Performance of a Fast Analog VLSI Implementation of the DFT

B. Buchanan *, V. Madisetti **, M. Brooke *

Abstract

A fast, analog implementation of the DFT/IDFT requires solutions to the problems of I/O bottleneck encountered by large, parallel input sequences, the slow execution time of long sequential sequences, and the resultant error. We present an architecture based on several modifications to Goertzel's Algorithm that provides balances between input serialization, circuit area, execution time, and output error.

1 Introduction

The potential speed advantages to Digital Signal Processing (DSP) applications through the use of auxiliary analog processing circuits encourage the development of circuits for this purpose. Much of the theory behind DSP is developed for discrete time signals, and often applies to analog as well as to digital signal representations. In this paper we propose a modular analog circuit architecture for the calculation of Discrete Fourier Transforms and Inverse Discrete Fourier Transforms (DFT/IDFTs) based on Goertzel's Algorithm that allows for flexible design compromises between circuit area, speed, resultant error, and I/O requirements.

Though extremely fast and relatively accurate, completely parallel implementations of a DFT/IDFT for large input sequences are impractical because of circuit I/O limitations, the extremely high interconnection of the input signals' distribution, and the large silicon area necessary. For Real input signals, each of the N output samples of such an architecture would require N multipliers and an adder with a fan-in of N. For an input sequence of length N=512, and a conservative 100 square microns per multiplier, the multipliers alone in a fully parallel implementation would require an area a little over 5 mm^2 per output point.

Goertzel's Algorithm (GA) [1,2] provides a recursive solution to complex vector multiplication under certain conditions. DFTs and IDFTs meet the requirements of this special case, each point of the DFT or IDFT requiring one invocation of GA. When serially provided with the components of a length N input vector, GA requires N cycles of the recursion to produce the final result by sequentially adding the current component of the input vector to the accumulated results, and then rotating the total in the complex plane by the angle of the primitive Nth root of unity [Figure 1]. The desired result is the accumulated value after the Nth cycle of recursion.

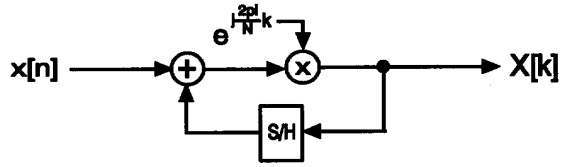


Figure 1: Goertzel's Algorithm

Note that the most primitive root required using this method for a DFT/IDFT calculation, $e^{j2\pi/N}$, can be used to calculate each point of the output sequence by either reordering the input sequence, or padding it with zeros and extending the recursion appropriately. This allows an individual implementation of GA to compute any of the N output points in a DFT/IDFT calculation.

2 Architecture

Such an individual implementation of a DFT/IDFT based on GA requires only four multipliers, three adders, and two sample and hold (S/H) circuits [Figure 2]. In contrast to the fully parallel implementation, which requires $O(N^2)$ area and O(1) time to compute all output points of a DFT/IDFT, a fully serial implementation using a single GA circuit requires O(1) area and $O(N^2)$ time. In addition to swapping concerns over circuit area and signal interconnection for time, each step of the recursion is an opportunity for error accumulation.

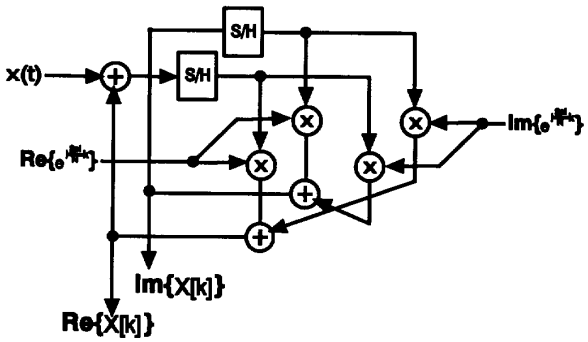


Figure 2: GA Architecture

The most obvious engineering compromise between these two extremes is to compute each of the N DFT/IDFT output points in parallel using multiple instances of GA; for example, all N output points could be computed using N instances of GA, requiring O(N) area and O(N) time. Input connectivity becomes a problem as the demand for different input sequences increases. For the fully parallel case, this is not a problem though since each of the N GA circuits use the same input sequence Output connectivity remains a consideration, but it is a problem inherent to the DFT/IDFT calculation. One possible solution is to serialize the output and step it out during the input of the next sequence.

*Microelectronics Research Center, Georgia Institute of Technology, 791 Atlantic Drive, Atlanta, Georgia 30332-0269, Phone 404-853-9920

**Digital Signal Processing Laboratory, School of Electrical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-0250

This research is supported in part by a grant from Analog Devices and an IBM Faculty Development Award

0-7803-0510-8/92$03.00 ©1992IEEE

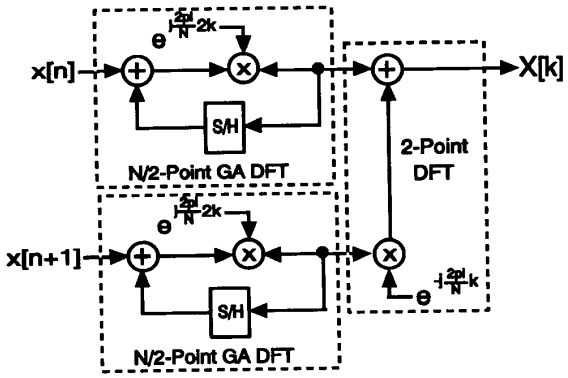


Figure 3: Example FFT Type Factorization of GA

For a non prime N, the DFT/IDFT calculation for each output sample can be factored using decimation in time FFT techniques to produce an even further parallelized version of GA [Figure 3][3]. The advantage to this distributed architecture is that the overall circuit's speed is no longer tied directly to individual GA circuits' cycling time. For example, if M is a factor of N, each point of an N length DFT/IDFT can be calculated using 1+M GA circuits in 1+M+N/M time steps (where a time step can be as little as the storage capacitor's charging time in the S/H circuits). Several layers of DFT/IDFT's can be built up this way to reduce the circuit's execution time and both the degree of recursion that each input sample is subjected to and the consequent error accumulation.

Buffers between the multiple layers can pipeline the calculation, reducing the effective calculation time even further. This reduces the effective calculation time to 1+P time-steps/result, where P is the largest factor of N that any of the layers cycle. For the N=512 example, seventy three identical 8-point GA circuits distributed across three layers can compute an output point in an effective 9 time-steps/result, with the highest degree of recursion any input sample is subjected to being 24 cycles.

3 Implementation

Since the magnitude of each of the four multiplication's in a GA circuit is less than unity, our approach to their implementation has been to use voltage dividers [Figure 4: Note that the location of the inverter(s) is dependent on the output point, k]. The additions, which can be built into the S/H circuits, are subsequently done by sampling the voltage between the dividers, and using either the stored value, its inverse, or both as necessary in the next cycle of the recursion.

With reasonable resistance values in the voltage dividers (as discussed three paragraphs below), the slowest element in this architecture is the S/H circuit. A reasonable estimate for its speed is low MHz, so for the preceding example of a real time, Real valued input sequence and N=512, results could be expected within 10 usec/result.

There are three considerable sources of error inherent in this architecture. By far the most critical is the accuracy of the voltage dividers. Additionally, the S/H circuits introduce some non-linearity and offset during each cycle of the recursion.

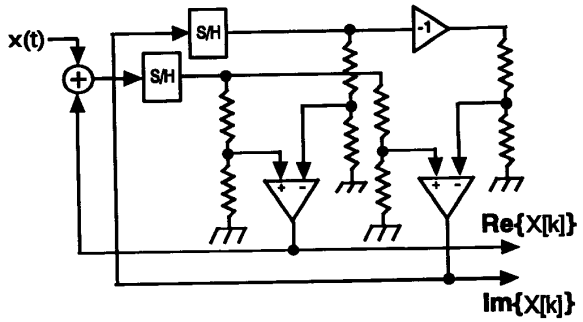


Figure 4: GA Implementaion Example

While active resistors require considerably less space than poly resistors, they are also considerably less accurate [4]. SPICE simulations indicated that voltage dividers constructed of active resistors could be expected to have as much as 10% error with ±1V input [5]. Diffusion resistors were eliminated from consideration primarily because of their high capacitance, though it may be that the consequential speed penalty isn't significant when compared to the S/H circuit's speed. High process variability is another negative feature.

The disadvantages to voltage dividers made from large poly resistors are that, due to their size, they are affected by distributed variations in process parameters, as well as the inaccuracies in contact placement. Limiting the S/H circuits to ±1V signals and ±100uA output current, at least 10KΩ of resistance is necessary. The typical MOSIS 2um Process poly resistance is 20Ω/sq., so resistors around 500 sq. overall length are necessary [6]. With a 2um minimum width and 2um minimum spacing between poly sections, an area approximately 42um to a side is required for each divider. Uncertainty in contact placement is within 0.5um, so the error due to it in the case of this example resistor is in the range of 0.05%.

The S/H circuits can be expected to introduce both offset and non-linearity error. Reasonable estimates for the purpose of simulation are 3% error or less for the non-linearity and an offset of less than 50mV. Methods using EEPROMs currently exist that make it possible to make the S/H circuit very compact and with much higher levels of accuracy [7,8]. This same EEPROM technology may allow the precision necessary to make use of four quadrant multipliers instead of the larger resistive bridges [9,10].

4 Error Analysis

While closed form solutions to the resultant error are obtainable for a given input sequence, they don't give an intuitive understanding into the behavior of the circuit for different input sequences, nor is their acquisition easy. The resultant error is limited, however, and has a maximum over the finite space of all possible bounded input sequences. The actual value of this maximum error provides some measure of the circuit's performance.

Ignoring all but the divider uncertainty, the resultant error at the end of the recursion can be modeled as:

$$error[k] = X_{actual}[k] - X_{ideal}[k]$$

$$= \sum_{n=0}^{N-1} x[n]\,(re^{j\phi}e^{j2\pi k/N})^{N-n} - \sum_{n=0}^{N-1} x[n]\,(e^{j2\pi k/N})^{N-n}$$

$$= \sum_{n=0}^{N-1} x[n]\,(re^{j\phi(N-n)} - 1)\,(e^{j2\pi k/N})^{N-n} \qquad (1)$$

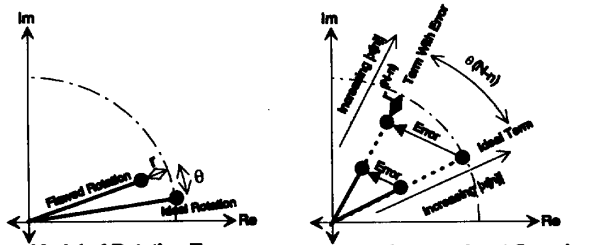
where $re^{j\phi}\approx 1$ models the magnitude and phase inaccuracies in the complex multiplier [Figure 5a]. The magnitude of each term in the summation's error is maximum when $|x[n]|$ is maximum [Figure 5b]. Assuming that the input sequence is bounded, for example restricting $|x[n]|\leq 1$, then for some sequence of x[n]'s with $x[n]=\pm 1$, the complex error terms constructively add to produce the maximum error.

Modeling the effects of the S/H circuit as,

$$f(x) = x - ax^3 + b \qquad (2)$$

where **b** is the offset and **a** the coefficient of the most prevalent term in the non-linearity, the maximum error introduced per term is when $|x|$ is maximum, with the additional constraint that **b** is small. $|x|$ will be largest throughout the recursion when the sequence of x[n]'s is such that its terms are as large as possible and constructively add throughout the recursion. Again, for some sequence of x[n]'s with $x[n]=\pm 1$, the resultant error is at its maximum.

If it wasn't for the non-linearity in the S/H circuit, the offset's effect would not be related to the magnitude of the input sequence, but would instead be an additive constant that could conceivably be removed or compensated for at the end of the recursion. The non-linearity's effect, however, is altered depending on how the magnitude of the result accumulates during the recursion, and this effect is at its maximum when the accumulating result is at its maximum as well. Our approach has been to determine its effect through simulations.



(a) (b)

Figure 5: Voltage Divider Error

5 Simulation

Since the maximum error lies somewhere on the edge of the input space (i.e., when $|x[n]|$ is as large as possible), there are only 2^N candidate input sequences for the one that produces the maximum error. For the example case of N=8, there are 256 sequences that need to be considered in the search for the maximum error.

A carefully optimized C program was written to scan the edge of N=8 input space. The divider error was assumed to be normally distributed, and we concentrated on standard deviations in the range of 0.01% to 5.1%. The investigated range of the 3rd order non linearity coefficient was also 0.0001 to 0.051, with offsets of 0 V, 1mV, and 50mV. At each error combination, 1,000-10,000 circuits were simulated, each circuit having different randomly generated errors for all four of the voltage dividers in the GA circuit. For each of the error combinations, the maximum error along the edge of the input space over all 1,000-10,000 of the randomly generated circuits was found. The results for the offset=50mV simulation are presented in Figure 6.

6 Results

In addition to the maximum error magnitude, the C program retained that error's generating input sequence. As predicted, the maximum error in each case corresponded to an input sequence that maintained the accumulating result at its maximum magnitude. As verification of the hypothesis that this point of maximum accumulation was on the edge of the input space, an N=8 point simulation was also performed that scanned the interior of the input space across 9 points per input sample for each of the N inputs. The 168K fold increase in search space necessitated limiting this simulation to only 10 randomly generated circuits per error combination. As expected, each of the maximum errors in this simulation corresponded to a point where the accumulating result maintained its maximum magnitude, and as a direct consequence, was along the edge of the input space as well.

Of note is that the largest errors in the simulation are for the output points X[0] and X[4]. This is because these two accumulations take place exclusively on the Real axis, and as such have input sequences capable of the maximal accumulation for non-Complex input sequences. This suggests that the circuits that implement X[0] and X[4] should not include the half that performs the Imaginary part of the recursion.

The most noteworthy observation is that the resultant error is essentially linear with respect to the simulated levels of implementation error. This implies that a first order model of the error may work well over this region. Also, as can be seen in the figure, that in such a first order model, the coefficient for the divider error will be larger than that for non-linearity of the S/H circuits by roughly an order of magnitude. Fortunately, the error of poly resistor based dividers is expected to be very low (as discussed above), so that the region our implementation is expected to operate in is somewhere along the *Standard Deviation of Divider Error*= 0.001 line. Also, the S/H offset's contributing error is substantial enough to warrant its minimization.

7 Outlook

Future steps in this research include resolving the issues around the implementations of the S/H circuits and the input adder (including any scaling factor), the fabrication and testing of 8-point GA circuits, and the performance analysis of larger GA based analog DFT/IDFTs such as the three layer N=512 circuit proposed above.

References

[1] G. Goertzel, "An Algorithm for the Evaluation of Finite Trigonometric Series," *American Math. Monthly*, vol. 65, pp. 34-35, January 1958

[2] A. Oppenheim, R. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Inc, New Jersey, pp. 585-87

[3] H. V. Sorensen, C. S. Burrus, D. L. Jones, "A new efficient algorithm for computing a few DFT points," *1988 IEEE International Symposium on Circuits and Systems Proceedings*, vol. 2, pp. 1915-18, June 1988

[4] M. Banu, Y. Tsividis, "Floating Voltage-Controlled Resistors in CMOS Technology," *Electronics Letters*, vol. 18, no. 15, pp. 678-9, 22 July 1982

[5] L. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits," *Memorandum No. M520*, University of California Berekely, May 1975

[6] The MOSIS Project, USC/ISI, 4676 Admiralty Way, Marina del Rey, CA 90292-6695

[7] A. Thomsen, M. Brooke, "A Floating-gate MOSFET with Tunneling Injector Fabricated Using a Standard Double-polysilicon CMOS Process," *IEEE Electron Device Letters*, vol. 12, no. 3, pp. 111-113, March 1991

[8] A. Thomsen, M. Brooke, "A Programmable Piece Wise Linear Large Signal CMOS Amplifier," to appear in *IEEE Journal of Solid State Circuits*

[9] Z. Wang, "A CMOS Four-Quadrant Analog Multiplier with Single-Ended Voltage Output and Improved Temperature Performance," *IEEE Journal of Solid State Circuits*, vol. 26, no. 9, September 1991

[10] L. Carley, "Trimming Analog Circuits Using Floating-Gate Analog MOS Memory," IEEE Journal of Solid State Circuits, vol. 24, pp. 1569-75, December 1989

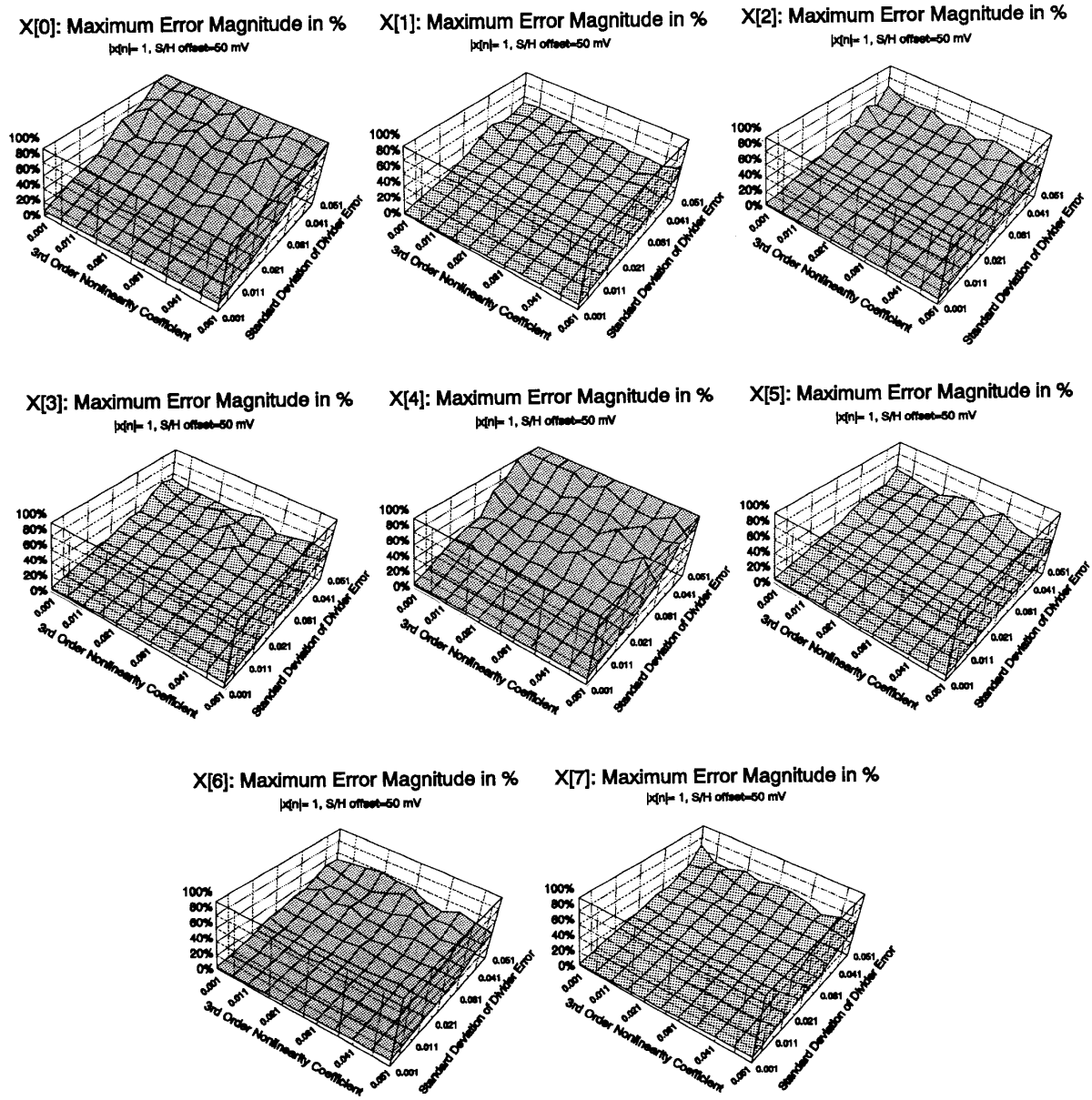


Figure 6: Maximum Error Magnitude for 10K Random Circuits