/4, 0)
4). purple MUXs : Prob(1) of 2 inputs are (1/4, 1/4)
5). yellow MUXs : Prob(1) of 2 inputs are (1/2, 1/4)

As soon as we are aware of Prob(1) of each internal node of the shifter, the switching activity, i.e., Prob(0→1), can be derived as well. It turns out that the switching activity of MUXs in this kind of shifter is either 5/8 or 7/8. Those MUXs with 5/8 switching activity is located at the lower left side of the shifter, while the rest of the MUXs are with 7/8 switching activity. Fig. 1 is degenerated to be Fig. 2, where the yellow MUXs attain 5/8 switching activity, while the green MUXs have 7/8.

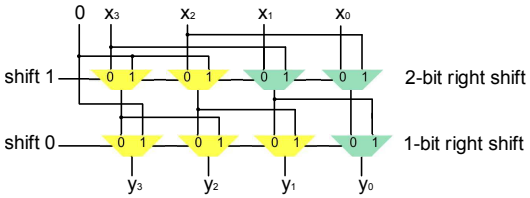

Fig. 2. Degenerated 4-bit logic right shifter

By the similar thought, the switching activity of an n-bit shifter can be analyzed as well. First of all, the number of MUX layers is found as follows.

$$\text{stage} = \text{ceil}\left(\log_2\left(\text{n}\right)\right) \tag{1}$$

where ceil( ) is the ceiling function. Notably, the number of yellow MUXs in each layer is close to a geometric series. For instance, the first (top) layer has 2, the second layer has 2 + 1, and so on. The ratio is 1/2 in this series. In summary, the number of yellow MUXs in the f-th layer is concluded as follows.

$$\text{S}\left(\text{f}\right) = \sum_{\text{w}=1}^{\text{f}} 2^{(\text{stage}-1)} \cdot \frac{1}{2}^{(\text{w}-1)} \tag{2}$$

The total bit switching number of the f-th layer can be derived by 2 different scenarios : the number of inputs is larger than that of the yellow MUXs in the same layer or not.

case 1). If the number of inputs is smaller than that of the yellow MUXs in the same layer, then all the MUXs are "yellow" type.

case 2). If the number of inputs is larger than that of the yellow MUXs in the same layer, then the extra MUXs are "green" type.

The above observation leads to the formulation of the following equations, where $\text{P}_{\text{shifter}}\left(\text{f}\right)$ denotes the overall bit switchings of the f-th layer.

$$\text{P}_{\text{shifter}}\left(\text{f}\right) = \begin{cases} \frac{5}{8} \cdot n & \text{if } n \leq \text{S(f)} \\ \frac{5}{8} \cdot n + \frac{7}{8} \cdot (n - \text{S(f)}) & \text{if } n > \text{S(f)} \end{cases}$$

Thus, the overall bit switchings of n-bit shifter is summarized as follows, where "stage" stands for the number of layers defined in Eqn. (1).

$$\text{P}_{\text{shifter}}\left(\text{n}\right) = \sum_{\text{f}=1}^{\text{stage}} \text{P}_{\text{shifter}}\left(\text{f}\right) \tag{3}$$

C. Switching activity analysis of array multipliers

Although array multipliers are not as fast as other types of fast counterparts, e.g., booth multipliers, it has the advantage of regularity such that the bit switching activity is predictable. Fig. 3 is a simple 4-bit array multiplier for illustration of the switching activity therewith. Besides conventional logic gates, e.g., AND, HA (half adder) and FA (full adder), the core of the array multiplier consists of MFA (modified full adder) and MHA (modified half adder). An interesting and obvious feature of the array multiplier is that the distribution of those mentioned logic blocks are quite regular, which can be easily derived by commercial software tools, e.g., MATLAB. A total of the n^2 AND gates are needed in the multiplier, including those placed in the black dashed box at the upper left corner and those utilized in MFAs and MHAs. MHAs in the red dashed box right below the top AND gates have inputs $(A_{(i-n+1)}, A_i)$. Those MFAs in the green dashed box are driven by inputs of $(S_{(i-n+1)}, A_i, C_{(i-n)})$. By the similar observation, those in the purple dashed box are driven with $(S_{(i-n+1)}, C_{(i-n)})$, those in the yellow dashed box are

$(S_{(i-n+1)}, C_{(i-n)}, C_{(i-1)})$, those in the blue dashed box are $(A_{(i-n+1)}, A_i, C_{(i-n)})$, and those in the brown dashed box are $(A_{(i-n+1)}, C_{(i-n)}, C_{(i-1)})$, where A_i, C_i, S_i are the generated addends, carry out, and sum, respectively, of each block. Although the number of the blocks in Fig. 3 is $n(n+1)-1$, the blocks left to be derived with respect to the switching activity is MFA and MHA. The reason is the switching activities of AND, HA and FA, are well known.

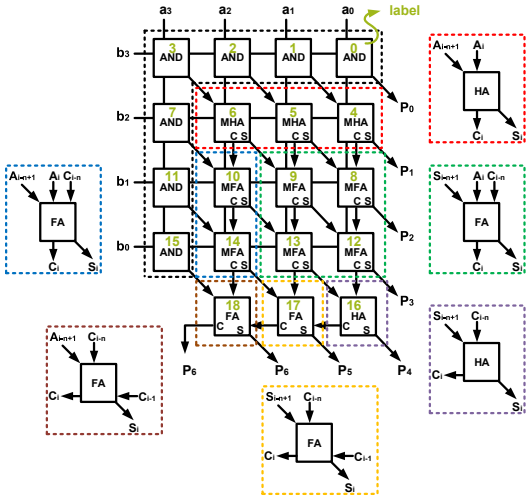

Fig. 3. 4-bit array multiplier

• analysis of MFA : Referring to Fig. 4, assume the probability of logic 1 at every input is 1/2. Thus, the following switching probability at each node is derived.

1). $P_{net1} = Prob(0) \cdot Prob(1) = \frac{3}{4} \cdot \frac{1}{4} = \frac{3}{16}$
2). $P_{net2} = Prob(0) \cdot Prob(1) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$
3). $P_{net3} = Prob(0) \cdot Prob(1) = \frac{3}{4} \cdot \frac{1}{4} = \frac{3}{16}$
4). $P_{net4} = Prob(0) \cdot Prob(1) = \frac{7}{8} \cdot \frac{1}{8} = \frac{7}{64}$
5). $P_S = Prob(0) \cdot Prob(1) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$
6). $P_{Cout} = Prob(0) \cdot Prob(1) = \frac{5}{8} \cdot \frac{3}{8} = \frac{15}{64}$

In summary, the total switching activity of an MFA is the summation of all the above numbers, which is $\frac{39}{32}$.

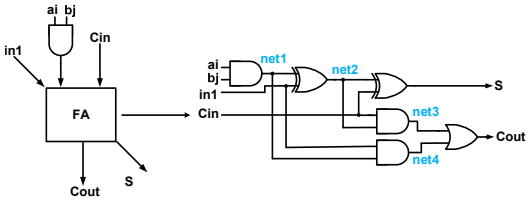

Fig. 4. MFA

• analysis of MHA : The derivation of MHA is similar to that if MFA. Referring to Fig. 5, the following results are attained.

1). $P_{net5} = Prob(0) \cdot Prob(1) = \frac{3}{4} \cdot \frac{1}{4} = \frac{3}{16}$
2). $P_S = Prob(0) \cdot Prob(1) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$
3). $P_{Cout} = Prob(0) \cdot Prob(1) = \frac{7}{8} \cdot \frac{1}{8} = \frac{7}{64}$

Thus, we conclude that the overall switching activity of an MHA is $\frac{35}{64}$.

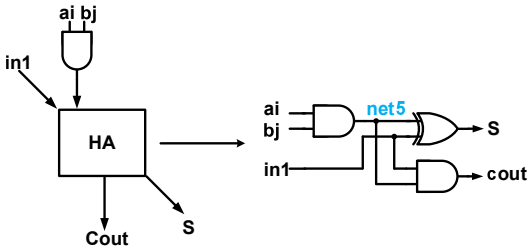

Fig. 5. MHA

By plugging all the switching activities of AND, HA, FA, MFA and MHA into the general form generated form derived from Fig. 3, the overall switching activity of an $n \times n$ array multiplier can be attained.

D. Application to ROM-less DDFS design selection

As stated earlier, although DDFS is a critical component in many applications, it is hard to predict its power dissipation before it is physically realized. By taking advantage of the above approach, it is quite easy to find out the possible performance of power dissipation for different realization architecture. Fig. 6 shows a straight forward implementation of DDFS realization using linear interpolation equations as follows.

$$y_i(x) = a_i x + b_i, \quad i = 1 \sim 8 \tag{4}$$

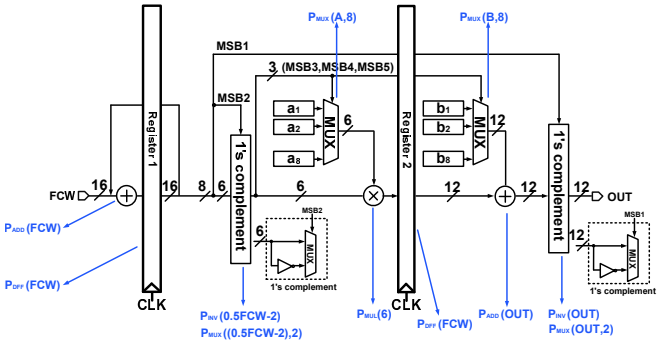

Fig. 6. Realization of a linear interpolation DDFS

The total switching activity of the linear interpolation realization for ROM-less DDFS is summarized in Eqn. (5), where FCW stands for the frequency count word of the DDFS. In other words, any ROM-less architecture is "power-consumption" predictable as long as the word length and the characteristic equations are given.

$$\begin{aligned} P_{linear-DDFS} &= P_{ADD}(FCW) + P_{DFF}(FCW) \\ &+ P_{INV}(0.5 \cdot FCW - 2) \\ &+ P_{MUX}((0.5 \cdot FCW - 2), 2) \\ &+ P_{MUX}(A, 8) + P_{MUL}(6) \\ &+ P_{DFF}(FCW) + P_{MUX}(B, 8) \\ &+ P_{ADD}(OUT) + P_{INV}(OUT) \\ &+ P_{MUX}(OUT, 2) \end{aligned} \tag{5}$$

III. Simulation and Verification

To verify the performance of the proposed approach, 3 different types of interpolation-based ROM-less DDFS with FCW = 32 bits, namely, parabolic, quasi-linear, and linear are carried out using TSMC 0.18um Mixed Signal/RF Process cell library and Altera FPGA Cyclone II EP2C35F672C6 platform. All of the realized DDFSs are functionally proved given different system clocks, including 20, 40, 50, 60, 80, 100, and 120 MHz, to justify the power dissipation prediction.

Fig. 7 summarizes the power dissipation of the 3 types of DDFS designs given different clock rates using the mentioned TSMC cell library, where the verification tool is Design Compiler. By contrast, the same 3 designs are also realized and downloaded to the mentioned FPGA platform to be verified. Fig. 8 shows the overall power dissipation over the designated frequency range of this experiment.

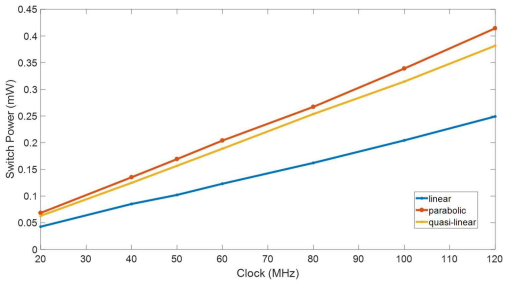

Fig. 7. Power dissipation summary by TSMC cell library in Design Compiler

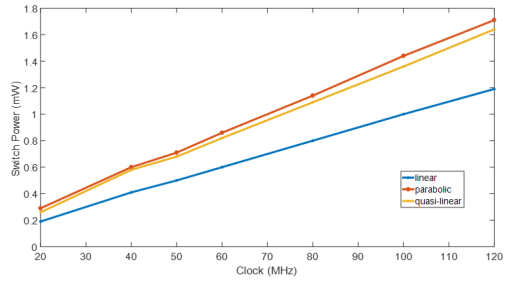

Fig. 8. Power dissipation summary by Altera FPGA platform

Since the parabolic interpolation has been recognized as the most accurate DDFS implementation approach [10], it is fair to use it as the standard normalization factor to make a fair comparison. The accuracy of this work compared with a recent report is given in Table I. Apparently, the proposed approach drastically increases the estimation accuracy by least 10%, and up to 78%.

$$\frac{\text{linear}}{\text{parabolic}} = \frac{\text{switch power of linear interpolation}}{\text{switch power of parabolic interpolation}}$$

$$\frac{\text{quasilinear}}{\text{parabolic}} = \frac{\text{switch power of quasilinear interpolation}}{\text{switch power of parabolic interpolation}}$$

TABLE I
Performance Comparision

	[11]	$\frac{\text{linear}}{\text{parabolic}}$ of this work	$\frac{\text{quasilinear}}{\text{parabolic}}$ of this work
year	2017	2018	2018
error	29%	26.14%	6.17%

IV. Conclusion

An effective and simple approach to analyze the bit switching activities of major logic circuits such that the dynamic power profile can be estimated before the ROM-less DDFS is realized on silicon. Physical verification by both cell library and FPGA platform also verify the correctness and the accuracy of the derived analytic solutions, which will relax and facilitate the low-power DDFS design as soon as the architecture and the polynomial equations are given.

Acknowledgement

This investigation is partially supported by Ministry of Science and Technology, Taiwan, under grant MOST 107-2218-E-110-004-, and 107-2218-E-110-016-. The authors would like to express their deepest gratefulness to Chip Implementation Center of National Applied Research Laboratories, Taiwan, for their EDA tool support.

References

[1] Y. Sun, X. Yu, W. Rhee, D. Wang, and Z. Wang, "A fast settling dual-path fractional-N PLL with hybrid-mode dynamic bandwidth control," *IEE Microwave and Wireless Components Letters*, vol. 20, no. 8, pp. 462-464, Aug. 2010.

[2] W.-Y. Shin, M. Kim, G.-M. Hong, and S. Kim, "A fast-acquisition PLL using split half-duty sampled feedforward loop filter," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1856-1859, Aug. 2010.

[3] M. Kesoulis, D. Soudris, C. Koukourlis, and A. Thanailakis, "Systematic methodology for designing low power direct digital frequency synthesizers," *IET Circuits, Devices & Systems*, vol. 1, no. 4, pp. 293-304, Aug. 2007.

[4] J. Tierney, C. Rader, and B. Gold, "A digital frequency synthesizer," *IEEE Trans. on Audio and Electroacoustics*, vol. 19, no. 1, pp. 48-57, Mar. 1971.

[5] A. Ashrafi and R. Adhami, "An optimized direct digital frequency synthesizer based on even fourth order polynomial interpolation," in Proc. of *38th IEEE Southeastern Symposium on System Theory*, pp. 109-113, Mar. 2006.

[6] H. Jafari, A. Ayatollahi, and S. Mirzakuchaki, "A low power, high SFDR, ROM-less direct digital frequency synthesize," in Proc. of *2005 IEEE Conf. on Electron Devices and Solid-State Circuits*, pp. 50-54, Dec. 2005.

[7] E. Lopelli, J. D. van der Tang, and A. H. M. Roermund, "Mimimum power-consumption estimation in ROM-based DDFS for frequency-hopping ultralow-power transmitters," *IEEE Trans. on Circuits & Systems - I : Regular Papers,* vol. 56, no. 1, pp. 256- 267, Jan. 2009.

[8] C.-C. Wang, C.-H. Hsu, C.-C. Lee, and J.-M. Huang, "A ROM-less DDFS based on a parabolic polynomial interpolation method with an offset," *Journal of Signal Processing Systems* vol. 61, pp.1-9, May 2010.

[9] J.-Y. Tsai, "Low power techniques for digital IC design," *CICeNEWS,* vol. 86, pp. 1-22, Dec. 2007.

[10] D.-S. Wang, Y.-S. Liu, and C.-C. Wang, "A novel frequency-shift readout system for CEA concentration detection application," *2016 The 13th Inter. SOC Design Conf.,* pp. 133-134, Oct. 2016.

[11] W. Wang, Y.-Y. Xu, and C.-C. Wang, "Dynamic power estimation for ROM-less DDFS designs using switching activity analysis," *2017 The 14th Inter. SoC Design Conf.,* pp. 280-281, Nov. 2017.