

# Energy-Aware Architectures for a Real-Valued FFT Implementation

Alice Wang and Anantha P. Chandrakasan

Department of EECS  
Massachusetts Institute of Technology,  
Cambridge, MA 02139

{aliwang, anantha}@mit.edu

## ABSTRACT

Energy-aware design is highly desirable for systems that encounter a wide diversity of operating scenarios. This is in contrast to traditional low power design for the worst case scenario, which may not be globally energy efficient. Energy-aware design focuses on enabling architectures which scale down energy as quality requirements are relaxed. A new energy-scalable system design methodology is proposed for a Real-Valued FFT processor which supports variable bit precision (8 and 16-bit precision) and variable FFT length (128- 512 point). Two energy-aware architectures, Ensemble of Point Solutions method and Reuse of Point Solutions method, are described and evaluated. Simulated and measured results show a 66% energy savings for 8-bit datapath and 52% savings for 128-point FFT length over a non-scalable approach.

## Categories and Subject Descriptors

B.4.1 [Data Communications Devices]: Processors. B.6 [Logic Design] B.7[Integrated Circuits]

## General Terms

Algorithms, Performance, Design.

## Keywords

energy-awareness, energy-quality scalability, wireless sensor networks, microsensors, fast fourier transform (FFT), scalable architectures, source tracking and localization, energy efficient, digital signal processors (DSP).

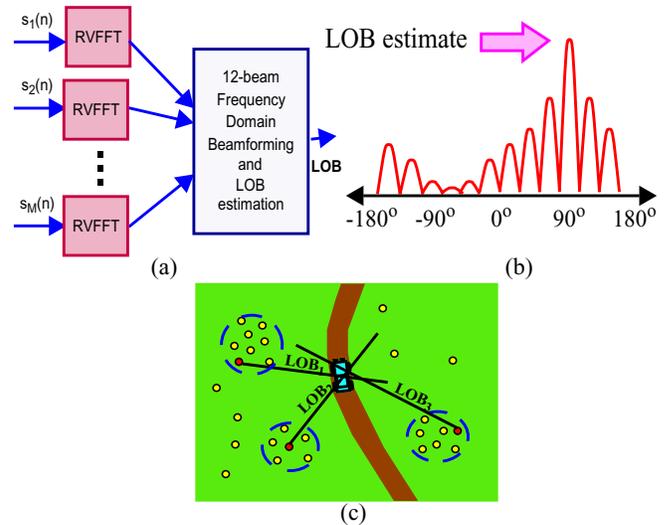
## 1. INTRODUCTION

Energy efficient digital signal processors (DSP's) are becoming increasingly important with the growth of portable, wireless, battery-operated appliances such as cellular phones, Personal Digital Assistants (PDAs) and laptops. One such application is wireless sensor networks, where tens to thousands of battery-operated microsensors are deployed remotely and used to relay sensing data to the end-user. Given

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISLPED '03, August 25-27, 2003, Seoul, Korea.

Copyright 2003 ACM 1-58113-682-X/03/0008...\$5.00.



**Figure 1. The FFT can be used in sensor signal processing for source tracking through multiple Line of Bearing (LOB) Estimation (a). LOB estimation uses the FFT in frequency domain beamforming (b) and estimates the LOB to be the direction with maximum energy (c).**

the constantly changing environments of portable devices and the extreme constraints on battery lifetimes, system level energy-aware design considerations should be taken into account.

Energy-aware design is in contrast to low power design, which targets the worst case scenario and may not be globally optimal for systems with varying conditions. A new metric for design is to maximize energy-awareness [1]. The energy-awareness of a system can be increased by adding additional hardware to cover functionality over many scenarios of interest and to tune the hardware such that over a range of scenarios, the system is energy-efficient.

One algorithm that is widely used in sensor and wireless applications is the Fast Fourier Transform (FFT). In the area of sensor signal processing, the FFT is used in frequency domain beamforming, source tracking, harmonic line association and classification. For example, in Figure 1, the FFT is used in source tracking via Line of Bearing (LOB) estimation of multiple sensors. First,  $M$  sensor signals are transformed via FFT into the frequency domain for beamforming. Next, 12-beam beamforming is performed and the LOB is estimated to be that beamformer output with the maximum signal energy (Figure 1a and 1b). Multiple LOB's can be used to triangulate the source's location (Figure 1c).

Energy-quality scalability for an system is needed if the environment of the device changes constantly. An energy-aware FFT will be able to adapt energy consumption as energy resources of the system diminish or as performance requirements change. Therefore it is advantageous to design the FFT with energy scalability hooks such as variable memory size and variable bit precision, so that it can be used for a variety of scenarios. Our design focuses on a real-valued FFT (RVFFT) which can scale between 128-512-point FFT lengths and can operate at both 8 and 16-bit precision computation.

In this work two energy-aware architecture designs are discussed. These architectures are evaluated in the context of a variable bit precision and variable FFT length RVFFT. The scalability of the RVFFT was implemented in a 0.18μm process for energy-awareness measurements and hardware verification.

## 2. REAL-VALUED FAST FOURIER TRANSFORM (RVFFT)

The FFT is an efficient way to transform a length-N complex sequence into the frequency spectrum, making it one of the more widely used signal processing algorithms. For example, the FFT is an integral part of the frequency-domain delay-and-sum beamforming algorithm used in Line of Bearing estimation for acoustic sensor systems.

The complex-valued FFT (CVFFT) of a sequence  $s(n)$  is defined to be

$$S(k) = \sum_{n=0}^{N-1} s(n)W_N^{kn} \quad k=0,1, \dots, N-1 \quad (1)$$

where  $W_N^{kn} = e^{-j2\pi kn/N}$ . The CVFFT equation assumes that the input and output are both complex sequences. Since the data for most sensor applications is real-valued, the imaginary half of the  $s(n)$  are zero's and half of  $S(k)$  are redundant, due to its symmetrical properties. The inherent symmetry can be exploited to compute one N-pt. Real-Valued FFT (RVFFT) from one N/2-pt. CVFFT [2], by taking the CVFFT of a complex signal  $z(n)=s(2n)+j*s(2n+1)$ , created from the even indexed inputs and the odd indexed inputs of  $s(n)$ . The last stage of computation uses the symmetry to compute the relevant  $N$  RVFFT coefficients from the  $N/2$  CVFFT coefficients.

### 2.1 RVFFT Architecture

The block diagram of an RVFFT architecture, is shown in Figure 2, and has four functional blocks: Butterfly datapath, Data Memory, Twiddle ROM's and Control logic. The label beneath each block indicates the energy-scalable function enabled within the block. The architecture is designed to perform either a N-pt. RVFFT or an N/2-pt. CVFFT where  $N=128-512$ . Also the bit precision can be either 8 or 16 bit.

The data is clocked into the memory through the 12-bit DATAIN port and reordered so that upon output, the FFT coefficients are ordered from the lowest to highest frequency in the memory. The memory consists of two data buffers, so that while the FFT computation is being performed with one memory block, the second memory block is buffering up the next sequence of data. Thus, the latency of performing one FFT is reduced. During the FFT computation, in one clock

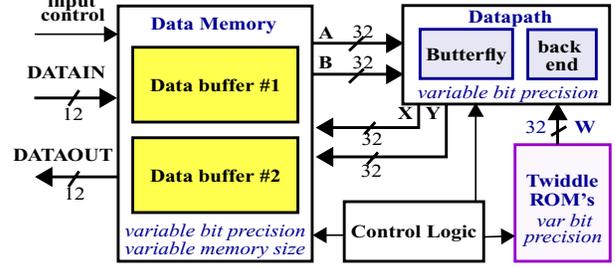


Figure 2. A block diagram of the RVFFT architecture. Each block has an additional label which indicates the energy-scalable feature(s) enabled within the block.

cycle, two complex values (A,B) are read from the memory, one complex twiddle factor (W) is read from the Twiddle ROM's, and a butterfly computation is performed. The butterfly outputs (X,Y) are written back to memory on the next clock cycle. In the last stage of computation, the backend RVFFT hardware is enabled. When all butterfly operations are completed, the output is placed on the 12-bit DATAOUT port.

Desired energy-scalable hooks for the RVFFT are FFT length and bit precision. As the FFT length is increased the frequency resolution is improved, which leads to better analysis of the sensor data. As the bit precision is increased, the accuracy of the FFT coefficients is improved.

### 2.2 Benchmarking the RVFFT on a Microprocessor and FPGA

The general purpose microprocessor is widely used for its ease of implementation and flexibility. Scalable algorithms can be adapted to run in software on current low-power microprocessors (e.g., StrongARM). One drawback to using an microprocessor implementations is the limit to available energy hooks. Thus another fabric, the Field Programmable Gate Array (FPGA) can also be used to implement energy-scalable systems. This reconfigurable hardware is highly flexible and facilitates the implementation of a variety of different algorithms.

Application Specific Energy-Scalable hardware is preferable to both the general purpose microprocessor and FPGA's, because the hardware is tailored to specific functions needed to perform the algorithm. This leads to both better performance and lower energy dissipation because the voltage supply can be aggressively scaled for minimal energy dissipation [3]. Table 1 shows a comparison of the energy dissipation of the RVFFT implemented on the StrongARM-1100 microprocessor, on a Xilinx FPGA and on an application specific energy-scalable hardware system. The energy for the microprocessor and the FPGA are nearly the same, but the application specific energy-scalable hardware dissipated two orders of magnitude less energy. For applications that require extremely low energy dissipation, application specific hardware is critical.

Table 1. Comparing energy dissipation of the RVFFT

	128 pt	256 pt	512 pt.
StrongARM	16 μJ	37 μJ	82 μJ
Xilinx	17 μJ	40 μJ	89 μJ
Custom	<b>116 nJ</b>	<b>269 nJ</b>	<b>628 nJ</b>

### 3. ENERGY-AWARE ARCHITECTURE DESIGN

The energy-awareness of a system can be improved by adding new scenarios over which the system can operate. However, as the number of scenarios increase, the additional control logic complexity for enabling scalability leads to increased energy dissipation and area overhead. We propose a methodology to improve energy-awareness without significant overhead cost.

The breakdown of the energy consumption of the custom RVFFT shows that the datapath consumes 52% of the energy and memories consume 44% of the energy. Since the majority of the energy is consumed by the datapath and memory, the largest impact of energy-aware design can be achieved by focusing on enabling energy-scalable hooks in the datapath and the memory.

One proposed method of designing energy-scalable systems is to use the *ensemble of point solutions method* [1]. This method consists of multiple point solutions, each optimized to a particular scenario, and an ensemble is chosen so as to cover the entire scenario space. Then low-overhead routing and control is added to route the inputs to the correct point solution. Also clock and data gating is used to avoid unnecessary switching in unused point solutions.

A more area efficient design methodology proposed is the *reuse of point solutions method*. In most scalable systems, scenarios of interest are similar enough so common functional blocks can be shared and reused. Thus the design of all functional blocks are tightly integrated to avoid the additional area penalty. In this section, further analysis and comparison between these two architectural design methods are performed for the datapath and the memory design of the RVFFT.

#### 3.1 Energy-Aware Datapath

The datapath of the RVFFT is shown in Figure 3. The main engine of the datapath is the CVFFT butterfly, which contains a complex valued multiplication followed by complex addition/subtraction. The function of interest is

$$X = A+B*W \text{ and } Y=A-B*W, \quad (2)$$

where A, B and W are complex inputs and X and Y are the complex outputs. This is performed for  $n$  stages of the  $2^n$ -pt. CVFFT and for  $n-1$  stages of the  $2^n$ -pt. RVFFT. In the last stage of the RVFFT, the inputs are fed into the backend processing block which performs,

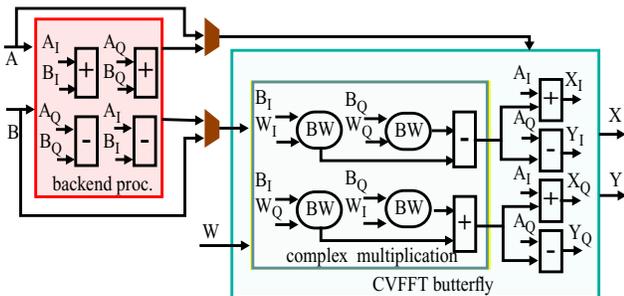


Figure 3. FFT butterfly and backend processing datapath

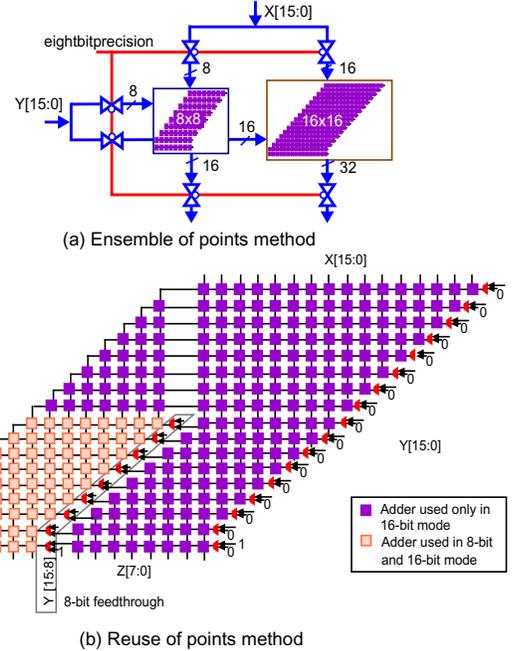


Figure 4. 8-bit and 16-bit scalable Baugh Wooley Multiplier using the ensemble of points method (a) and the reuse of point solution method (b). In the reuse method the 8-bit multiplier is reused for the 16-bit multiplication, thereby adding scalability without a large area penalty.

$$A_{\text{backend}} = (A_I+B_I) + j (A_Q-B_Q) \quad (3)$$

$$B_{\text{backend}} = (A_Q+B_Q) + j (A_I-B_I). \quad (4)$$

where  $A_I$  and  $A_Q$  represent the real and imaginary parts of complex value A. The backend computation block is only enabled for the last stage of computation. Complex multiplication consists of four Baugh Wooley multipliers for two's complement multiplication. In the butterfly datapath design 8 and 16 bit precision scaling is enabled.

An example of energy-scalable bit precision datapath design is showcased by the Baugh Wooley (BW) multiplier design for two's complement arithmetic. A non-scalable design would optimize the worst case scenario by building a single 16-bit BW multiplier. An ensemble of point solutions implementation contains both an 8-bit and a 16-bit BW multiplier and have input gating to route the data to the appropriate hardware (Figure 4a). A better scalable BW multiplier design is shown in Figure 4b. When 16-bit multiplication is needed, the entire multiplier is used. However, if only 8-bit multiplication is needed, the 8-bit feedthrough logic is enabled and the Y inputs are gated. Therefore only the adders in the lower left are driven, thus reusing hardware from the 16-bit multiplier.

Design parameters and energy simulations in *nanosim* of the different butterfly datapath architectures are shown in Table 2. The reuse method leads to a more area efficient design of the butterfly computation than the ensemble of point solution method. There is a 39% reduction in area and 21% fewer transistors needed because the entire 8 bit datapath could be reused for the 16 bit datapath. However, the extra control logic of energy scalable designs incur an area and device count penalty when compared to a non-scalable design.

In Table 2, the reuse of point solution design methodology proves to be globally more energy efficient than a non-scalable solution. For the 8-bit butterfly datapath, the data and input gating to the unused hardware leads to a 66% energy dissipation reduction over a non-scalable design. However, for 16-bit computations, the energy dissipation of the scalable designs increases over the non-scalable design, due to the energy overhead from the energy-scalability enabling logic. Nevertheless, scalable design is globally more energy-efficient than the non-scalable datapath as lower bit precision results are required.

**Table 2. Comparing different RVFFT datapath implementations**

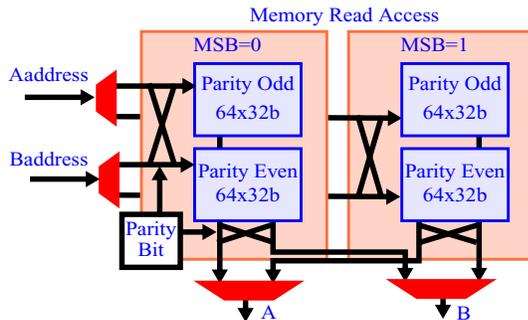
	Non-scalable	Scalable	
		Ensemble of Points	Reuse of Points
Area	0.13 mm <sup>2</sup>	0.23 mm <sup>2</sup>	0.14 mm <sup>2</sup>
Transistor Count	47K	63K	50K
Energy (16-bit, 512pt.)	331.9 nJ	351.4 nJ	341.2 nJ
Energy (8-bit, 512pt.)	287.9 nJ	165.3 nJ	96.4 nJ

### 3.2 Energy-Aware Memories

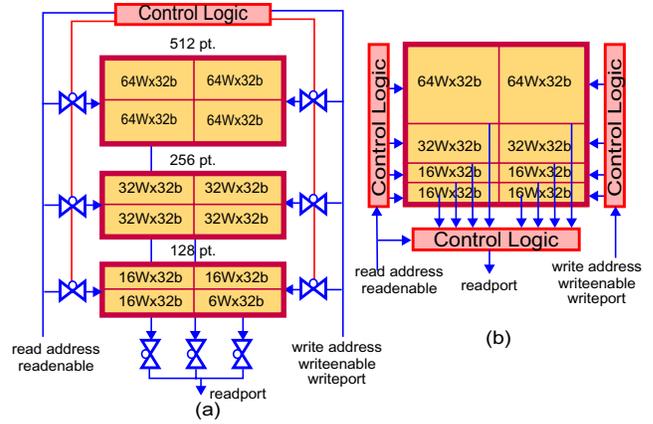
There are two data memory banks for concurrent data input and FFT computation. Each bank is able to read and write two complex values or four real values in one clock cycle. Also 32-bit complex twiddle factors are stored in scalable ROM's.

The memory banks for the RVFFT are designed to avoid memory hazards when accessing two complex values from the memory. Analysis of the memory accesses for the CVFFT show that for every butterfly the memory addresses only differ in their parity [4]. Therefore, the N-point memory can be split into two blocks, and a parity crossbar configuration can be used to access the memory while avoiding memory hazards during the CVFFT. For the RVFFT backend processing, the indices to the memory differ not by parity, but by the most significant bit (MSB). This leads us to partition each bank further, so that there are four banks, each addressable by their address parity and by the MSB. This memory architecture with MSB/parity crossbar is shown in Figure 5 for a 512-pt. RVFFT. Each partition is 64 Word addressable and stores a 32-bit complex number.

The memory designed for the 512-pt. RVFFT can be configured for 128 and 256 pt. processing, but would incur



**Figure 5. Cross Bar and MSB control to ensure parallel read and write memory access for a 512-pt. Real Valued FFT.**



**Figure 6. Ensemble of point solutions (a) and Reuse of point solutions (b) scalable memory.**

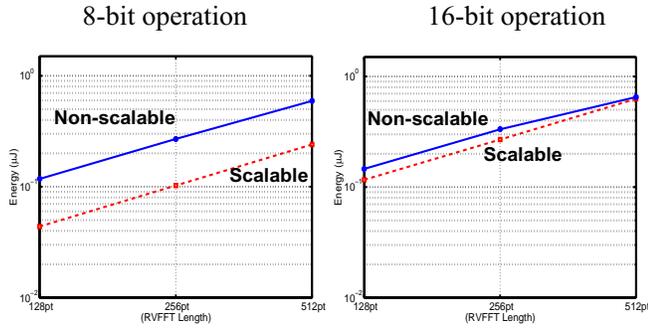
significant power overhead over optimized point solutions. Variable sized memories can be used for more efficient energy-awareness with different FFT lengths. For variable memory design, either an ensemble of point solutions or reuse of point solutions is proposed. For the ensemble of point solutions, the memory architecture for one scenario is created, then duplicated and scaled for all other scenarios. Muxes are then used to route the inputs and clocks to the appropriate memory block, and gate the inputs and clocks to the memories which are not accessed. This design can be seen in Figure 6a.

Using the reuse of point solutions method, the memory is partitioned internally. This memory structure has more complicated control logic which selects the inputs and addresses based on the read/write address. The advantage of internal partitioning is much less area required for the layout, at the expense of more complex control logic design and more energy switched due to the control logic. Figure 6b shows a memory block designed with the reuse of point solutions method for the 128-512 point RVFFT application.

The two memory architectures were implemented with the High-Speed Dual-Port Register File Generator and the control logic was implemented using the Standard Cell Library both provided by Artisan Components [6][7]. Simulations at 1.5V operation were performed in order to evaluate the two scalable memory architectures. The ensemble of point scenario implementation has much higher transistor count (1.8x) than the reuse of point solution implementation. However, the reuse of points design does have a higher transistor count (1.3x) overhead when compared to a non-scalable implementation.

The energy dissipation of the three memory implementations were simulated in *nanosim*. When compared to a non-scalable design, the greatest impact on energy dissipation is seen for smaller FFT lengths. The overhead of accessing large memories during 128-pt. computation leads to energy savings of 52% and during 256-pt. computation leads to energy savings of 49%.

Variable bit precision of 8 and 16 bit precision was also enabled. Each 32 bit memory block was created from two 16 bit memories. The inputs and clocks were gated for 8-bit operation. The Twiddle ROM's were also designed for variable bit precision operation in a similar manner. Variable bit precision leads to a 57% energy savings at 8-bit precision operation.



**Figure 7. Energy-Scalable N-pt. RVFFT over the scenarios of N=128-512 pt. and for 8 and 16 bit computation shows the scalable RVFFT to be more energy-efficient than a non-scalable design.**

### 3.3 Energy-Scalable Systems

System level scalability incorporates the scalability of the datapath and memory with the control logic. The reuse of point solution method was used in the design of the RVFFT. The reasons were less area overhead, fewer transistors and lower energy dissipation. System level ensemble of point solution method would imply that multiple RVFFT modules be built for each scenario, but in the reuse of point solution method, each functional block is designed using the reuse method and brought together with appropriate control logic and an energy-scalable interface.

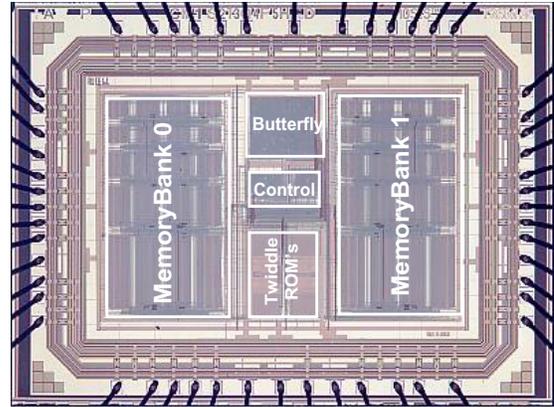
The energy-quality scalability characteristics of the RVFFT are shown in Figure 7, for 128-512 pt FFT lengths and for 8 and 16 bit precision computation. In these figures, we compare the energy and quality of the scalable RVFFT to a non-scalable RVFFT. The non-scalable FFT length architecture is designed for the worst-case scenario (512-pt., 16-bit). The simulations show that there is a definite advantage for an energy-scalable architecture. The scalable architecture is more energy-efficient for all but the high-quality point (512 pt., 16 bit). At the high quality point, there is no energy advantage of the scalable architecture, but rather a disadvantage due to the overhead of the scalability logic.

## 4. PERFORMANCE MEASUREMENTS

The Energy-aware RVFFT was designed and implemented in a 0.18  $\mu\text{m}$  process. Figure 8 shows a die photograph of the scalable RVFFT chip. The chip's functional blocks (memory, butterfly, control, twiddle ROM's) are clearly marked. The datapath and control logic are generated from Artisan Standard Cell Library [5] using the *Synopsys* automatic layout generation design flow. For the memory implementation, the design incorporates Artisan memory generators (Dual-Port Register Files [6] and Single-Port ROM's [7]). Additional process details are shown in Table 3.

**Table 3. Process details**

Process	0.18 $\mu\text{m}$
Chip dimensions	2.1 mm x 3.0 mm
Total transistor count	622K
Metal layers	6



**Figure 8. Die photograph of the energy-aware RVFFT chip.**

The RVFFT was designed using an algorithm hardware design flow incorporating algorithm benchmarking, verilog design and automatic layout generation. Benchmarking in *MATLAB* is performed in order to evaluate algorithmic design choices, such as transforming the RVFFT from floating point to a fixed point processor with multiple bitwidths. Next, the behavioral *MATLAB* benchmark is translated into structural verilog, where scalable hardware is created from a standard cell library. Finally, by incorporating test vectors from the verilog testbench, the functionality of the entire RVFFT algorithm for a variety of input signals is verified using Epic's *nanosim* simulator. This tool is also instrumental in evaluating and comparing energy efficiency of different scalable designs.

The functionality of the RVFFT chip has been fully verified for up to 512 point, for 8 and 16 bit operation. The chip operates between 1.0-2.0V and between 250kHz and 50 MHz clock speeds. Table 4 shows the measured energy dissipation of the RVFFT in comparison to the numbers from the *nanosim* simulation. The simulated energy dissipation is lower than the measured energy for the 8-bit RVFFT and higher than the measured energy for the 16-bit RVFFT.

**Table 4. Measured vs. simulated energy dissipation.**

	8-bit		16-bit	
	meas	sim	meas	sim
128 pt.	46nJ	43nJ	81nJ	116nJ
256 pt.	121nJ	102nJ	216nJ	269nJ
512 pt.	304nJ	240nJ	564nJ	628nJ

When comparing this design to other FFT implementations, it is important to normalize over different technologies and operating voltages. Table 5 shows the comparison of the energy-scalable RVFFT and current implementations. At the worst case operating point (512 pt., 16 bit) the Energy-Scalable RVFFT proposed has lower energy dissipation than other implementations. The energy awareness of this implementation is demonstrated by the slope of the energy curves. The slope of the 16-bit and 8-bit energy is steeper than the other implementations as the FFT length is reduced to 256 and 128 point operation, due to the energy-scalability datapath

and memories. These results show that our implementation has better energy scalability characteristics, thus making this design globally energy efficient than other designs.

**Table 5. Comparing the energy dissipation at 512 pt. FFT across different implementations**

FFT implementation	Energy dissipation for 512 pt.
Wosnitza [8]	8925 $\mu$ J
DoubleBW [9]	2834 $\mu$ J
DSP-24 [10]	1822 nJ
Spiffiee [11]	781 nJ
Energy-Scalable 16-bit	564 nJ
Energy-Scalable 8-bit	304 nJ

## 5. CONCLUSIONS

Energy-aware design provides a new methodology for systems that require low power dissipation in highly changing environments. This is especially true for wireless sensors, which demand extended battery lifetime and are required to operate over many different scenarios. In this work, two different energy-aware design methods were evaluated: ensemble of point solutions method and reuse of point solutions method. The reuse of point solutions technique was preferred due to lower area overhead and lower energy dissipation at the expense of complex control logic design.

The reuse of point solutions method was used to design an energy-aware RVFFT. The RVFFT is an algorithm widely used in a variety of applications. The two scalability hooks considered in the design of the RVFFT datapath and memories were variable FFT length and variable bit precision. Simulations verified energy efficiency of a scalable architecture over non-scalable architecture.

The RVFFT was fabricated using a standard cell library for a 0.18 $\mu$ m process and the functionality was verified. Measured results showed our design to have better energy-scalability characteristics, and thus be more globally efficient than other current implementations over a variety of scenarios.

## 6. ACKNOWLEDGMENTS

This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research

Laboratory, under agreement number F33615-02-2-4005. Alice Wang is supported by the Intel Corporation Fellowship.

## 7. REFERENCES

- [1] M. Bhardwaj, R. Min and A. Chandrakasan, "Quantifying and Enhancing Power-Awareness of VLSI Systems", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, issue 6, Dec. 2001, pp. 757 -772.
- [2] H. Sorensen, D. Jones, M. Heideman, C. Burrus, "Real-Valued Fast Fourier Transform Algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, No. 6, June 1987.
- [3] J. Goodman, A. Chandrakasan, "An energy-efficient reconfigurable public-key cryptography processor," *IEEE Journal of Solid-State Circuits*, Vol. 36, Issue 11, Nov. 2001, pp. 1808-1820.
- [4] M. C. Pease, "Organization of large scale Fourier processors", *JACM*, vol. 16, pp. 474-482, July 1969.
- [5] Artisan Components, *TSMC 0.18 $\mu$ m Process 1.8 V SAGE-X Standard Cell Library*, February 2002.
- [6] Artisan Components, *TSMC 0.18 $\mu$ m Process High-Speed Dual Port Register File (HS-RF-2P) Generator User Manual*, July 2000.
- [7] Artisan Components, *TSMC 0.18 $\mu$ m Process High-Speed Synchronous Single-Port Diffusion ROM Generator (ROM-DIFF-HS) User Manual*, December 2001.
- [8] Wosnitza, M., M. Cavadini, M. Thaler, and G. Troster., "A High Precision 1024-point FFT Processor for 2D Convolution," *Digest of Technical Papers 1998 IEEE ISSCC*, volume 41, pp. 118-119, 424.
- [9] DoubleBW Systems BV, *Data Sheet PowerFFT PCI32-Card*, 2002.
- [10] Digital Signal Processing Architecture Inc., *DSP-24 Datasheet, Rference DSPA-DSP24DS*, 2002.
- [11] B. Baas, "An energy-efficient single-chip FFT processor," *IEEE Journal of Solid-State Circuits*, Vol 34, Issue 11, March 1999, pp. 380 -387.