Exponential Split Accumulator for High-Speed Reduced Area Low-Power Direct Digital Frequency Synthesizers

Edward Merlo
emerlo@rwsc.com

Kwang-Hyun Baek
kbaek@rwsc.com

Myung-Jun Choe
mchoe@rwsc.com

Rockwell Scientific Company
1049 Camino dos Rios
Thousand Oaks, CA 91360

ABSTRACT

A new split accumulator architecture to be used in direct digital frequency synthesizers (DDFS) systems is presented. This new design eliminates the need of adders on the section of the accumulator that is not used to address the phase to amplitude mapping block thus reducing area, constraints in implementation, and up to 82% in power consumption compared to standard designs.

Categories and Subject Descriptors

B.6.1 [**Logic Design**]: Design Styles – Combinational Logic.
B.7.1 [**Integrated Circuits**]: Types and Design Styles – VLSI.

General Terms

Performance, Design, Algorithms.

Keywords

Direct digital frequency synthesizer, DDFS, numerically controlled oscillator, NCO, low power, phase accumulator.

1. INTRODUCTION

With today's access to high-speed, high-integration processes, DDFS systems are becoming an alternative to analog-based phase looked loop (PLL) synthesizers. Advantages of DDFS Systems over analog frequency synthesizers include sub-Hertz resolution, fast switching between frequencies, and less susceptibility to aging and temperature changes [1]. A simplified block diagram for a traditional DDFS implementation is shown in Figure 1. It has an accumulator that generates a ramp representing phase values in the 0:2pi interval that will address the sine wave mapping where the phase is translated into amplitude and is scaled to the range of the digital-to-analog converter (DAC) input (Figure 1).

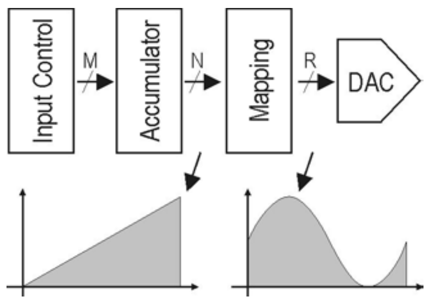

Figure 1. DDFS block diagram.

The ramp generated by the phase accumulator sweeps from 0 to 2^M-1. On present designs M can range from 24 to up to 64 bits. The frequency of the sweep is controlled by the frequency control word (FCW), making the frequency of the output sine wave F_{out} = (FCW * CLOCK) / 2^M. The sine wave mapping block implementation falls into two categories: Computational and Lookup Table (LUT) methods. Computational methods like the CORDIC algorithm, Taylor series and parabolic approximation [5] use arithmetic to approximate the waveform while LUT methods use read only memory (ROM) that contains the amplitude values of specific phases. They are addressed by the output of the phase accumulator. A survey of different designs can be seen in [2]. On either method the number of bits coming out of the accumulator has a direct influence on size and complexity of sine wave mapping. In the case of a 32bit accumulator with a LUT mapping and a 12bit DAC, the size of the ROM needed to implement the LUT will be 2^{32} x 12 or more than 51 x 10^9 bits.

The solution is not to use the full accumulator's output but just a fraction of it, as shown in Figure 2, where only N bits out of M bits are used. The accumulator output word is shown in Figure 3. The effect of this truncation is a reduced size ROM and a phase precision loss. This loss will appear as spurious frequencies on the output spectrum, and its amplitude will dictate the size of the truncation according to system goals. Although truncation eases sine wave mapping, it does little to help the accumulator design. It still needs to operate at full resolution, adding a full M bits every clock cycle.

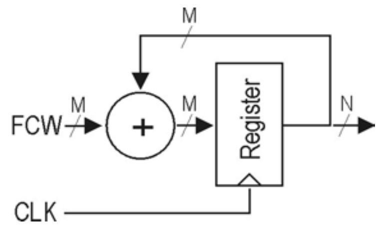

Figure 2. Mbit phase accumulator with truncated output to Nbit.

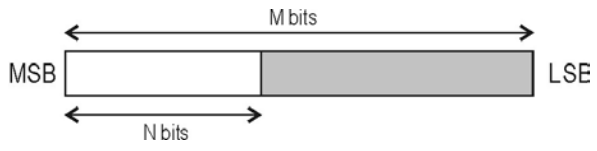

Figure 3. Accumulator output word.

2. PHASE ACCUMULATOR

The standard approach to implementing an accumulator for high-speed DDFS systems is to use a pipelined architecture. Although it can achieve high throughput it has high power consumption and uses a large area due to the number of registers needed to keep the data coherent. The split accumulator and the exponential split accumulator address those issues.

2.1 Pipelined Accumulator

Until now, to overcome technology restrictions to achieve high-speed designs, pipeline architectures were used as exemplified in Figure 4. A number of registers are needed to keep the input and output of the accumulator coherent. The registers before the adders on the higher left corner are called preskewing registers, and the ones at the upper right corner, after the adders, are called deskewing registers [3]. Deskewing registers are needed on the N bits that are part of the clipped output of the accumulator. The size of each stage is defined by the pipeline clock period. Although this architecture allows for a higher throughput, latency is introduced. Now 8 clock cycles are required to compute a full addition in the accumulator in Figure 4.

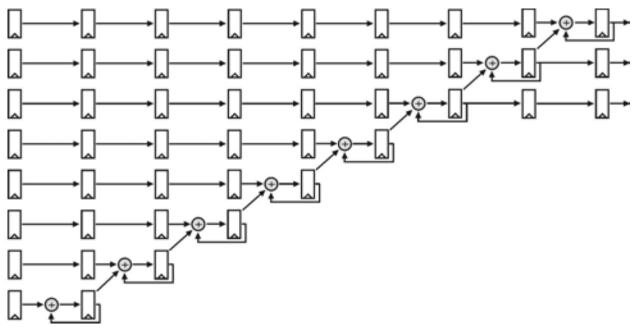

Figure 4. Standard pipelined accumulator

2.2 Split Accumulator

As seen above, the pipelined accumulator is running at full clock speed and the pre and deskewing registers use a large area. The split accumulator [4] reduces the number of registers needed in the pipeline and runs the portion of the accumulator not visible to the mapping block at a lower clock frequency, reducing the area and using less power.

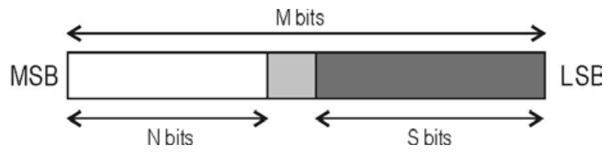

Figure 5. Split accumulator word.

On DDFS applications the desired accumulator has a large width to improve resolution. To reduce the size or the computation effort of the sine wave mapping block, its output is truncated to a smaller width. Lower significant bits (the discarded part) of the accumulator do not directly influence the sine wave mapping. Therefore the only section of the accumulator that needs to be updated constantly is the one that supplies the bits used by the mapping function. The accumulator can then be divided into two sections, the upper section containing the bits that will address the sine wave mapping block and the lower section that is discarded ((M-S) and S on Figure 5). The only connection between the two is the carry data from the lower to the upper section. This means that as long as the upper section of the accumulator gets updated with the correct carry data, there are no constraints on how to implement the lower section.

The idea, as shown in Figure 6 without the pipeline, is to have the lower section operating at half the speed of the upper part. By doing so, constraints on implementation of the lower part are eased; it uses less power (operates at lower frequency) and saves area since fewer pipeline stages are needed. The problem is to make sure that the carry data between sections gets updated correctly. Since the lower section operates at half the frequency, the FCW for that section needs to be twice the value as the original one. In Figure 7 this is done by the X2 block or by shifting the lower part of the FCW. This will generate a carry that will be used by the upper section every two clock cycles. The lower section of the accumulator will also generate its own carry every two clock cycles. Those two carry bits are interleaved by the MUX shown in Figure 6.

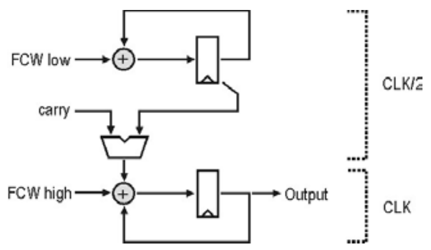

Figure 6. Split accumulator architecture.

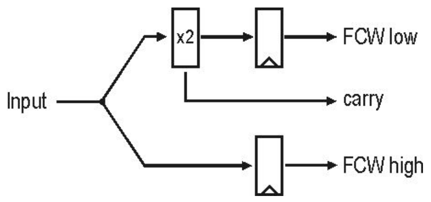

Figure 7. Accumulator front-end showing split FCW.

One problem that arises from this architecture is that the carry feed to the upper section has a jitter. It can, occasionally, be off by one clock cycle compared to the full implementation of the accumulator. It means that the least significant bit of the upper section can be occasionally off the correct value by a value of one. Avoiding the use of that bit to address the sine wave mapping corrects the problem. A full pipelined split accumulator circuit, comparable to the standard pipelined accumulator (Figure 4), is shown in Figure 8.

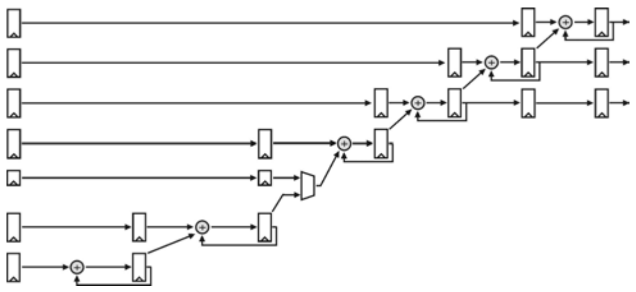

Figure 8. Pipelined split accumulator.

2.3 Exponential Split Accumulator

On the split accumulator the interface between the lower section and the upper section is done by a carry signal. This carry signal is generated by interleaving the carry out from the X2 block or the most significant bit (MSB) of the S section of the accumulator input with the carry from the lower part of the accumulator that now have (S-1) bits and runs at half the clock frequency.

The same reasoning can be applied to the remaining (S-1) section of the lower part of the accumulator. The connection between the (S-1) section to the rest of the accumulator can be done by a carry signal that is the interleaved result of the MSB of the (S-1) section and the carry from the (S-2) remaining bits that will be running at ¼ the speed. By recursively applying this reasoning until the least significant bit of the accumulator, we arrive at the exponential split accumulator design, shown with 8 bits on the split section on Figure 9. The main difference between the exponential split accumulator and previous designs is that no adders are used in the lower part of the accumulator. A MUX coupled to every input bit in the lower part selects the appropriate carry to be propagated to the upper part of the accumulator. The selection of which input is propagated through the MUX is done by a control signal generated from the operating clock. Every bit away from the upper part of the accumulator has its MUX controlled by the control signal of the previous bit divided by 2. Another advantage is that only one pipeline stage is needed for the lower part of the accumulator, thus reducing its latency.

Figure 10 shows the circuit for the pipelined exponential split accumulator. Comparing with the previous pipelined designs, it has 5 stages, while the split accumulator has 6 stages and the traditional pipelined accumulator has 8 stages.

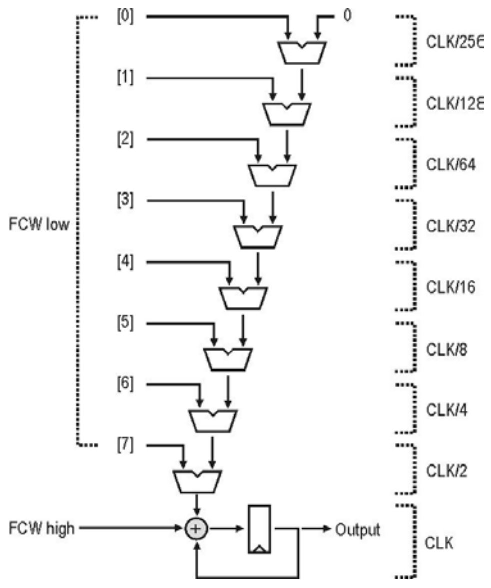

Figure 9. Exponential split accumulator architecture with 8bit split section.

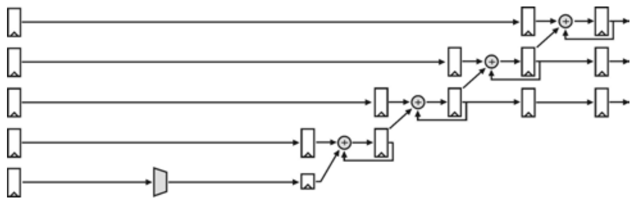

Figure 10. Pipelined exponential split accumulator.

3. RESULTS

All three architectures were simulated in Matlab to record the SFDR degradation due to the truncation of the accumulator's output. The FCW that yields the worst case of phase truncation spurs; in this case, a 32bit accumulator with its output truncated to 14bit, is a FCW with bit 17 (M-N-1) set to 1 or 0x00020000 [6]. The result (Figure 11) shows that all three architectures yield the same SFDR value of –80.36dBc. This is true since the upper section of the accumulator is the same for all architectures. Using a FCW that exercises the lower section, with bit 12 set to 1 or 0x00001000, the results were still the same (-83.37dBc) for all.

To evaluate the advantage of the proposed architecture over the standard and split accumulators, all three designs were created and simulated in HSPICE using the standard cell library from Atmel 0.35 micron process. The configuration chosen was of a

32bit accumulator with its output truncated to 12bit. On the split and exponential split architectures the accumulator was divided in half with 16bit upper and 16bit lower sections.

The results show a saving of 47% / 20% in area (Figure 12) and 61% / 21% in power over the standard / split accumulator respectively (Figure 13). Figure 13 also shows the normalized power results for the three architectures on three different frequency control words. FCW1 (7777 7777h) exercises the whole accumulator, while FCW2 (7777 0000h) and FCW3 (0000 7777h) exercise the upper and lower section. There is no difference between the split and exponential split accumulator when FCW2 is used, due the fact that the upper part of the accumulator is the same for both architectures. The difference becomes obvious with FCW3. With only the lower section of the accumulator operating, the exponential split accumulator shows an improvement of 82% over the standard and 46% over the split accumulator in power.

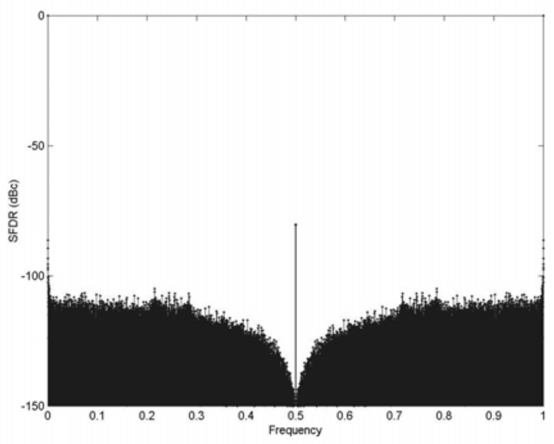

Figure 11. Output spectrum of a NCO with 32bit accumulator truncated to 14bit.

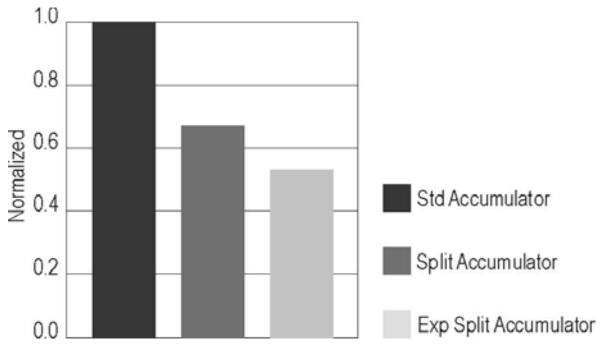

Figure 12. Area results

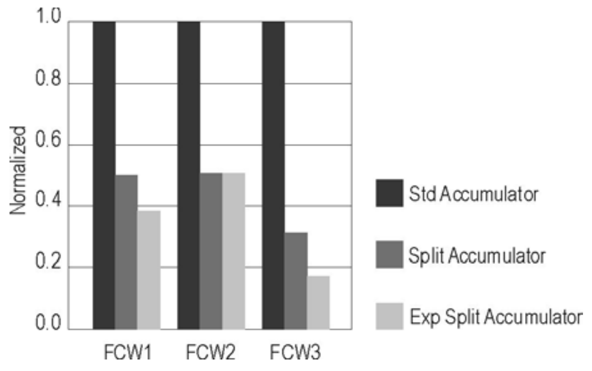

Figure 13. Power consumption results

4. CONCLUSION

This paper shows a new accumulator architecture for high-speed low-power DDFS systems that has advantages over the existing designs. By having the non-visible part of the accumulator implemented without the use of adders and running at exponentially lower speeds, it uses less power, less area and eases restrictions on the implementation. With this architecture wider accumulators can be implemented, improving the frequency resolution of high performance DDFS systems without compromising size and power consumption.

5. ACKNOWLEDGMENTS

This work was supported by Rockwell Collins as part of the SiGe Data Conversion Circuits Development Program.

6. REFERENCES

[1] *A Technical Tutorial on Digital Signal Synthesis*, Analog Devices, 1999.

[2] K. A. Essenwanger, and V. S. Reihardt, “Sine Output DDSs a Survey of the State of the Art,” 1998 IEEE International Frequency Control Symposium, pp. 370-378.

[3] F. Lu, H. Samueli, J. Yuan, and C. Svensson, “A 700Mhz 24-b Pipelined Accumulator in 1.2-μm CMOS for Application as a Numerically Controlled Oscillator,” IEEE Journal of Solid-State Circuits, vol. 28, no. 8, pp. 878-886, August 1993.

[4] E. Merlo, K.-H. Baek, and M.-J. Choe, “Split Accumulator with Phase Modulation for High Speed Low Power Direct Digital Synthesizers”, ASIC/SOC 2002 – 15th Annual IEEE International ASIC/SOC Conference, September 25-28, 2002.

[5] A. M. Sodagar, G. R. Lahiji, “Parabolic Approximation: A New Method for Phase-to-Amplitude Conversion in Sine-Output Direct Digital Frequency Synthesizers,” ISCAS 2000 – IEEE International Symposium on Circuits and Systems, pp. 515-518, May 28-31, 2000

[6] A. Torosyan, and A. N. Willson, Jr., “Analysis of the Output Spectrum for Direct Digital Frequency Synthesizers in the Presence of Phase Truncation and Finite Arithmetic Precision,” ISPA 2001 – 2nd International Symposium on Image and Signal Processing and Analysis