An Energy-Efficient Single-Chip FFT Processor

Bevan M. Baas

Department of Electrical Engineering, Stanford University, Stanford, CA 94305-4055

Abstract

A 1024-point single-chip Fast Fourier Transform processor that employs algorithm, architecture, and circuit techniques to achieve an energy efficiency over 60x greater than the best known published processor, is presented. It is designed to operate at supply voltages less than 400mV in a low threshold-voltage CMOS technology, and uses a unique “data-caching” memory architecture and “hierarchical bitline” memories to reduce power.

Introduction and Design Goals

For many CMOS circuits, the switching energy is proportional to the supply voltage squared. It is therefore possible to dramatically reduce energy consumption by lowering the supply voltage [1]. However, to retain good performance, the transistor threshold voltages (Vt) must also be lowered [2].

This paper describes a Fast Fourier Transform (FFT) processor designed to operate at supply voltages below 400mV in a low-Vt CMOS process, similar to one that has produced circuits operating down to 125mV [3].

Several design goals were set for the processor. Because the amount of noise on a large chip running at very low supply voltages is not known, circuits were designed to perform robustly in a noisy (relative to Vdd) environment. Emphasis was placed on achieving a high energy efficiency with a lesser emphasis on performance, because throughput can be easily increased through the use of parallel processors. Regular data flow and simple control were also high priority design goals.

Algorithm and Architecture

The FFT is an efficient method of computing the Discrete Fourier Transform [4]. Although it is not the most efficient in terms of number of operations, the radix-2 Decimation In Time (DIT) form of the FFT was chosen because its regularity and simplicity make it attractive for VLSI implementation. The basic unit of computation is a *butterfly* where two outputs (X, Y) are calculated from two data inputs (A, B) and a complex coefficient W: $X = A + BW$, $Y = A - BW$. In general, all data is complex, so each butterfly requires 4 multiplies, 6 add/subtracts, 2 reads, 2 writes, and 1 coefficient read. Fig. 1 is a dataflow diagram for a straight-forward FFT.

Energy consumed by a processor can be divided into two groups: energy used by functional units to actually perform work, and energy used to support that work. A large part of the latter includes energy used to communicate data to and from storage elements. This energy could be reduced if a smaller set of the working data were stored in a separate memory much smaller than the N-word main memory. Because of its size, this *data cache* would be lower energy, closer to the datapath, and faster than the main memory.

Because of its global communication, the regular dataflow pattern of Fig. 1 is not suited for caching. However, if we re-draw the dataflow diagram with an extra step in the middle as shown in Fig. 2, the transform is now easy to cache. While the global communication can never be removed, this algorithm concentrates it into one step (in this example). This works well in a cached processor because this interchange is done while the caches are being dumped and loaded.

We define an *epoch* as the period of the FFT where each of the N words in main memory are loaded into the cache, processed, and written back once. Given an N-point radix-2 transform, E epochs, and a power-of-2 cache size; we can find the cache size C from (1). Designs with $N = C^E$ are consistent across epochs and will therefore have simpler control.

$$C = 2^{\left\lceil \log_2 \sqrt[E]{N} \right\rceil} \tag{1}$$

FFT algorithms that cache data are not new; in fact, a type of caching algorithm was used to cache data from a magnetic drum using a core memory [5]. Our design is unique, however, in that it is designed to minimize energy. This is achieved both through having a small cache, and by designing a more efficient main memory since it will run at a fraction of its former speed.

Low Vt Circuits

From a circuit design perspective, the biggest difference when using low-Vt transistors at a low supply voltage is leakage. The difference between the current of an “on” transistor and an “off” transistor is much less; which can cause high fan-in circuits and certain dynamic circuits to fail. Probably the most common high fan-in circuit is a bitline in a memory. During a read operation of an *M*-word memory, *M-1* cells are off but slightly leaking, and one cell is trying to swing the bitlines.

We use a *hierarchical bitline* structure to reduce the effective fan-in of memory bitlines in both ROM and SRAM memories. Fig. 3 shows the circuit used for a 128-word 36-bit SRAM. Transistors M1 precharge both gbl and lbl to Vdd. PMOS transistors M2 are very weak and are used during low-speed operation. With 16 words per lbl, the area overhead of the pass gates is 6% for the SRAM and 34% for the ROM. Although the absolute area required is nearly the same for both memories, the relative area penalty is much larger for the ROM because the cell size is so much smaller. Another added benefit is that for many memory sizes, the bitline capacitance can be nearly halved compared to a non-hierarchical-bitline design. Fig. 4 is a spice simulation of a non-hierarchical-bitline 128-word SRAM read failing at Vdd=300mV because of leakage on bl_ under worse-case conditions. Fig. 5 shows a read succeeding on the same SRAM with hierarchical bitlines. Simulations show that varying the nwell bias on the M3 PMOS pair reduces the wordline -> out time by as much as 45%. For this reason, the node Vbp_sa is biased separately from other nwells.

Results and Conclusion

A single-chip, 1024-point, radix-2, DIT FFT processor has been designed. The algorithm computes an FFT over 2 epochs, so the cache size is 32 words ($\sqrt[2]{1024}$). The caches are organized as two banks of two sets of 16-word dual-ported memories. There is a single non-iterative datapath that computes 1 butterfly per cycle. Fig. 6 shows the 9-stage pipeline. The processor operates on 36-bit complex data, has 20-24bit internal datapaths, and contains 460,000 transistors. Switch-level simulations based on extracted layout estimate 4% of the total dynamic power will be consumed in the main memory and 19% in the caches. In a 0.50μm low-Vt process, simulations predict 85MHz operation at Vdd=400mV. Table 1 contains a comparison with several other FFT processors. The last column contains a figure of merit which estimates energy efficiency and attempts to factor out process technology and datapath width. Even though this weighting favors designs with narrow datapaths, the processor presented here is 60x more efficient than the previously best known processor.

The processor was recently fabricated in a standard high-Vt 0.7μm (L=0.6μm) process and is now in test. Fig. 8 is a microphotograph of the 5.985mm x 8.204mm die.

Acknowledgments

This research was supported by NASA (NGT-70340). The author would also like to thank Jim Burr and Masataka Matsui for valuable guidance and suggestions.

References

[1] A. Chandrakasan, S. Sheng, R. Brodersen, "Low-power CMOS digital design," *IEEE JSSC*, vol. 27, No. 4, April 1992.

[2] J. Burr, A. Peterson, "Energy considerations in multichip-module based multiprocessors," *IEEE Intl. Conference on Computer Design*, pp. 593-600, 1991.

[3] J. Burr, "A 200mV self-testing encoder/decoder using Stanford ultra-low-power CMOS," *IEEE ISSCC*, vol. 37, pp. 84-85, February 1994.

[4] L. Rabiner, B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1975, pp. 361.

[5] N. Brenner, "Fast Fourier transform of externally stored data," *IEEE Trans. on Audio and Elect.*, vol. AU-17, No. 2, pp. 128-132, June 1969.

[6] E. Bidet, D. Castelain, C. Joanblanq, P. Senn, "A fast single-chip implementation of 8192 complex point FFT," *IEEE JSSC*, vol. 30, pp. 300-305, March 1995

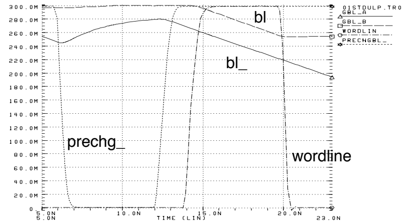

Fig. 4 Read failure with non-hierarchical-bitline SRAM

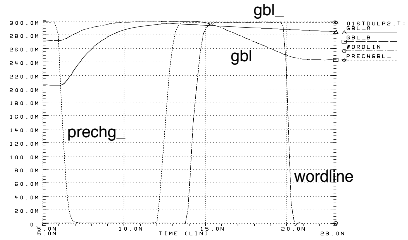

Fig. 5 Successful read with hierarchical-bitline SRAM

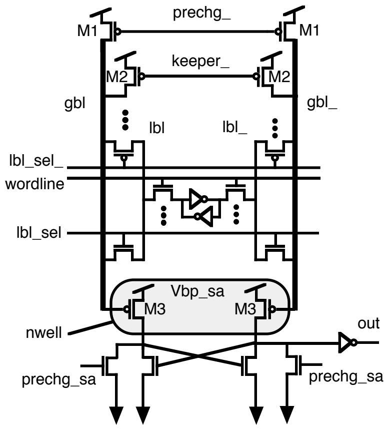

Fig. 3 Schematic of SRAM with hierarchical bitlines

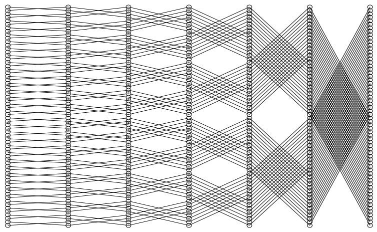

Fig. 1. Regular N=64, radix-2, DIT FFT dataflow diagram

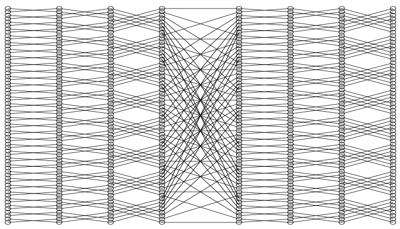

Fig. 2. Proposed N=64, radix-2, DIT dataflow diagram suited for caching

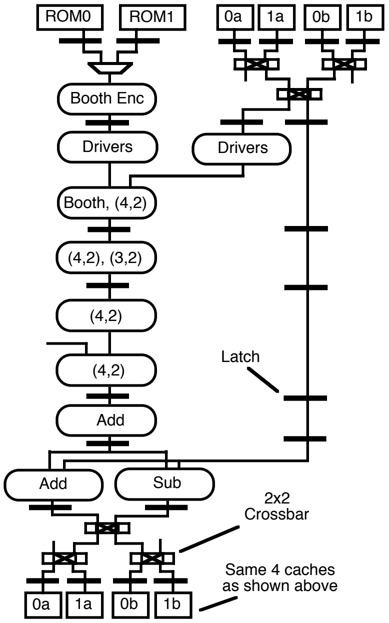

Fig. 6 Partial pipeline diagram of FFT processor

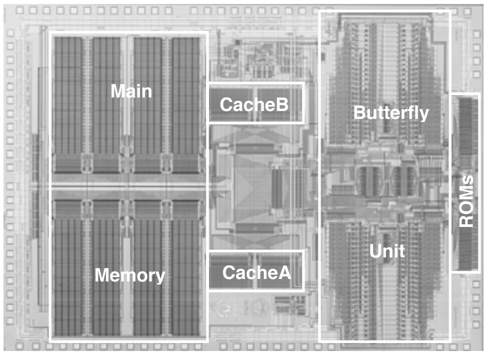

Fig. 8. Die microphotograph

Table 1 Comparison of most efficient known FFT processors

Processor	Technology (μm)	Datapath width (bits)	Supply voltage (V)	Throughput (1024-pt xforms/s)	Power (mW)	Adjusted transforms/mJ (Thruput*Tech*Dpath/Pwr/10)
Plessey PDSP16510A	1.4μm	16	5.0	10,200	3000	8
E. Bidet [6]	0.5μm	10	3.3	19,531	300	33
This work	0.5μm	20	0.4	16,265	8	2030