Advanced Hardware and Software Technologies for Ultra-long FFT's

Hahn Kim, Jeremy Kepner, M. Michael Vai, Crystal Kahn
{hgk, kepner, mvai, kahn}@ll.mit.edu
MIT Lincoln Laboratory, 244 Wood St., Lexington, MA 02420-9108

Abstract

The desire to digitally perform as many processing steps as practical in sensor applications continues to push the point of digitization toward the RF front-end. Modern sensor applications thus often have to collect and process large amounts (Terabytes) of data at high sampling rates (Gigasamples per second). Developing real-time signal processing technologies (e.g., real-time Gigapoint FFT's) for these applications has become increasingly more difficult. In this paper, we will discuss advanced hardware and software technologies for ultra-long FFT operations.

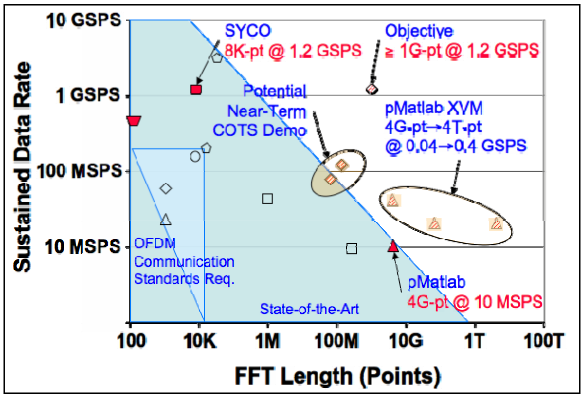

Figure 1 – Survey of FFT technologies

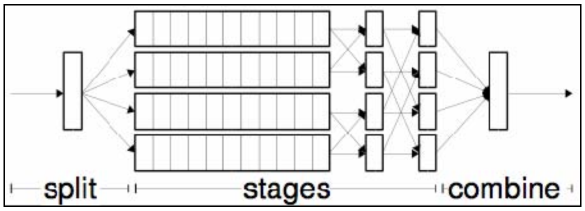

Figure 2 – Parallel pipeline FFT architecture

1 Introduction

Modern sensors are capable of collecting data with significantly large sizes at very high rates. For example, as digital receivers move well into the GHz sampling regime, it has become common to collect Terabytes of data. Developing signal processing technologies that can process these large data sizes in real-time is becoming increasingly difficult.

The MIT Lincoln Laboratory has been at the forefront of developing advanced FFT technologies. A survey of state-of-the-art FFT technologies is shown in Figure 1. COTS FFT technology, shown in the lower-left corner, is driven by communication applications. The near-term objective is to develop systems that can process 1 Gigapoint FFT's at 1.2 Gigasamples per second (GSPS). Clearly, current COTS technologies cannot meet these criteria.

Real-time implementations of FFT's require dedicated hardware, e.g., ASIC's or FPGA's. Lincoln has developed embedded hardware technologies that can process FFT's at high data rates in real-time; a systolic FFT architecture implemented with FPGA's can process an 8K-point FFT at 1.2 GSPS [1].

Parallel processing technologies, such as the Parallel MATLAB library (pMatlab), allow offline processing of data [2]. An FFT implemented with pMatlab running on 128 CPU's can process a 4 G-point FFT at 10 MSPS [3].

We have developed an extension to pMatlab known as eXtreme Virtual Memory (XVM), which enables pMatlab applications to process data sets orders of magnitude larger by using out-of-core methods. Out-of-core methods use memory as a "window" to view a section of the data stored on disk at a time. Software technologies such as pMatlab XVM provide a means for developing and validating algorithms for ultra-long FFT's before implementing them in hardware.

Sections 2 and 3 will discuss Lincoln hardware and software technologies for ultra-long FFT's, respectively. Section 4 will summarize strategies for combining these technologies to develop real-time systems for ultra-long FFT's.

2 Hardware Technology

Lincoln has developed a systolic architecture for real-time FFT's, optimized in modularity and local communication for ASIC and FPGA implementations [1]. Figure 2 depicts this architecture as multiple parallel pipelines of serial FFT's.

The inter-pipeline interconnections in the last stages are determined by the number of parallel pipelines, independent of the FFT size. In theory, this architecture can be extended to any size input. For example, a 1 G-point FFT would require 30 stages of butterfly modules. Large-scale implementations, however, present several challenges. For example, external memory would be required to store the twiddle factors and to implement FIFO buffers between butterfly stages. A 1 G-point FFT would require 1 Gigawords of memory for the twiddle factors and 1 Gigawords of memory

This work is sponsored by the Department of the Air Force under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government

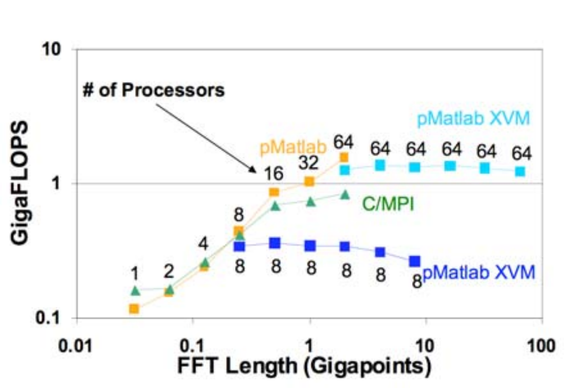

Figure 3 – Parallel FFT Software Performance

for the buffers. If one word is 4 bytes, this equals 8 GB of memory.

In theory, an ASIC can be created with the required memory but, in practice, this is very difficult. Integrating external memory is feasible using state-of-the-art memory technology, but the access data rate makes implementing the interface between memory and the FFT chip a challenge.

No FPGA has the required amount of memory. So far, the largest FFT implemented in an FPGA is about 32 Kilopoints. Long FFT's can be built by cascading smaller FFT's connected by cornerturns. A 1 G-point FFT, for example, can be formed with two 32 K-point FFT's. Lincoln has developed a cornerturn architecture that can process a 1 G-point FFT at 1 GSPS.

3 Software Technology

pMatlab is widely used at Lincoln for implementing computationally intensive applications. pMatlab XVM extends the parallel processing capabilities of pMatlab with out-of-core methods. pMatlab XVM uses hierarchical global arrays to structure and swap data between memory and disk storage in a manner that is optimal for a particular algorithm, hiding the large amount of index bookkeeping required to implement out-of-core algorithms. Hence, applications can use all available disk storage as memory to accommodate extremely large data sets while leveraging the power of multiple processors, with little sacrifice in performance.

DARPA's High Productivity Computing Systems (HPCS) program [4] has created the HPCchallenge benchmark suite [5] in an effort to redefine how productivity is measured in the HPC domain. We have applied pMatlab XVM to the HPCchallenge FFT benchmark. The benchmark was implemented using MATLAB, pMatlab, C+MPI, and pMatlab XVM.

The results show that the pMatlab XVM version of the benchmark is easy to write and understand and incurs a relatively small performance overhead while performing a 64 Gigapoint FFT. Figure 3 compares the performance of the various versions of the FFT benchmark. These results show that pMatlab XVM:

- Provides 80% of pMatlab's performance for a specific size and number of CPUs
- Is able to increase problem size with minor degradation in performance, and
- Allows all available local disk storage to be used, enabling a 1 TB (64 Gigapoint) FFT.

4 Summary

Modern sensors are capable of collecting data at high rates. The increased data rates and sizes present challenges to the capabilities of existing hardware technologies. Lincoln has been developing innovative hardware technologies to meet the requirements of real-time FFT's.

Software technologies such as pMatlab XVM enable offline data processing of large data sets that are beyond the capabilities of current hardware systems. Lincoln has developed an FFT implemented with pMatlab XVM for a DoD application that requires a 1 G-point FFT.

The inherent flexibility of these software technologies also enables proof-of-concept demonstrations for large-scale FFT algorithms. For example, pMatlab XVM will allow analysts and systems designers to experiment with various FFT design parameters (e.g. fixed-point vs. floating-point, dynamic range, etc.) before implementing the design in hardware to perform in real-time. pMatlab XVM also enables analysis of the overall system, allowing analysts to also determine how the FFT operation affects the overall signal processing chain.

References

[1] P. A. Jackson, C. P. Chan, J. E. Scalera, C. M. Rader, and M. M. Vai, "A Systolic FFT Architecture for Real Time FPGA Systems," *High Performance Embedded Computing (HPEC) Workshop 2004*, September 2004.

[2] J. Kepner, N. Travinin, "Parallel MATLAB: The Next Generation," *High Performance Embedded Computing (HPEC) Workshop 2003*, September 2003.

[3] R. Haney, A. Funk, J. Kepner, H. Kim, C. Rader, A. Reuther, N. Travinin, "pMatlab Takes the HPCchallenge," *High Performance Embedded Computing (HPEC) Workshop 2004*, September 2004.

[4] HPCS – High Productivity Computer Systems, http://www.highproductivity.org.

[5] HPCchallenge, http://icl.cs.utk.edu/hpcc.