Accelerated Signal Processing in XMidas Using the Cell Broadband Engine

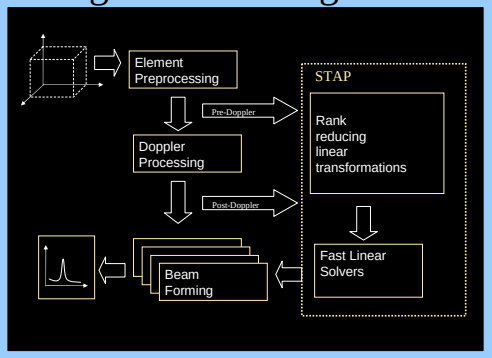
Joseph A. Jezak, Donald M. Chiarulli, Steven P. Levitan – University of Pittsburgh, Charles Berdanier – AFRL Wright Patterson AFB

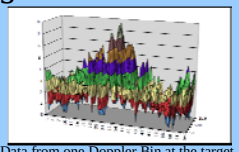
Radar Signal Processing in XMidas

XMidas Software Platform:
- Software toolkit for DSP
- Similar in function to Matlab
- Widely used by the United States Defense Department

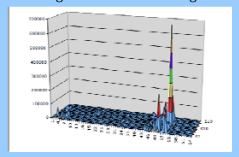
Example Radar Application:
1. Create Doppler bins
2. Use Space-Time Adaptive Processing (STAP) algorithm to optimize beam forming
3. Determine the location of targets

Project Goals:
- Port XMidas to the Cell Broadband Engine
- Accelerate common DSP functions using both the PPU and SPUs





Data from one Doppler Bin at the target range before beam forming



The same data after beam forming using weights from STAP

Heterogeneous Computing Platform

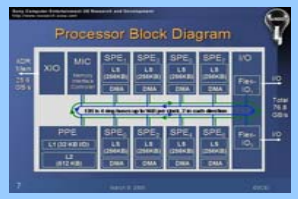
Goal:
- Provide a platform for high performance radar processing
- Allow programs written for XMidas to be run real-time in the field

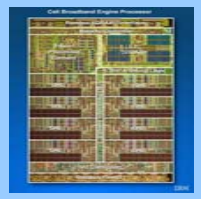
Approach - Heterogeneous architecture:
- FPGA DSP data acquisition system (ICE-PIC 5)
- Dual, quad core Xeon host
- Cell Broadband Engine
- Nvidia Tesla GPU

Heterogeneous platform provides maximum performance and flexibility for Radar DSP applications

.

Cell Broadband Engine Architecture

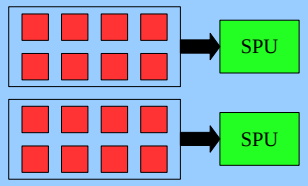




The Cell Broadband Engine is a unique multiprocessor, combining a traditional processing core with up to 8 synergistic processors operating at 3.2GHz. The Synergistic Processing Units (SPUs) are in-order vector processors with 256Kb of local storage. These cores are connected by a ring bus which can transfer data at maximum rate of 204.8 Gb/s when fully utilized.

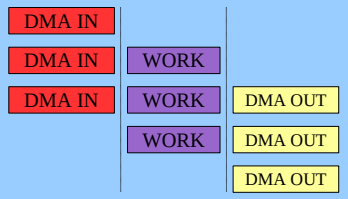
Parallel Programming Paradigms

Block Parallel



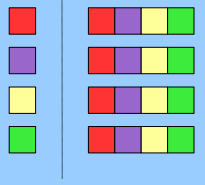
The CBE offers 8 SPUs with which to divide work as with traditional multiprocessors.

Pipelined Parallel



Data Transfers to the CBE are overlapped with processing using double or triple buffering.

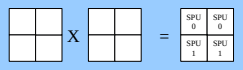
Vector Parallel



Data is grouped so that it can be processed in parallel, improving throughput.

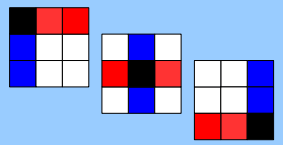
Radar/DSP Support Algorithms

Matrix Multiplication



Blocked Matrix Multiply
- Matrices are partitioned
- Blocks are dispatched to SPUs
- SPUs multiply blocks
- Results are combined

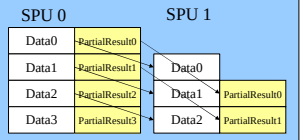
Matrix Inversion



Blocked Matrix Inversion[1]
- Matrix is partitioned
- Inversion performed on black block
- Red, White and Blue blocks are dispatched to SPUs
- Update partial rows and columns

[1]Yokoyama, S.; Matsumoto, K.; Sedukhin, S.G.; , "Matrix Inversion on the Cell/B.E. Processor," *High Performance Computing and Communications, 2009. HPCC '09. 11th IEEE International Conference* on , vol., no., pp.148-153, 25-27 June 2009

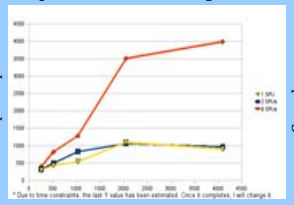
Convolution / FIR



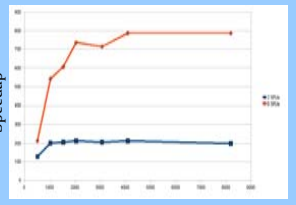
Pipelined FIR
- Taps are partitioned over SPUs
- Data passed to next SPU immediately
- Results are passed to the next SPU at half speed

Performance Results

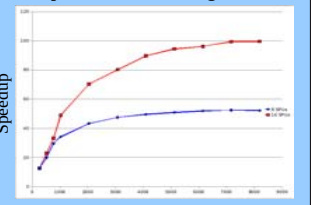
Matrix Multiplication
Up to 3500x Faster using 8 SPUs



Matrix Inversion
Up to 800x Faster using 8 SPUs



Convolution / FIR
Up to 100x Faster using 8 SPUs



All speedup results are SPU accelerated implementations compared to naive PPU implementations
- Achieved these speeds by utilizing partitioning, pipelining and vectorization of data on SPUs
- Optimized DMA transfer block sizes by balancing and overlapping transfer time with computation

Conclusions:
- These results show the advantages of multiple parallel programming paradigms in a heterogeneous computing environment
- Further work will investigate further integration of heterogeneous components for use within radar applications