

DESIGN OF A COMPACT DIRECT DIGITAL FREQUENCY SYNTHESIZER WITH 12 BIT AMPLITUDE AND 32 BIT FREQUENCY RESOLUTION

G. Fischer

Dept. of El. & Comp. Engr.
University of Rhode Island
Kingston, RI 02881

N. K. Modadugu

Dept. of El. & Comp. Engr.
University of Rhode Island
Kingston, RI 02881

Abstract - This paper describes the design of a monolithic direct digital frequency synthesizer. The circuit realizes a 12 bit output sine wave with a frequency resolution of 32 bit. The core of the $1.2\mu\text{m}$ CMOS implementation consists of approximately 6,000 transistors and occupies an area not larger than 1.5mm^2 . The circuit is aimed at a maximum tuning range of 100MHz, or equivalently, a clock rate of 200MHz. This upper value yields a minimum frequency increment of 0.023Hz. The system exhibits a total latency of 14 clock periods.

INTRODUCTION

Frequency synthesizers are widely used components in communication systems. They are frequently applied in signal modulation/demodulation and filter tuning circuits. Most of these tasks require high frequency resolution as well as high bandwidth. With the advent of frequency agile communication systems such as spread spectrum LAN's, frequency-hopped systems and digital cellular telephones, demands on switching speed of frequency synthesizers have significantly increased.

In conventional synthesizers based on phase-locked loops (PLL's), there exists an inherent trade-off between frequency resolution and switching speed [1]. Hence, PLL type circuits are ill-suited for applications in frequency agile systems, where both high resolution and high switching speeds are required. By contrast, direct digital frequency synthesizers (DDFS's) represent an approach where the resolution is independent of the switching speed.

The most advanced CMOS based DDFS's reported in the literature enable a maximum tuning range of up to 400MHz [2]. Such a high operating frequency has been achieved by a highly parallel circuit layout. In contrast to these extremely high-speed implementations, the objective of our design was rather to minimizing the device count and the chip area¹. Apart from minimizing implementation costs, a very compact layout should also be attractive

¹For economical reasons, our circuit had to fit on a die size of not more than 4mm^2 .

from the point of view of fabricating with a very advanced CMOS or possibly GaAs process where the yield is correspondingly lower.

THE DDFS APPROACH

Most DDFS's are implemented using an architecture developed by Tierney et al. [3]. This approach exploits the modulo 2^N overflowing property of an N-bit accumulator to generate the 2π -periodic phase argument of the sine wave. During each clock cycle, a constant phase increment in form of an externally applied digital control word $\Delta\Phi$ is added to the previous content of the phase accumulator such that $\Phi(n+1) = \Delta\Phi + \Phi(n)$. The output frequency of the DDFS is therefore given by

$$f_{out} = f_{clk} \frac{\Delta\Phi}{2^N} \quad (1)$$

The incremental phase value $\Delta\Phi$ can range from 0 to 2^{N-1} . Consequently, the frequency resolution is determined exclusively by the number N of bits used in the phase accumulator. In addition, a new frequency is readily synthesized by altering the phase increment $\Delta\Phi$. Thus, a DDFS may tune between any two frequencies within the duration of one reference clock period. It is therefore well suited for applications which demand very high switching speeds.

Finally, it should be mentioned that the DDFS approach not only offers direct digital frequency control but also an equally straight-forward way of controlling the phase. A phase shift is readily accomplished by adding a numerical offset to the output of the accumulator.

SYSTEM PERFORMANCE

The spectral purity of a DDFS is limited by three error sources: phase truncation errors, finite word length effects and computational rounding errors. Computational errors can be avoided by generating the sine function via a look-up table, i.e., a ROM. Phase truncation errors can be minimized by utilizing all bits of the phase accumulator as address bits for the look-up table. However, this straight-forward approach requires an enormous memory capacity (e.g. 48 Gbit for 32 address bits and 12 bit amplitude resolution). A more pragmatic approach is to keep the phase truncation error smaller than the quantization error. In our implementation, we have chosen 12 bit amplitude resolution and 14 bit phase resolution. This yields a worst case phase truncation error of 0.39LSB's, or a more relevant rms error of 0.16LSB's, respectively. The amplitude of the worst case spurious frequency is limited by 2^{-14} [4]. The truncation of the phase to 14 bit reduces the memory size to 192 kbit. Even though significantly smaller than the original 48 Gbit, this number is still too large to enable an area efficient implementation with a minimum data access time. By means of a novel combination of different

memory compression techniques such as exploiting the quarter wave symmetry of the sine function, subdividing the memory into coarse and fine resolution sections and using incremental data storage, (see the block diagram in Fig.3), we could reduce the amount of digital storage from 192 kbit to approximately 2.2 kbit. This reduction in memory capacity has to be paid for by a more complex computational procedure which requires three additional adders. Furthermore, since the algorithm generates the sine samples by adding memory entries from two independent look-up tables, the total rounding error increases by 41%, i.e., the total quantization noise power doubles. The spectral purity of the resulting sine function is thus dominated by finite word length effects. Since the sine values are represented by 11 magnitude bits and one sign bit, the average total noise power cannot be lower than -74dB. If we include phase truncation effects, this value slightly increases to -73.4dB. Recall that the corresponding SNR is 3dB lower than the magnitude of the total noise power because the sinusoidal output, being normalized to unity amplitude, possesses an equivalent power of -3dB.

The theoretical system performance has been confirmed by a behavioral model which exactly mimics the algorithm's finite word length effects and phase truncation errors. Fig.1 shows the simulated output spectrum for $f_{out1} = \frac{64}{4096} f_{clk}$, a case which only generates harmonics, while Fig.2 displays the output spectrum for $f_{out2} = \frac{63}{4096} f_{clk}$, a situation which also gives rise to spurious frequencies. To achieve coherent sampling and thus avoid windowing effects, the two spectra have been computed via a 4096 point FFT. In the first example, the total harmonic power amounted to -75.6dB while the second case yielded a total noise plus harmonic power of -73.8dB, respectively.

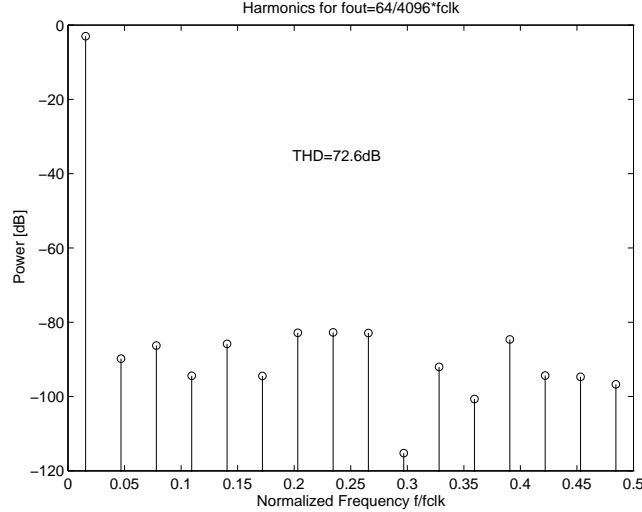


Figure 1: Output spectrum if the output frequency is an integer fraction of the clock frequency.

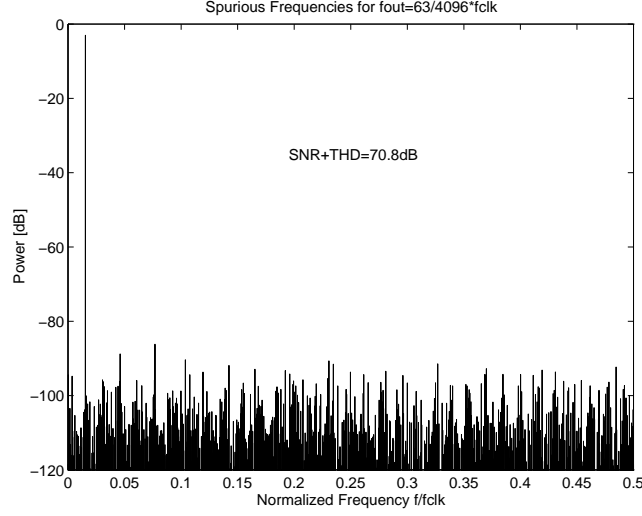


Figure 2: SNR+THD if the output frequency is a non-integer fraction of the clock frequency.

CIRCUIT ARCHITECTURE

The block diagram of the entire system is shown in Fig.3. To reduce the number of output pins, the 32 bit digital frequency control word is loaded in two steps of 16 bit each.

To minimize propagation delay, the 33 bit adder of the phase accumulator has been subdivided into 4-bit slices which have been 8-fold pipelined. The remaining LSB stage has been realized by a simple toggle flip-flop (sum bit) and an AND gate (carry out). While this solution is very area efficient, it introduces a latency of 8 clock cycles.

Apart from exploiting the quarter wave symmetry of the sine function, the content of the look-up table has been further compressed by having applied the following trigonometric approximation:

$$\sin\left(\frac{\pi}{2}[x_1 + x_2 + x_3]\right) \approx \sin\left(\frac{\pi}{2}[x_1 + x_2]\right) + \cos\left(\frac{\pi}{2}x_1\right)\sin\left(\frac{\pi}{2}x_3\right) \quad (2)$$

where x_1 denotes the first few MSB's, x_2 the subsequent intermediate bits and x_3 the remaining LSB's of the argument of the sine function (i.e., the address of the ROM). The values of $\sin(\frac{\pi}{2}[x_1 + x_2])$ form the content of a *coarse* ROM while the products $\cos(\frac{\pi}{2}x_1)\sin(\frac{\pi}{2}x_3)$ are stored in a *fine* ROM. The maximum approximation error resulting from (2) is approximately

$$\epsilon_{Max} \approx \left(\frac{\pi}{2}\right)^2 x_{2Max} x_{3Max} \quad (3)$$

When assigning the first 5 bits of the phase argument to x_1 , the next 3 bits to x_2 and the remaining 4 bits to x_3 , the maximum approximation error is

bounded by $0.8 \times (\frac{\pi}{2})^2 2^{-13}$, or about 0.5LSB's. This represents an acceptable compromise for the achieved memory compression.

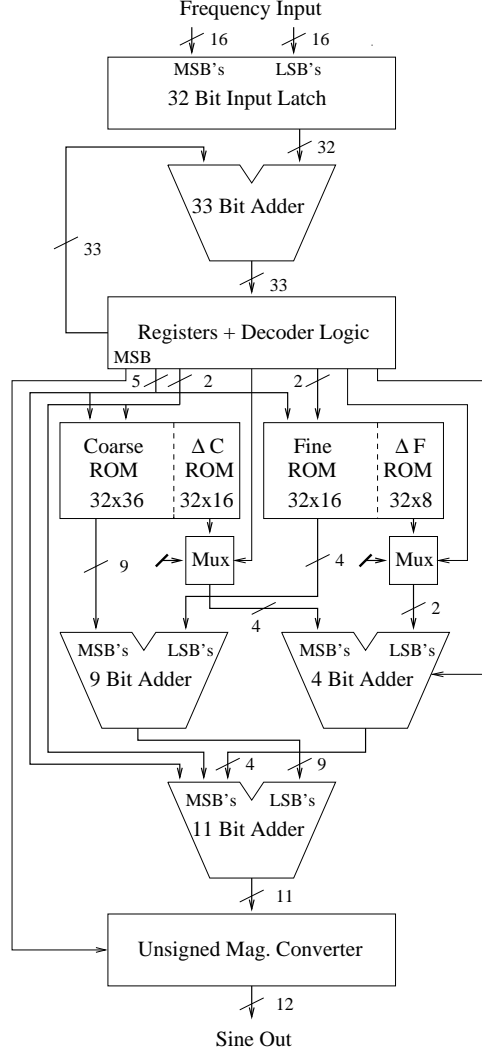


Figure 3: Block Diagram of the proposed frequency synthesizer.

By replacing $\sin(\frac{\pi}{2}x)$ by the function $[\sin(\frac{\pi}{2}x) - x]$, the coarse ROM could be reduced by an additional 2 bits per word. The missing x value, which consists of the 7 MSB's of the coarse ROM address, is subsequently added to the ROM output in the final 11 bit adder stage. A third memory reduction has been achieved by having replaced every second memory entry by the increment from its preceding value. Finally, we have removed the resulting 5 all-zero columns that occurred in the fine ROM section of the look-up

table. Having applied all these compression techniques, the total memory size did shrink from 48 kbit for the original uncompressed quarter wave sine function to approximately 2.2 kbit. This not only enabled a very compact implementation, but also helped to reduce the crucial ROM access time.

In order to minimize quantization and approximation errors, the resulting memory values have been subjected to a numerical optimization procedure which minimized the rms error of all 4096 samples of the quarter wave sine function. The thus adjusted memory values yield an rms quantization plus approximation error of 0.41LSB's. This is only 41% more than the quantization error resulting from an uncompressed look-up table with 11 bit magnitude resolution. In fact, it is exactly equal to the expected error resulting from the addition of two independently quantized 11 bit table entries.

As mentioned before, the price for the compactness of the memory has to be paid for in form of three additional adder stages. The assembly of the sine samples from the different memory locations is accomplished in two sequential steps. Since these additions have again been pipelined by employing 4 bit ripple-carry adder sections, they introduce an additional latency of 4 clock cycles.

The bottom block in Fig.3, finally, adds the missing sign bit to the 11 bit ROM output. In order to be able to directly feed the 12 bit digital output into a current-mode digital-to-analog converter, the output has been encoded in unsigned magnitude format.

LAYOUT CONSIDERATIONS

The floor plan of the DDFS circuit closely followed the block diagram depicted in Fig.3. This *natural* data flow avoided excess routing area and thus minimized circuit loads. Apart from the two ROM sections which have been realized by an automatic layout tool, the layout of the 1.2 μ m implementation has been accomplished by the public domain tool *Magic* following MOSIS' scalable design rules.

To ease testing, the phase accumulator possesses an external reset. Timing for the entire circuit is controlled by a single-phase clock. To minimize skewing, the clock line branches out from a central stem into 5 parallel sections with approximately equal load conditions. Each of these sections possesses a separate driver.

As far as operating speed is concerned, the most critical building blocks are the adders. The employed 4-bit slices have been realized as ripple carry circuits. The propagation delay has been minimized by utilizing an optimized carry generator formed by a complex gate followed by an inverter. The topology of the basic adder cell is depicted in Fig.4. According to a transistor level simulation carried out by *CAzM*, the 4-bit adder slice exhibits a maximum propagation delay of 2.8ns. We thus conclude that the pipelined adders should be able to settle within a minimum cycle time of 5ns. This would translate into a maximum clock frequency of 200MHz, or a tuning range of 100MHz, respectively.

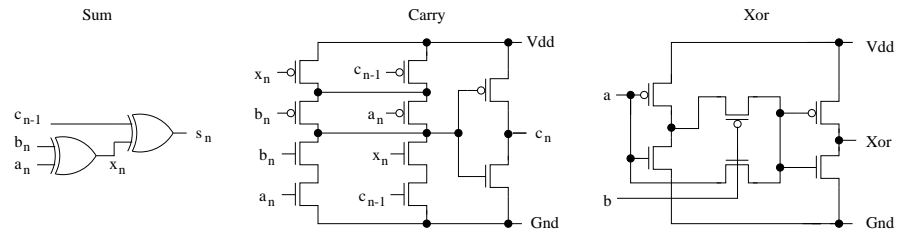


Figure 4: Implementation of 1 bit ripple-carry adder cell and Xor gate.

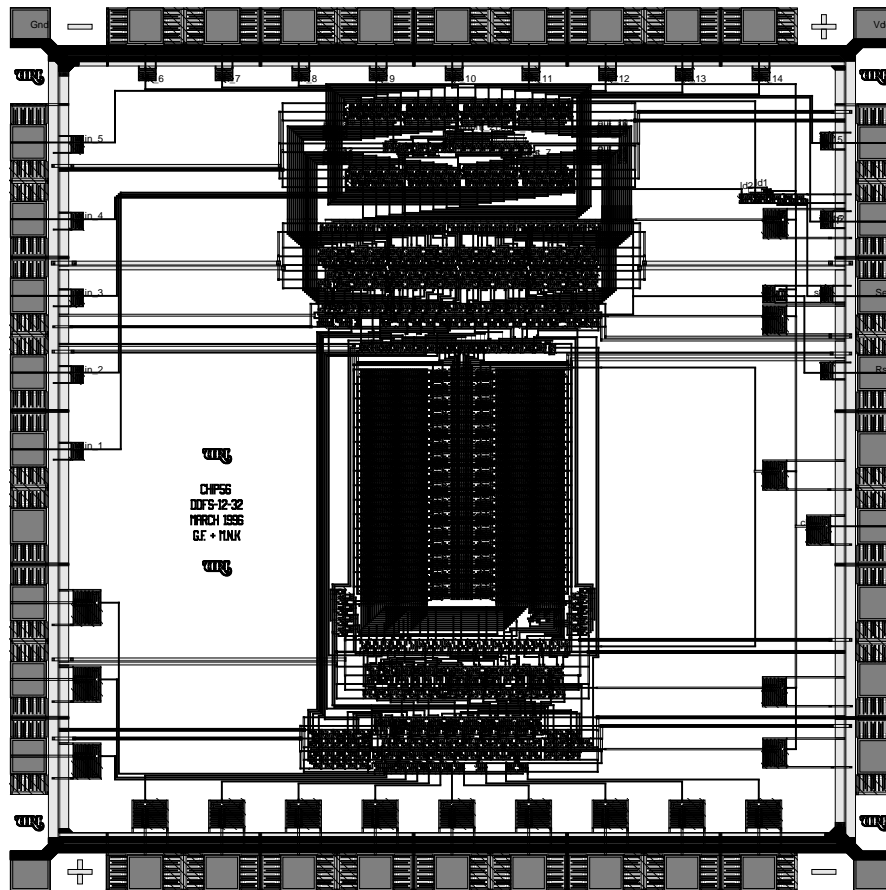


Figure 5: Complete layout of the frequency synthesizer chip.

The functionality of the layout has successfully been tested at the maximum clock rate of 200MHz by means of the switch level simulator *IRSIM*.

At the time of this writing, the circuit is still in fabrication so that no experimental results can be reported.

The complete layout of the DDFS chip is shown in Fig.4. The main building blocks are, from top to bottom, the 32 input latches, the 33 bit phase accumulator, the coarse and fine ROM sections, the three additional adders and the unsigned magnitude code converter. The entire chip comprises approximately 6,500 transistors (including I/O stages) while the central DDFS core occupies an area of less than 1.5mm².

CONCLUSIONS

The design of a compact DDFS circuit with 12 bit amplitude and 32 bit frequency resolution has been presented. By employing a novel combination of memory compression techniques, the look-up table for the sine function could be reduced from 48 kbit for the uncompressed quarter wave to a mere 2.2 kbit. This not only enabled a very area efficient implementation but also minimized the data acquisition time.

The circuit features a spurious free dynamic range of 84dB and a typical SNR of 71dB, i.e. 3dB less than the maximum for 12 bit amplitude resolution.

The wide adder circuits have been pipelined into 4 bit slices so that the 1.2μm implementation is expected to enable a maximum tuning range of approximately 100MHz. The pipelining increased the total latency of the circuit to 14 clock cycles. With approximately 6,000 transistors and a core size of less than 1.5mm², the presented DDFS circuit is indeed compact.

If the circuit were to be fabricated by a more advanced 0.8μm CMOS process, it is expected that the tuning range of the synthesizer could be as wide as 200MHz.

References

- [1] V. Manassewitsch, "Frequency Synthesizers, Theory and Design", 2nd Ed., New York: Wiley, 1989.
- [2] L. K. Tan, E. Roth, G. E. Yee, and H. Samueli, "An 800MHz quadrature digital frequency synthesizer with ECL-compatible output drivers in 0.8μm CMOS", *Dig. of IEEE ISSC95*, pp.258-259, San Francisco, Feb. 1995.
- [3] J. Tierney, C.M. Rader, and B. Gold, "A digital frequency synthesizer", *Trans. Audio Electro-acoustics*, vol. AU-19, pp.48-57, 1971.
- [4] H. T. Nicholas, III, and H. Samueli, "An analysis of the output spectrum of direct digital frequency synthesizers in the presence of phase-accumulator truncation", *Proc. of 41st Annual Frequency Control Symp. USERACOM* (Ft. Monmouth, NJ), pp.495-502, May 1987.