A MATHEMATICAL APPROACH TO A LOW POWER FFT ARCHITECTURE

Kenneth S. Stevens

Intel
Strategic CAD Labs
Portland, OR 97124

Bruce W. Suter

Air Force Research Laboratory/IFGC
525 Brooks Rd
Rome NY 13441

ABSTRACT

Architecture and circuit design are the two most effective means of reducing power in CMOS VLSI. Mathematical manipulations have been applied to create a power efficient architecture of an FFT. This architecture has been implemented in asynchronous circuit technology that achieves significant power reduction over other FFT architectures. Multirate signal processing concepts are applied to the FFT to localize communication and remove the need for globally shared results in the FFT computation. A novel architecture is produced from the polyphase components that is mapped to an asynchronous implementation. The asynchronous design continues the localization of communication and can be designed using standard cell libraries such as radiation-tolerant libraries for space electronics.

We present a methodology based on multirate signal processing techniques and asynchronous design style that supports significant reduction in power over conventional design practices. A test chip implementing part of this design has been fabricated and power comparisons have been made.

1. INTRODUCTION

The structure of design and its figures of merit have been slowly evolving to keep pace with the the relentless reduction of feature sizes in CMOS technology. While performance remains a primary figure of merit for any design, the increase in transistor count and smaller feature size of each process generation is elevating the importance of other figures of merit. Power, skew, increasing process variations, and increased capacitance of non-local communication are becoming increasingly important challenges to architecture and circuit design.

As design sizes increase, the ability to view a die as a unified circuit controlled by a single frequency becomes less viable. We feel that future architectures will be modular where each section contains its own frequency domain, and where they communicate not on bidirectional shared busses but via point-to-point unidirectional communication links. Such a design has significant power and potential performance advantages when compared to "traditional" architectures.

Recently, we have investigated bringing formal mathematical approaches to bear on the new design realities. We have focused our research on a simpler problem - that of numerical applications - and chosen an application in this domain to investigate the validity and tradeoffs of this approach.

An architecture efficient in terms of performance, power, and communication topology of a common numerical application, the FFT, has been created through mathematical transformations. The mathematical manipulations are done at a very high level directing the transformations in a way that optimizes the design for implementation features to mimic how we consider designs to appear in the near future.

Our approach is to design a methodology by applying formal mathematical rigor to reduce the power complexity of designs through mathematical parallelization techniques that result in modular designs. Then engineering vigor is applied to optimize logic and circuit structure, and to tune the architecture to best fit the particular implementation technology. These techniques can be directly adopted into today's design as well as those with characteristics predicted for future processes[6]. We thus start with a low power architecture, and apply best known engineering practices to obtain significant total power reduction for a wide range of voltage and application domains. Each module in the design is implemented using formal asynchronous protocols. This allows us to implement multiple frequency domains without energy for distributing and dividing a clock, as well as to control "computation gating" at the transistor level. The particular domain we have chosen to implement our test chip is a radiation tolerant library for space applications.

This work is a precursor to investigating more general high performance, low power numerical architectures.

This work is supported by a grant from the Space Technology Directorate of Phillips Laboratory, Kirtland Air Force Base, New Mexico

0-7803-4455-3/98/$10.00 © 1998 IEEE

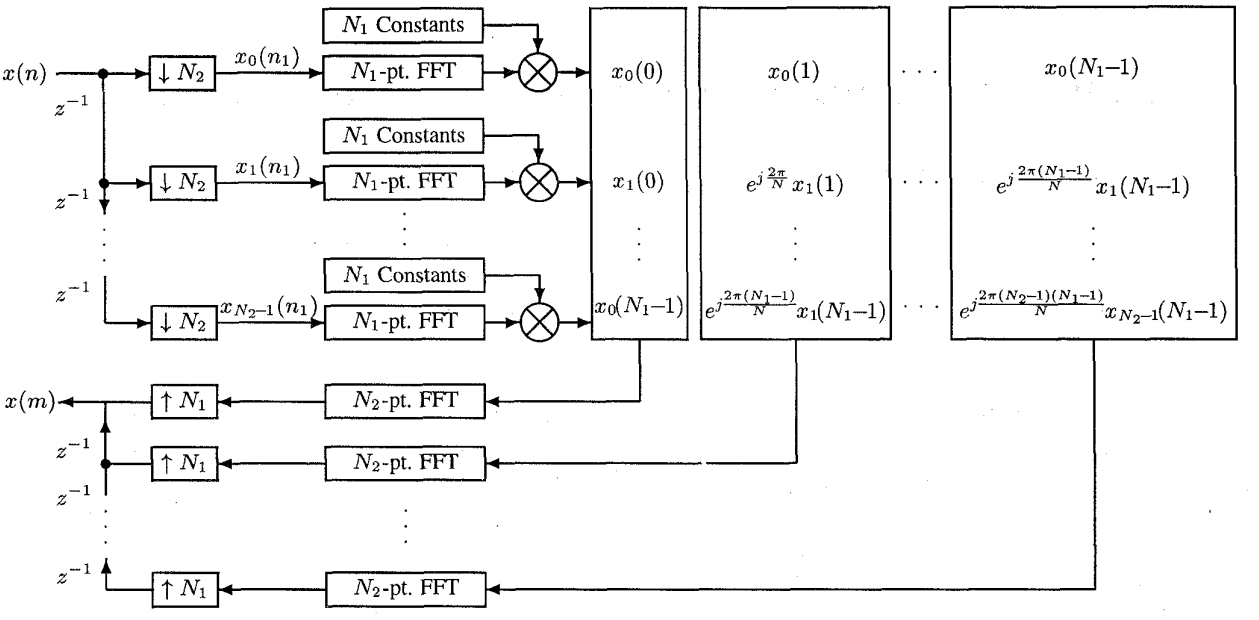

Figure 1: Low Power FFT Architecture

2. MATHEMATICAL APPROACH

Our mathematically approach is hierarchically formed and expressed in terms of the $W_N = exp(-j\frac{2\pi}{N})$ notation as shown in Equation 1. The derivation can be found in [5].

$$X_{m_1}(m_2) = \sum_{n_2=0}^{N_2-1}\left[W_N^{m_1 n_2}\sum_{n_1=0}^{N_1-1} x_{n_2}(n_1)W_{N_1}^{m_1 n_1}\right]W_{N_2}^{m_2 n_2} \quad (1)$$

This notation represents N_2 FFTs using N_1 values as the inner summation, which are scaled and then used to produce N_1 FFTs of N_2 values. The total operation achieves the desired FFT of size N.

Historically, equations for FFT systems similar to our approach have been developed for two applications. In the mid 1960's the problem of computing the FFT of a vector that was too large to fit in main memory was addressed. A approach similar to that presented here was created to limit the storage requirements in these primitive systems[2]. A second similar approach was achieved in the 1980's for multiprocessor applications of the FFT algorithm. The underlying architectures created from these equations are vastly different than that achieved here[1].

3. FFT ARCHITECTURE

The goals of this project are to attempt to take a common application area and investigate novel formal architectural approaches to architect low power and high performance with additional constraints on what we project future designs will require. We therefore emphasized in our formulation pipelining, increasing localization, hierarchy, and establishing multiple frequency domains where we attempt to push the critical path into concurrent frequency domains that support high performance.

The multiplicative complexity of our approach is the same as the conventional Cooley-Tukey FFT formulation, which is $O(NlogN)$. But, our approach permits localized computations, as opposed to globally computing butterflies. This in turn suggests a low power silicon implementation, which is shown in Figure 1.

The multirate formulation of this algorithm has resulted in an implementation parallelized in a pipelined fashion. Each "row" in the architecture contains point-to-point unidirectional data pipelines. The entire design is implemented using asynchronous finite state machines for control.

The frequency of each horizontal track in the architecture operates at $\frac{1}{N_2}$ the frequency of the initial sample rate due to decimation – an average cycle time of 160ns. The asynchronous design methodology allows the rate division to occur locally with much of the circuit idle consuming only leakage current when the operation is complete.

We will assume an ultra high performance sample frequency of 100MHz and a 32 bit word for this design example, where the 32 bit word is split into a 16 bit real component and a 16 bit imaginary component. The word size and frequency can be scaled dependent on the application. We will use a simple 256-point FFT for illustrative purposes due to its regularity, yet hierarchical nature without loss of

generality. For simplicity, choose N_1 and N_2 to be 16 in a 256-point FFT. We also assume a simple four-cycle handshake protocol for all communication in this design. Our 256-point design executes at one 32-bit sample per 10ns. Larger point FFT's have even simpler time constraints due to increased parallelism.

The down arrow blocks of Figure 1 are decimators[4]. The output of the M-fold decimator for a sampled signal $X(n)$ is given by $y(n) = x(Mn)$. This is effectively a demux operation where each output is selected in order. We implement decimation using simple one-hot technology similar to [3]. This simple implementation technique allows us to arbitrarily scale the size of N_1 by adding more one-hot cells. Upon reset the first one-hot is enabled. Upon each data transfer, the enabling token cascades to the next one-hot. The decimator control also drives the new data onto the line $x_i(n_1)$ and send a request to the N_1 FFT cell to process this new data sample. The FFT handshake must complete before the next 16 samples arrive (150ns) or the input stream will be delayed. The decimators contain edge-triggered flip flops so that the data on wires $x_i(n_1)$ will either toggle or remain at the same voltage for each new sample. The largest drawback to this part of the implementation in terms of power and performance is the shared load on the input of a wide, flat decimation. The input load can be greatly reduced by implementing n-way decimators (such as 2 or 4-way) and connecting them in a logarithmic tree. The optimal size of the decimation tree must be investigated for each implementation technology. Future technology will likely require multiple levels of decimation to reduce input load at the required high input frequency.

Each of the N_1 and N_2 blocks represent another FFT operation which can be a hierarchical instantiation of the structure in the figure where the values of $N_1 \times N_2$ equals N_1 or N_2 at the higher level in the hierarchy.

The product blocks multiply a stream of results coming from the N_1 point FFT units by a set of constant values. Both constants and results are complex numbers, requiring four multiplications and two additions per sample. At the top level of the design, where the highest frequency exists, a single complex multiply must occur with an average throughput of 6.25MHz. The constants are calculated by $W_N^{m_1 n_2}$, where $m_1 = 0, \ldots, N_1 - 1$ and $n_2 = 0, \ldots, N_2 - 1$. Therefore all constants in the top product block are unity, and the product logic is replaced with a wire in the asynchronous implementation. Every other product block contains at least one unity constant per set of N_1 constants. The asynchronous implementation of the product logic optimizes for power by bypassing the multipliers when the constants are unity. This block is the performance bottleneck of the architecture, and the size of the top-level decimation dictates the frequency of the multiplications. We implemented compact multipliers which execute a radix-4 Booth algorithm. A novel shared-control implementation is used where, due to the nature of a complex multiplication, we can share a single constant and control block with two ALU's to generate two multiplies concurrently. We therefore require that each multiplication have a duty cycle of 160ns in our target architecture. Figure 2 shows the implementation of the shared-control multiplier.

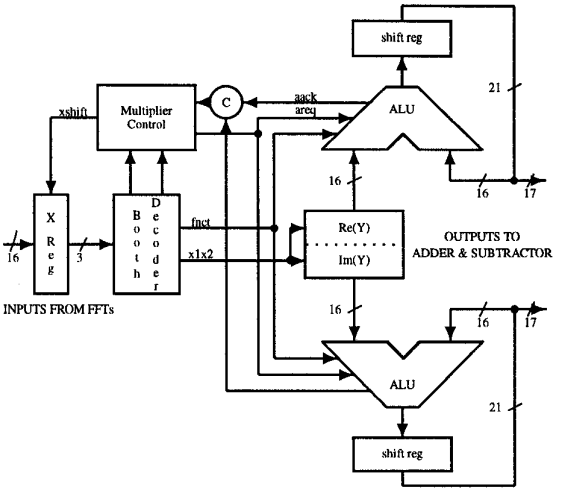

Figure 2: Multiplier Implementation

The large pipeline switch maps results from the product block to the N_2 FFT units. The N_2 FFT units take a transform of time displaced Fourier transform samples. Each N_1-point FFT provides one data sample to each of the N_2-point FFT units, the first row providing the first sample. The pipeline switch logic design is very similar to the design of the decimator cell with three exceptions. First, the input frequency has been scaled down to a request cycle time of 6.25MHz (160ns). Second, rows and columns are enabled using one-hot control in a two dimensional fashion where a token must be received from the top and the left cells. Therefore, $N_1 + N_2$ tokens exist, where two tokens are required to forward data, allowing N_1 concurrent transfers. Upon reset the leftmost and topmost cells each contain tokens. The third difference is that the outputs of the one-hot data drivers must be tristated because each N_2 FFT receives data from all N_1 FFT units. The output load from the product unit requires the request and data signals to drive N_2 (16) minimum sized gates. Both the input load and crossbar load can be driven slowly and use low power logic such as CVSL to reduce energy requirements. However, the results presented in this section use traditional fast slew rate full voltage swing transitions.

The 16-point N_1 and N_2 FFT logic blocks are both hierarchically decomposed into 4×4 FFTs with an architecture identical to that of the top level. However, the 4-point FFT and product blocks operate at a cycle time of 640ns.

Chip	Power per transform
DSP-24 (DSP Arch.)	143μJ / transform
SPIFFEE-1 (Stanford)	50μJ / transform
space FFT	97μJ / transform
earth FFT	18μJ / transform

Table 1: FFT Power Comparison

Each 4-point FFT cell can be implemented exclusively with addition. The low level FFT cell can easily be implemented with 16 registers, some self-timed control, and an adder. The adders and latches are 16 bit devices because they operate on real and imaginary components separately. In the process of design, we discovered that sharing a single adder in the FFT-4 required significant registers to hold intermediate values. Since the size of a register and an adder are basically equivalent, a much simplified and lower power design with the same area can be built by having each addition in the FFT-4 have a dedicated adder. All shared busses are removed and the remaining registers can be traded for some FIFO cells.

A stream of data $x_0(m_2), \ldots, x_{N-1}(m_2)$ is output by the N_2 FFT units to an array of expanders[4]. The output frequency of the expanders increases N_1 fold, with each expander cell providing a single data sample.

4. CONCLUSIONS

A new VLSI design methodology has been presented that illustrates potential the power benefits of a mathematical approach to design for low power implemented using self-timed methodologies. We have designed and submitted to MOSIS a circuit containing the FFT-4 logic using a radiation tolerant cell library. The power consumption of the fabricated FFT-4 and completed implementation of the FFT-16 has been used to estimate the overall power efficiency of a 1024-point FFT. These results are shown in comparison with other FFT designs in Table 1. All designs are measured using a 3.3V voltage source.

The space design uses a radiation tolerant cell library. Due to the radiation constraints, all logic including registers is implemented using static gates and the *minimum* size inverter measures 90λ for the p-fets and 50λ for the n-fets. Similar static implementations with appropriate sized devices requires approximately 15% of the area of the space cells, where minimum size devices can be 4λ for the p and n-fets or smaller. The asynchronous methodology is orthogonal to the device implementation, and dynamic implementations can be designed to save significant power over complementary static CMOS designs. The value for the Earth energy is estimated based on paper designs being compared to the radiation tolerant implementation.

Figure 3 shows a sample control circuit: the domino implementation of one of the two types of one-hot asynchronous finite state machine controllers. This dynamic controller requires only 3 domino inverters and 4 inverters. The sum-of-products equations for the static space version are lreq = greq $\times$ go + greq $\times$ lreq, next = $\overline{\text{greq}} \times$ Y, and Y = $\overline{\text{go}} \times$ lreq + $\overline{\text{greq}} \times$ Y. Control was synthesized using the 3D and MEAT tools, and the latches and data path elements were library cells or hand designed.

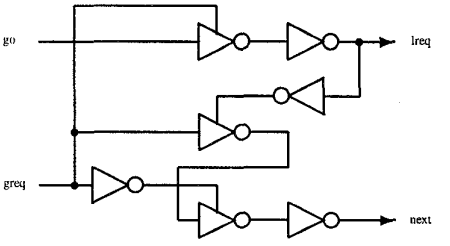

Figure 3: Domino 1-Hot Circuit

We feel that asynchronous design methodologies scale extremely well. When combined with this mathematical architecture an elegant design, with properties of future architectures, is produced with significant power advantages. The implementation results presented here leave substantial headroom for power optimization. We feel that the decreased power and throughput are a combination of greater locality and parallelism which permits frequency domains which are exploited to increase throughput and lower power.

5. REFERENCES

[1] D. Bailey. FFTs in External or Hierarchical Memory. *Journal of Supercomputing*, 4:23–35, 1990.

[2] W. Gentleman and G. Sande. Fast Fourier Transforms for Fun and Profit. In *AFIPS Conference Proceedings*, volume 29, pages 563–578, 1966.

[3] Lee Hollaar. Direct Implementation of Asynchronous Control Units. *IEEE Transactions on Computers*, C-31(2), February 1982.

[4] Bruce W. Suter. *Multirate and Wavelet Signal Processing*. Academic Press, 1997.

[5] Bruce W. Suter and Kenneth S. Stevens. Low Power, High Performance FFT Design. In A. Sydow, editor, *Proceedings of IMACS World Congress on Scientific Computation, Modeling, and Applied Mathematics*, number 1, pages 99–104, June 1997.

[6] The National Technology Roadmap for Semiconductors. 1997 edition. Technical report, Semiconductor Industry Association, 1997.