THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 14, Number 3/2013, pp. 255–258

FIXED-POINT REAL TIME MRI RECONSTRUCTION USING CONNEX ARRAY

Andrei ȚUGUI

"Politehnica" University of Bucharest, 1-3 Iuliu Maniu, 061071, Bucharest, Romania
E-mail: andrei.tugui@yahoo.com

This paper approaches the fixed-point computation involved in magnetic resonance image (MRI) reconstruction with most common technique, the Fast Fourier Transform (FFT) on a parallel-computing machine, Connex Array. We will show that fixed-point computation is a good alternative in real-time magnetic resonance imaging reconstruction as it can seriously improve reconstruction time (about 75% better than using floating-point operations) when large amount of data must be computed and image quality is not critical (dynamic angiography, cardiac magnetic resonance imaging).

Key words: connex array, fixed-point representation, magnetic resonance imaging (MRI) reconstruction, parallel computing, two-dimensional (2D) fast Fourier Transform (FFT).

1. INTRODUCTION

In real-time Magnetic Resonance Imaging (MRI) and dynamic Magnetic Resonance Imaging (dMRI), reconstruction speed is crucial as it must be achieved multiple images from a single examination [1]. Other applications require several images to be acquired and several K-space filling techniques like parallel MRI (pMRI) are used to spread the samples in K-space (Fourier Transform space) to perform data analysis easier [2]. We will show in this paper an alternative fixed-point parallel image reconstruction to regular floating-point approach on a powerful parallel computing machine, Connex Array [3] to overcome this issue. MRI multicore reconstruction (parallel computing) is required nowadays because clinical scanners have multi-channel acquisition systems [4]; also, high computational issues reveal cardiac applications where movement created by breathing for example requires specific processing techniques trying to speed-up reconstruction when resolution and image contrast are satisfactory and images must be processed faster (triggered multislice reconstruction [5]). Typical dMRI data amount brought into memory is about 2000 images (128 × 128 pixels), meaning about 128 MB data to process [6].

Most common reconstruction technique in MRI is Fast Fourier Transform (FFT) [7]. Unlike spiral methods and gridding algorithm, FFT does not require floating-point operations for reconstruction as floating-point computation is slower [8]. In this paper, we make a comparison between fixed-point and floating-point reconstruction of a human brain transversal slice using Radix-2 FFT algorithm, showing that computational speed can increase by 75% using fixed-point data representation on a general purpose parallel-computing machine, Connex Array.

2. MRI COMPUTATIONAL ENHANCEMENT WITH CONNEX ARRAY

Connex Array is a parallel computing machine which best fits into *Terra Architecture* concept [3], a high performance architecture including several types of parallelism to optimize chip's area and power consumption (Fig. 2, Table 3). Real performances of this processing machine are 400 GOPS and 117 GFLOPS [3]; that means about 75% speed-up enhancement when computing fixed-point versus floating-point operations. Its parallelism is described in [3, 9, 10]. Paper [16] is a tutorial guide for Connex Array programming, which is basically done in a C++ language extension introducing a new data type, *vector,* modeling vector data into Connex (K-space samples). As workbench, we used Eclipse IDE for C/C++ Developers © [17] to reconstruct a 2D FFT 128 × 128 image from a transversal slice of a normal human brain by taking as input data (K-space) the samples provided by [11].

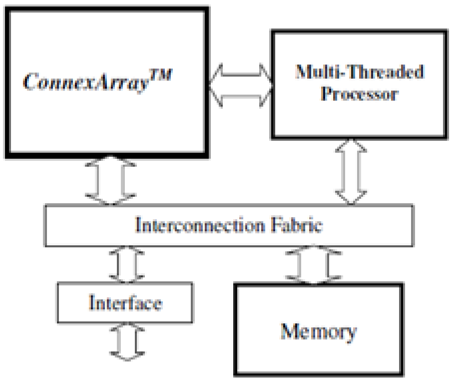

Fig. 1 – Connex Array.

It is known that when computing fixed-point operations, quantization and rounding errors may occur, but these errors can easily be eliminated by maintaining the optimal data precision. We took as dataset for our 2D FFT reconstruction some data samples [11] from a human brain transversal slice (Table 1) on 32 bits single precision. In 1D FFT reconstruction, for N = 8 (FFT dimension) bottleneck performance is reclaimed by the final stage of the Radix-2 algorithm [12], also by twiddle factors (sinus and cosines functions) and computation requiring operations as addition, multiplication, division. Connex uses only addition, multiplication and shift operations for the reconstruction process. All operations are complex, meaning we must compute FFT on both real and imaginary parts of K-space samples. If we consider only the multiplication operations ($2N^2$ for an image), computational time cost will decrease when using fixed-point computation about 50% as Connex makes one floating-point operation in 16 clock cycles but only 5 cycles for an integer or fixed-point operation.

Our data samples are represented as fixed-point real numbers with 1 digit entire part and 23 digits fractional part (Table 2). When representing fixed-point real numbers, the least N significant bits (N = 1) will memorize the integer part and the remaining bits the fractional one. One sample like – **0**.0000011 in our dataset is represented in binary as – **0**0000000000000000000000000001011 on 32 bits [13].

Table 1

K-space data of a real image of transversal slice from a normal human brain
for partial Fourier reconstruction, base 10 precision

Real K-space brain samples	Imaginary K-space brain samples
–0.00000090000000000000000	–0.00000110000000000000000
–0.00000230000000000000000	0.00000370000000000000000
–0.00000760000000000000000	–0.00000150000000000000000
0.00001140000000000000000	–0.00000180000000000000000
–0.00000010000000000000000	–0.00000160000000000000000
–0.00000150000000000000000	0.00000190000000000000000
–0.00000350000000000000000	–0.00000860000000000000000
0.00000440000000000000000	–0.00000210000000000000000

Although the exponent is missing, dynamic range for fixed-point samples is still acceptable because raw data samples in K-space are stored usually on 16 bits (0 to 65 535 values). Traditionally, FFT computation for MRI reconstruction is made with single precision (32 bits floating-point) with up to 7 digits fractional part or 64 bits (double precision) with up to 14 digits fractional part. These real samples are called *floating-point* because the point can be shifted allowing a bigger range for data representation. For MRI, input data (K-space raw data) is precision limited due to acquisition process; this way, by using a higher precision for a variable we can not achieve extra information [13], but only use much memory instead. After the FFT reconstruction process, all fixed-point output data remain in the same precision range (32 bits). After the IFFT of K-space samples, *xr[i]* represent the real part of the original image and *xi[i]* is the imaginary part of the image. Usually, the image is displayed using the magnitude, $I = \sqrt{xr[i]^2 + xi[i]^2}$.

Floating-point representation is preferred when it must be avoided quantization and rounding errors and a high image quality is required. But the higher precision approach means, on the other hand, higher computational time (e.g.: one standard MRI image, 64 × 64 × 32 voxels requires about 256 kB if it is stored as 16 bit integers, but only 1 024 kB when is stored as 64 bit floating-point [14]. The goal is to store data samples in fixed-point representation and to process them, and not as integers to avoid risk of information loss. Using a brain FFT image reconstruction we showed that information remains intact when we perform the reconstruction algorithm, 1D IFFT horizontally followed by another 1D IFFT vertically. We considered one 128 × 128 complex image, where each pixel has real and imaginary parts (Table 1). When we used IEEE Standard, Institute of Electrical and Electronic Engineers ANSI/IEEE Standard 754-1985: the IEEE Standard for Binary Floating-Point Arithmetic, Piscataway, NJ: IEEE, 1985, a K-space sample is represented on 52 bits for mantissa and 11 bits for the exponent, 350 frame/s frame rate was achieved for our 128 ×128 pixels image. Instead, using fixed-point computation Connex manages to reconstruct about 1390 frame/s, which is obviously a great speed-up enhancement. Connex performs both addition and multiplication identical for integers and fixed-point real numbers.

Table 2

Image data of the proposed transversal slice of a normal human brain for partial Fourier reconstruction, base 10 precision

Real and imaginary parts of the image (partial data)	
xr[i]	**xi[i]**
0.000000000000000000000000	0.000000000000000000000000
0.000000017677662000892269	–0.000000176776694615909950
– 0.000000325000030443334250	–0.000000399999976252729540
0.000000857366956097394000	–0.000001051821300279698300
– 0.000000075000080812515080	0.000000087500012568852981
– 0.000000247487349724906380	0.000000159099059260370270
– 0.000000050000000584304871	0.000000124999999684405340
–0.000001193242724184528900	–0.000000397747328406694580

Benchmark tests (Table 2) showed that results don’t exceed 32 bits precision imposed for accurate reconstruction. If we consider the quantization error (ε) related to different precision computation (fixed versus floating-point), this can be in the range [13]:

$$-2^{-(N+1)} < \varepsilon \le 2^{-(N+1)}, \quad (1)$$

where N is the number of digits holding the fractional part, but we did not took it into consideration from the beginning as this paper approaches only the speed-up enhancement for the real-time imaging where large data amount computation is the priority and the image quality is satisfactory for clinical research.

Table 3

FFT parallel implementation efficiency on different GPUs

Parallel Computing machine	Algorithm	Gflops	Data size (pixels)	Power consumed (W)	Power efficiency Gflops/watt
NVIDIA Quadro FX NV 40	2D FFT	486	128 × 128 32-bit, float	25	19.44
TMS320C4x DSP	2D FFT	147	128 × 128 32-bit, float	5	29.4
Connex Array	2D FFT	117	128 × 128 32-bit, float	**2.5**	46.8
IBM Cyclops-64	2D FFT	20	128 × 128 32-bit, float	83.22	0.24

Because of high speed image reconstruction, image quality loss from quantization and rounding errors are acceptable in MRI applications as cardiac MRI or dynamic angiography, but in fMRI applications these errors can affect seriously the image quality. It is known that 1 bit growth in precision involves 6 dB decrease of the quantization error.

The IFFT in a conventional image reconstruction process takes about 67% of the reconstruction's time, but using fixed-point reconstruction only 47%. With Connex, FFT algorithm is computed very fast [15] as it is a parallel computing machine. Additional information about Connex and FFT algorithm implementation on this machine can be found in [15] and [17].

3. RESULTS AND CONCLUSIONS

Table 4 shows a comparison between fixed-point and floating-point computation of one image on Connex, showing the effective speed-up in MRI and real-time MRI reconstruction when fixed-point computation is used instead of floating-point computation.

Table 4

Comparison between fixed-point and floating-point MRI 2D reconstruction using Connex Array

Function	Number of operations	Cycles used	Recon struction time (ms)	Frame rate (frame/s)
2D Image reconstruction 28 ×128 pixels 32 bits floating point	168668816	573440	0.67	350
2D Image reconstruction 128 × 128 pixels 32 bit fixed-point	168668816	143360	0.16	1395

REFERENCES

1. LIU K., ZHANG J., *Dynamic parallel image reconstruction based on adaptive coil sensitivity estimation*, 5th IEEE international symposium on biomedical imaging: From nano to macro, Paris, 14–17 May 2008.
2. CALHOUN V., ADDALI T., LIU J., *A feature-based approach to combine functional MRI, structural MRI and EEG brain imaging data*, Proceedings of the 28th IEEE EMBS Annual International Conference, New York City, USA, August 30–September 3, 2006.
3. ŞTEFAN Gh. M, *Integral Parallel Architecture In System-On-Chip Designs,* The 6th International Workshop on Unique Chips and Systems, Atlanta, GA, USA, December 2010, pp. 23–26. Available: http://www.arh.pub.ro/gstefan/2010ucas.pdf
4. CHANG C-H., JI J., *Compressed Sensing MRI with Multi-Channel Data Using Multi-core Processors*, 31st Annual international Conference of the IEEE EMBS, Minneapolis, Minnesota, USA, September 2–6, 2009.
5. NAYAK K. S., HU B. S., *Triggered Real-Time MRI and Cardiac Applications*, Magnetic Resonance in Medicine, **49**, *1*, pp. 188–192, 2003. Available: http://mrel.usc.edu/pdf/MRM_trig.pdf
6. COHEN M. S., *Real-Time Functional Magnetic Resonance Imaging,* Methods, **25**, *2*, pp. 201–220. Available: http://www.brainmapping.org/MarkCohen/Papers/Methods2002.pdf
7. BERNSTEIN M. A., KING K. F., ZHOU X. J., *Handbook of MRI Pulse Sequences*, Elsevier Academic Press, California USA, September 21, 2004.
8. HENNESSY J. L., PATTERSON D. A., *Computer Architecture: A Quantitative Approach,* 2nd Edition, Morgan Kaufman Publishers Inc., San Francisco, USA, 1996.
9. MALIŢA M., ŞTEFAN Gh. M., *Many-processors & KLEENE's model*, U.P.B. Scientific Bulletin – Series C, Vol. 72, 2010.
10. CALFA A. M., ŞTEFAN Gh., *Matrix Computation on Connex Parallel Architecture*, ICES 2010 – The International Conference on Signals and Electronic Systems, Gliwice, Poland, September 7–10, 2010.
11. *** http://www.k-space.org/ymk/medical_image_reconstruction.htm
12. COOLEY J. W., TUKEY J. W., *An Algorithm for the Machine Calculation of Complex Fourier Series. Math. Computation*, JSTOR Mathematic of Computation, **19**, *90*, 1965. Available: http://www.jstor.org/discover/10.2307/2003354?uid=3738920&uid=2&uid=4&sid=21101663284653
13. LIAO J-R., *Real-Time Image Reconstruction for Spiral MRI Using Fixed-Point Calculation*, IEEE Transactions on Medical Imaging, **19**, *7*, 2000.
14. POLDRACK R. H., MUMFORD J. A., *Handbook of Functional MRI Data Analysis,* Cambrige University Press, New York, USA, August 2011.
15. LORENTZ I., *Fitting FFT onto an energy efficient massively parallel architecture*, Proceedings of the Second International Forum on Next-Generation Multicore/Manycore Technologies, New York, USA, 2010.
16. MÎŢU B., *C Language Extension for Parallel Processing*, 2008. Available: http://arh.pub.ro/gstefan/VectorC.ppt
17. ŢUGUI A., *FFT Parallel Implementation for MRI Image Reconstruction*, U.P.B. Scientific Bulletin - Series C, **74**, *4*, 2012. Available: http://www.scientificbulletin.upb.ro/rev_docs_arhiva/fullb1f_106629.pdf

Received July 16, 2012