AN ENERGY-EFFICIENT RECONFIGURABLE FFT/IFFT PROCESSOR BASED ON A MULTI-PROCESSOR RING

Guichang Zhong, Fan Xu, Alan N. Willson, Jr.

Electrical Engineering Department
University of California, Los Angeles, CA 90095 USA
phone: +1 310 2062573, fax: +1 310 2064685, email: {zhong, fxu, willson}@icsl.ucla.edu

ABSTRACT

We have designed and built a single-chip reconfigurable FFT/IFFT processor that employs a ring-structured multiprocessor architecture. Multi-level reconfigurability is realized by dynamically allocating computation resources required by specific applications. The processor IC has been fabricated in TSMC 0.25-μm CMOS. It performs 8-point to 4096-point FFT/IFFT with power-scalable features and provides useful trade-offs between algorithm flexibility, implementation complexity and energy efficiency.

1. INTRODUCTION

In the domain of signal processing for communications, the current trends toward rapidly evolving standards and formats, and toward algorithms adaptive to dynamic factors in the environment, require programmable solutions that possess both algorithm flexibility and low implementation complexity. In addition, low power dissipation is critical in meeting portability requirements. Whereas software and hardware programmable designs using microprocessors and FPGAs provide implementation flexibility at the expense of higher power dissipation, compared to ASIC solutions [5], our goal is a better trade-off between algorithm flexibility, implementation complexity and energy efficiency.

Many recent communication standards, such as xDSL, wireless LAN or wireless Broadband access, propose OFDM or OFDMA as the primary modulation method. The Fast Fourier Transform (FFT) is essential for such modulation. The FFT operation has proven both computation-intensive and data-exchange-intensive, therefore the effectiveness of the FFT processor plays a key role in optimizing system performance.

We present here a single-chip reconfigurable FFT/IFFT solution based on a multiprocessor having a ring-structured topology, which performs 8-point to 4096-point FFT/IFFT with power-scalable features.

2. RECONFIGURABLE ARCHITECTURE

Higher radix FFT algorithms are known to require fewer operations than the classical radix-2 approach [1], and it is more efficient to implement higher radix FFT algorithms using multiprocessor schemes. A large-size DFT operation can be decomposed into smaller FFTs ("atom-FFTs"), followed by angle rotations [7]. Such a decomposition applies to any size FFT. In fact, for a specific FFT size, this decomposition is not even unique, and the efficiency of an FFT processor based on such a decomposition is highly dependent on the choice of the atom-FFTs. Considering factors such as latency, controller complexity and implementation efficiency, we have chosen 4-, 8- and 16-point FFTs as the primary atoms. These atom-FFT computations can be accomplished efficiently on a ring having four processors [7].

This work was supported by UC MICRO Grants 02-085 and 03-094, Analog Devices, Broadcom, Maxim Integrated Products, and Rockwell Scientific. We thank TSMC for fabricating the chip.

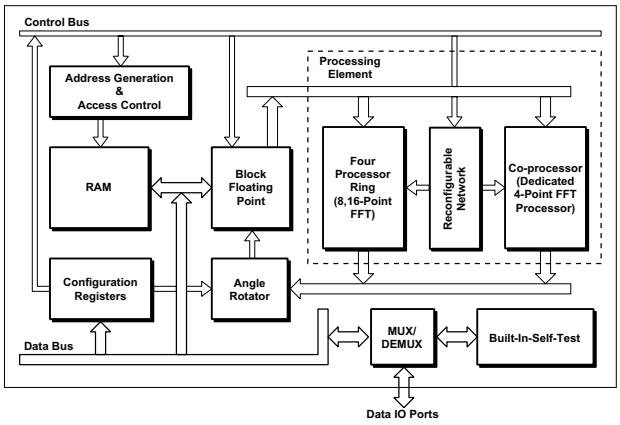

Figure 1: Single-chip FFT processor structure.

Based on such FFT decomposition algorithms, the IC architecture employed here consists of two computation components: a small-size FFT processor, mainly comprising a multiprocessor ring, and an angle rotator that performs phase-factor multiplications. As shown in Figure 1, our FFT processor integrates on a single chip: a programmable four-processor ring, a dedicated four-point FFT processor, an angle rotator, a memory interface and memory, address generators, a block-floating-point (BFP) unit, and built-in-self-test (BIST). The processor can be reconfigured, consistent with application and data-flow needs. The reconfigurability occurs at two levels, the first being a flexible selection of small-size FFT computations and angle rotators, and, at a lower level, by datapath reconfiguration inside each component. Furthermore, since the processor ring is programmable, it is less dependent on hardware to realize reconfiguration.

2.1 Atom-FFT Processors

A ring-structured multiprocessor [6] attempts to find a balance, for a set of signal processing tasks, between ASIC implementations and the use of general-purpose digital signal processors (DSPs). Unlike a DSP, the processor ring does not attempt to accommodate "all signal processing applications" but rather, realizes only certain types of algorithms, such as digital filtering and small-size FFTs. The ring-processor, however, in contrast to a typical full-custom IC, is not constrained to one specific filter and it retains much of the reconfigurability and progammability of a DSP.

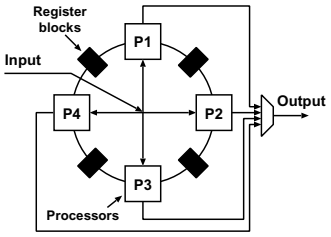

Figure 2: The processor ring.

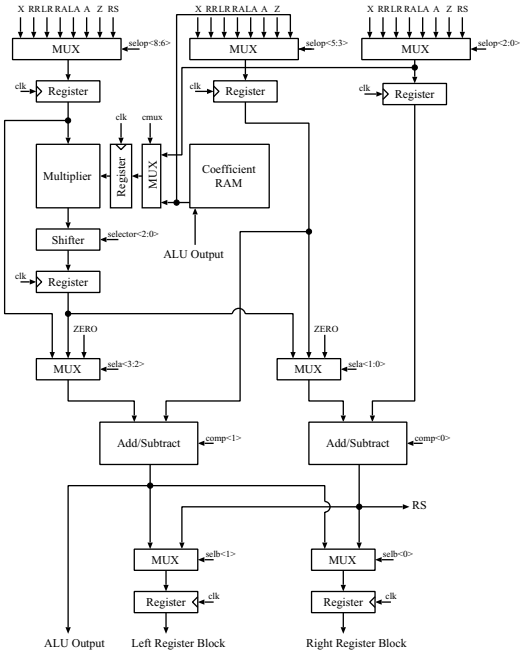

Figure 3: ALU of a single processor of the four-processor ring.

On a single chip, four processors are arranged in a ring topology with a dual-ported register block placed between adjacent pairs of processors, as shown in Figure 2. The processor ring employs parallel (reduced-instruction-set) processors executing very-long instruction words (VLIW). A typical instruction for any of the four processors specifies up to four separate operations: one multiply, one data-move, and two add/subtracts. Each processor's ALU, as shown in Figure 3, is pipelined and parallelized so that multiple instructions can execute simultaneously. Thus, the single-chip four-processor ring can execute 12 simultaneous operations per cycle [4 processors $\times$ (1 multiply + 2 add/subtract, simultaneously, per cycle, per processor)].

The basic FFT "butterfly" operation consists of four real multiplications and eight real additions and it can be accomplished in a single clock cycle on the four-processor ring (which has four multipliers and eight adders). Our FFT implementations achieve this "one butterfly per clock cycle" performance in a manner in which inter-processor data exchanges are significantly reduced. In particular, 8- and 16-point FFTs can be efficiently programmed on the four-processor ring: 12 clock cycles are needed for the 8-point FFT and 31 clock cycles for the 16-point FFT.

This system's ring topology also elegantly avoids memory-access contentions and requires no bus arbitration hardware. No penalty, in comparison to bus-type architectures, has yet been observed when programming the multiple processor ring for a variety of DSP applications. In addition

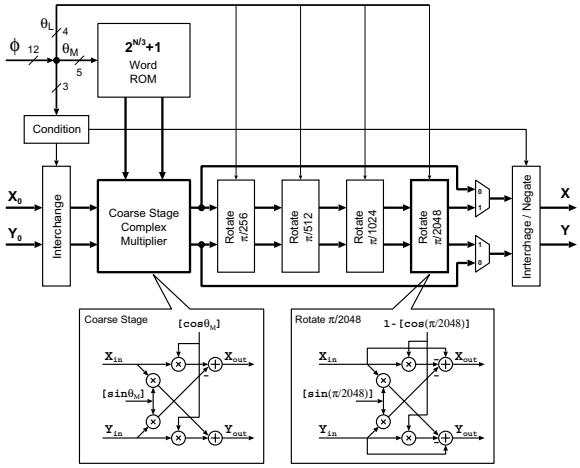

Figure 4: Reconfigurable angle-rotator architecture.

to high speed and throughput, the flexible ALU datapaths also endow the processor with great reconfigurability [7].

A dedicated (non-programmable) four-point FFT processor is also separately implemented to assist the processor ring when decompositions involve two different sized atoms. A simple radix-2 multi-path delay-commutator architecture is adopted for this special four-point FFT processor, which needs only adders and registers.

2.2 Angle Rotator

This component uses a ROM and dedicated arithmetic blocks to efficiently compute $(X + jY) = (X_0 + jY_0)e^{j\phi}$. That is, it rotates the complex data-point $X_0 + jY_0$ about the origin, through an input angle ϕ to produce the complex output data-point $X + jY$.

It is well known that ROM size can be reduced by a factor of eight by utilizing the quarter-wave symmetry of sine/cosine functions and trigonometric identities, i.e. the angle $\phi \in [0, 2\pi]$ can be mapped into an angle $\theta \in [0, \pi/4]$. The ROM size can be further reduced by decomposing the rotation into two stages: a coarse rotation followed by a fine rotation [2].

In an FFT processor, the angle control word is relatively short. Hence, so is the fine-stage control word [7]. Moreover, not all the bits of this short control word are always active. It therefore becomes more efficient to adopt an alternative to the two-stage preliminary architecture discussed in [4]. We propose a cascade structure of multiple stages for an FFT processor's angle rotation, as shown in Figure 4. This architecture consists of a coarse stage followed by four fixed angle rotation blocks (sub-rotators) which perform the rotations of $\pi/256$, $\pi/512$, $\pi/1024$, and $\pi/2048$ radians. Compared to the two-stage angle rotators, it may seem that the cascade architecture would occupy larger area and consume more power. Our efficient implementation, on the contrary, achieves smaller area and much less power consumption. The effectiveness is based on the fact that, for each fine stage, since the sine/cosine values are fixed, no approximation of these values, such as is adopted in [2] and [4], is needed. Furthermore, since the values of $\sin\theta$ and $1 - \cos\theta$ are small, by using CSA (carry-save addition) and sub-expression sharing design techniques, no general-purpose multipliers are necessary and IC area is significantly reduced.

This angle rotator distinguishes itself from angle rotators in [4] by its flexibility and reconfigurability: the angle rotator

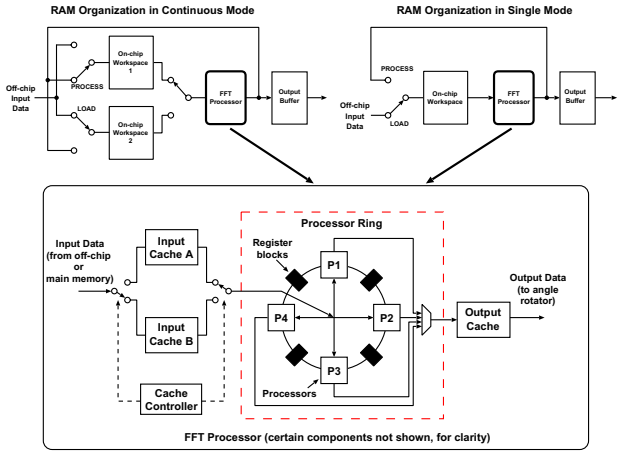

Figure 5: Memory hierarchy of the FFT processor.

can be configured by selecting outputs from the coarse stage or a sub-rotator, and simultaneously shutting down unneeded sub-rotators. Sub-rotators are only activated when the "resolution" of the angle is greater than $\pi/256$; and, moreover, a sub-rotator "stands by" if its control bit is zero, thereby reducing power consumption.

This architecture can be simply configured for IFFT operations, which use negative angle rotations. The configuration simply modifies existing FFT mapping operations; thus no additional logic circuits are required.

2.3 Memory Hierarchy

As shown in Figure 5, the FFT processor memories are organized in a four-level hierarchy: main memory storing input and output data; caches between main memory and the processor ring; register files between individual processors within the ring; and register files within each processor (containing program and coefficient data). Compared to caches in a general memory hierarchy, caches in the FFT processor not only reduce the memory access latency and processor-memory traffic, but also benefit FFT operations; the input caches reorder the input data so that they can be used efficiently by the processor ring, and the output cache also performs bit-reversal reordering for atom-FFT operations, thereby avoiding costly higher-level bit-reversal computations. On-chip dual-port SRAM is provided for main memory on our prototype chip to avoid an off-chip memory-transfer bottleneck.

The memory system can be reconfigured depending on the operation modes: continuous mode or single mode. In continuous mode, the RAM organization allows, simultaneously, the input of new data and the transforming of data already stored in RAM, as well as the outputting of previous results. In single mode, the IC is configured with separate load, transform and output operations. The memory interface controls the reconfiguration, data input/output, and data exchanges between main memory and processors.

2.4 Address Generation Unit

There are three address generators in the FFT processor: an input address and an output address generator providing the RAMs with correct addresses during the data input and output process, and an angle-rotator address generator providing the angle rotator with the corresponding control word. Unlike well-known radix-2 FFT processors, the address generators of our FFT processor do not perform the simple bit-reverse operations. Instead, they perform a switching operation in blocks of two, three or four bits, depending on the size of the atom-FFTs, and the block-switching pattern differs in different stages of the computation. A multiplier-free auto-reconfigured structure for the address generator is employed.

2.5 Block-Floating-Point

BFP operations have proven to significantly improve the numerical performance of an FFT processor [3]. In our processor, only one BFP block is used and the BFP shifter is located at the input port of the FFT processor, i.e., the data are not shifted until they are read into the processor for the next stage of computations ("input scaling").

2.6 BIST

An on-chip BIST circuit has been incorporated into the system to facilitate post-fabrication functional testing. It consists of two components—the first component is a circuit that generates a pseudo-random sequence to be used as chip inputs. The processor ring, with the help of accessory circuits, can then select this PN sequence as its input in the test mode, and its output is sent to the second component of the BIST circuit, which analyzes the output and detects if any error has occurred. This component produces a signature from each output of the processor ring. After the entire PN-sequence has been input, the final signature is held by internal registers. It can then be sent out and compared with the result from prior simulation to verify the chip's functionality.

3. PROTOTYPE IMPLEMENTATION

The FFT processor was implemented using a standard-cell-based design methodology based on the 0.25-μm Artisan Components library plus full-custom cache and register files. The chip was fabricated in the TSMC 0.25-μm 5-level-metal CMOS process. Figure 6 shows the chip microphotograph, and Table 1 summarizes its overall characteristics.

3.1 High-Level Design

Matlab was used for algorithm-level simulation and verification. Logic design was done using Verilog and a Verilog-XL simulator. The processor-ring design was based on gate-level structural descriptions, while the design of the angle rotator, BFP and other control logic was based on behavioral descriptions, and synthesized to get the gate-level netlist.

3.2 Full-Custom Cell Design

The processor system's cache memories and register files must be both high-speed and low-power, since its performance is primarily affected by these memory elements. We therefore used fully custom designed and optimized caches and register files. The program and coefficient memories are made up of four 40-word by 54-bit register files and four 32-word by 16-bit single-port register files, respectively. The single-port register file core cell contains six transistors. Cache memories and inter-processor register files comprise seven 32-word by 24-bit dual-port register blocks using eight-transistor cells.

3.3 Floorplan

The prototype chip's floorplan is shown in Figure 6, where the logic part of the processor occupies approximately 1/4 of the core area and contains approximately 0.52 million transistors, while the SRAM blocks holding complex data

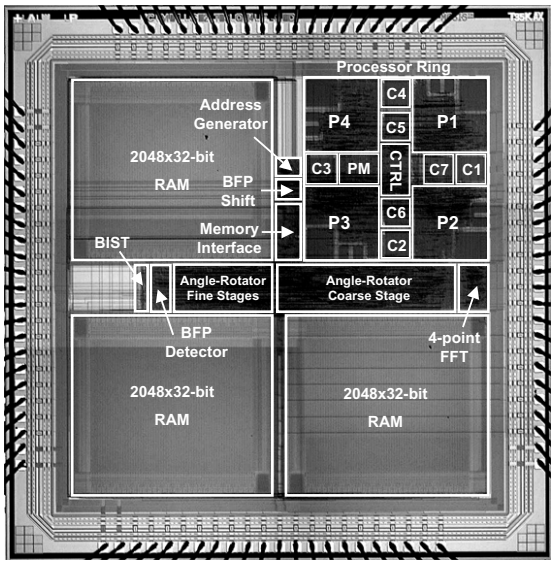

Figure 6: Chip microphotograph.

Table 1: Chip characteristics.

Technology	TSMC 0.25-μm 5-level-metal CMOS
Instruction format	VLIW
Die size	$4.28mm \times 4.28mm$
Core size	$3.38mm \times 3.38mm$
Transistor count	2,265,732
Port buswidth	16-bit
Internal buswidth	24-bit
Program memory size	4×(40-word×54-bit)
Coefficient memory size	4×(32-word×16-bit)
Cache memory size	7×(32-word×24-bit)
Main memory size	3×(2048-word×32-bit)
Maximum clock rate	200 MHz (@2.5 V)

occupy 3/4 of the core area. When using the FFT/IFFT processor as a component in a system-on-chip application, this memory can likely be provided by other on-chip shared memory. In such applications, the area consumed by the FFT processor would more correctly be assessed as approximately 1/4 the Figure 6 area, i.e., $1.7mm \times 1.7mm$.

For such a large-scale ASIC, proper floorplanning for the clock distribution network is a key factor in achieving high frequency and low power dissipation. Our manually designed clock-tree has been carefully tuned for low clock-skew. For this multi-processor chip, another concern is to provide equal path-lengths to each processor for input and output, and to keep the paths between a processor and its two adjacent register blocks as short as possible. The four-processor ring, sitting in the IC's upper-right corner, has its input and output blocks and controller centrally located, and processors and register blocks are placed symmetrically between them. Furthermore, the ALU, program memory and coefficient memory in each individual processor are placed such as to minimize the path delay between them.

4. RESULTS AND CONCLUSION

IC functionality, speed and power dissipation have been tested using an HP82000 Tester and on-chip BIST features. The HP82000 IC tester verified the logic functions of the FFT processor chip and the on-chip BIST measured the maximum operation frequency and power consumption.

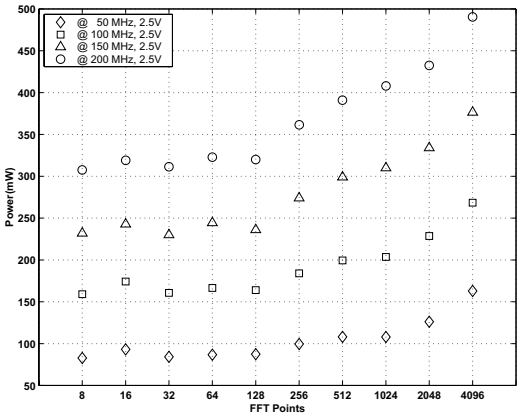

Figure 7: Power consumption vs. FFT size.

The measured power dissipation, when programmed for various FFT sizes, is presented in Figure 7, demonstrating a significantly scalable power-dissipation feature with varying FFT size. Given this feature, our FFT processor can well outperform programmable FFT processors whose power consumption does not scale with FFT size.

The processor computes a 1024-point FFT in 5,280 clock cycles, taking approximately 26.4 ms at the maximum clock speed of 200 MHz, with a supply voltage V_{dd}=2.5 V. Compared to industry benchmarks of programmable fixed-point DSPs, in terms of cycles, it is 2.46 [2.46 = 12972/5280] times faster than the Texas Instruments TMS320C62x, and 1.14 [1.14 = 6002/5280] times faster than the TMS320C64x. With a real sinewave input for a 1024-point FFT, the signal-to-quantization-noise ratio is measured as 61.23 dB, and the Spurious-Free Dynamic Range is measured as 72.67 dB.

Compared to state-of-the-art programmable processors, the trade-off between algorithm flexibility, implementation complexity and energy efficiency achieved by this multiprocessor-ring based FFT processor makes it an ideal candidate for systems requiring multi-standard and multi-algorithm capability.

REFERENCES

[1] C. S. Burrus, T. W. Parks, and J. F. Potts, *DFT/FFT and Convolution Algorithms*. New York: Wiley, 1985.

[2] D. Fu, *Efficient Synchronization Architectures for Multimedia Communications*. Ph.D. thesis, UCLA, 2000.

[3] W. W. Smith and J. M. Smith, *Handbook of Real-Time Fast Fourier Transforms*. Piscataway, NJ: IEEE Press, 1995.

[4] A. Torosyan, D. Fu, and A. N. Willson, Jr., "A 300 MHz quadrature direct digital synthesizer/mixer in 0.25-μm CMOS," *IEEE Journal of Solid-State Circuits*, vol 38, pp. 875–887, June 2003.

[5] N. Tredennick and B. Shimamoto, "Go reconfigure," *IEEE Spectrum*, vol 40, pp. 36–40, December 2003.

[6] M. J. Werter and A. N. Willson, Jr., "Automated programming of digital filters for parallel processing implementation," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol 41, pp. 285–294, April 1994.

[7] F. Xu, *On Designing High-Performance Signal Processing Algorithms for a Ring-Structured Multiprocessor*. Ph.D. thesis, UCLA, 2001.