



MR IMAGE RECONSTRUCTION ON THE CELL PROCESSOR

J. Czechowski, B. Moore, S. Zingelewicz, L. Marinelli, C. Hardy
GE Global Research Center, Niskayuna, NY, United States
 M. El-Demerdash
GE Healthcare, Waukesha, WI, United States

INTRODUCTION

Computer chip-making capabilities have advanced to the point where multiple, heterogeneous compute engines can be placed onto a single chip. We examine the applicability of one type of these chips, the Sony/Toshiba/IBM Cell multi-processor¹, to MRI. The Cell multi-processor^{2,3} is a highly parallel, single precision floating point compute engine with a very high-speed on-chip data interconnect. Because of these features, we believe it is well suited to the sort of computational loads characteristic in several types of MR reconstruction algorithms. We describe the acceleration of three of these algorithms using the Cell over conventional computer processors such as the AMD Opteron.

As shown in Figure 1, each Cell has nine processors: one general-purpose PowerPC core and eight Synergistic Processing Elements (SPEs). The SPEs are 128bit SIMD engines, which can process four 32-bit data elements in a single instruction. Each SPE has limited memory (256KB) and no cache, so a double buffering scheme was used. The Cell architecture allows DMA (direct Memory Access) operations to occur in parallel with computations, and its high-speed bus (EIB) is sufficiently fast (for this size dataset) that the DMA operations are completely covered by the IFFT computations. Figure 2 illustrates the general methodology and partitioning used for these algorithms.

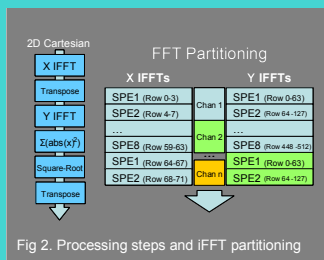


Fig 2. Processing steps and IFFT partitioning

For a full k-space recon – the simplest case –, the processing is performed in 2 stages where each stage involves one complete pass through the data. The SPEs are synchronized after each stage.

During the first stage, each SPE DMA's in one of its stripes (4 rows) of data at a time, performs an IFFT on the stripe, and completes the transpose as the stripe of data is DMA'ed out. Each SPE continues doing this until it has processed all of its stage 1 data. At the end of stage one, all of the x-axis IFFTs have been performed, and the data has been transposed and sent back to main memory.

During the second stage, each SPE DMA's in the one of its stripes (4 rows) of data at a time, performs an IFFT on the stripe, and accumulates the sum of the squares of the absolute value of the results. Each SPE continues doing this until it has processed all of the analogous stripes for each of the coils. The SPE then computes the square-root of the accumulated results and again completes the transpose as the stripe of resultant image is DMA'ed back to main memory. Each SPE continues doing this until it has processed all of its stage 2 portion of the dataset. At the end of stage two, the resultant images are in main memory.

The Homodyne is similarly performed in 2 stages -- it just has twice as many SPE buffers (high pass and low pass). For ASSET, it takes 2.5 passes through the data to compute the sensitivity map, but only one pass to apply it.

RESULTS & DISCUSSION

The timings in the right-most column of Table 1 were attained using 16 SPEs (both Cell chips) on an IBM Cell blade (version QS20) with SDK 2.0. In general, performance gain increases both with the complexity of the particular computations (See Table 1.) and with the number of SPEs used. See Figure 3. (Note: The effects of Amdahl's law⁶ become more apparent when the second Cell processor – with 8 more SPEs, totalling to 16 SPEs -- is introduced.)

	Opteron	Cell Blade
Full k-space	0.09	0.05
Homodyne	0.50	0.13
ASSET	2.74	* 0.17

Table 1: Performance Comparison (processing four - 512x512 slices from 8 coils)
 * - includes time to compute sensitivity map

Exploiting even a fraction of the theoretical potential of a multi-processor system is non-trivial. As multiprocessor systems proliferate, software tools will evolve to make the programming task more manageable. In this particular case, the Cell implementation of these MR reconstruction algorithms was written in C using IBM provided libraries and compilers. One program was written to run on the Cell's PowerPC, and a single program was written to run on each of the SPEs. The latter was possible because of the symmetry of the processing needed for the MR recon algorithms. Each SPE simply focuses on processing a different stripe of the data during each phase of the computations.

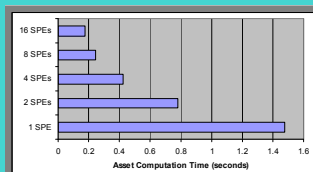


Fig 3. Number of SPEs vs Asset computation time

Sample results are shown in Figure 4.



Figure 4: Sample 512x512 output from Homodyne

CONCLUSION

Additional performance gains are possible and expected. In particular, the code generated in this study can be further optimized, and more Cell chips could be used, in parallel. Also, further Cell chip performance increases are anticipated. In conclusion, we can expect MRI system performance to benefit from multi-processor computer chips. This is likely to be needed as the number of coils expands, the size of images increases, and the usage of 3D datasets continues to grow.

REFERENCES

1. S. Williams, et al., "The Potential of the Cell Processor for Scientific Computing", *ACM International Conference on Computing Frontiers*, May 2006.
2. J. A. Kahle, et al., "Introduction to the Cell Multiprocessor", *IBM Journal of R&D*, issue 49-4/5, July 2005.
3. M. Gschwind, et al., "Synergistic Processing in Cell's Multicore Architecture", *IEEE Micro*, March 2006.
4. D. C. Noll, et al., "A Homogeneity Correction Method for Magnetic Resonance Imaging with Time-Varying Gradients", *IEEE Trans. on Medical Imaging*, October 1991.
5. K. King, et al., "Spiral Navigated Sensitivity Calibration for Parallel Imaging", *Proc 12th ISMRM*, 2004.
6. "Amdahl's law", *Wikipedia*, 2007.

ACKNOWLEDGEMENTS

We wish to thank Jason Polzin (MR chief engineer, GE-Healthcare) for his support of this effort. We are also appreciative of the Cell processor technical support that we received from IBM, especially from Gordon Fossom (STI Design Center).

MATERIALS & METHODS

Several relevant MR reconstruction techniques were implemented on a Cell. In particular, this included a full k-space recon using a 2D IFFT, a Homodyne⁴ recon, and an ASSET⁵ recon. Each algorithm was implemented and compared against reference implementations on conventional PC processors.

Our test dataset consisted of four 512x512 images (slices) from 8 coils. While the input data consisted of 16-bit integers, all computations were performed using single precision floating point operations.

The algorithms were implemented and tested on an IBM Cell prototype blade with two 3.2GHz Cell multi-processor chips, each supplemented with 512MB of main memory.

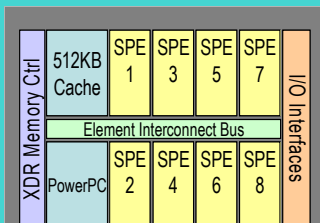


Fig 1. Block architecture diagram of the Cell Multi-processor