High-Speed Image reconstruction based on CBP and Fourier Inversion Methods

K. Rajan
Department of Physics
Indian Institute of Science
Bangalore 560 012
India.
rajan@physics.iisc.ernet.in

L.M. Patnaik
Microprocessor Applications Laboratory
Indian Institute of Science
Bangalore 560 012
India
lalit@micro.iisc.ernet.in

J. Ramakrishna
Department of Physics
Indian Institute of Science
Bangalore 560 012
India
jr@physics.iisc.ernet.in

Abstract

Computed Tomography (CT) image reconstruction algorithms such as convolution backprojection (CBP) and the Fourier inversion method (FIM) are highly compute-intensive applications for today's single processor systems. Shared memory multiprocessor systems are simple to implement. However, the shared bus used to connect the processors to memory is a major bottle neck in a shared memory multiprocessor system. We have designed a hierarchical bus-based system (HBBS) which solves the bus congestion of the bus-based system by hierarchically increasing the number of buses. The architecture is truly expandable and it retains the simplicity of bus-based systems. We have implemented the CBP and FIM algorithms on a HBBS, built using commercially available high-performance floating point digital signal processor (DSP) devices. The HBBS is found to be efficient in executing the computed tomography (CT) image reconstruction algorithms. The motivation for this work is not to develop new algorithms, but to evaluate the relative performance of well-known algorithms when applied to real practical architectures. The results show that an 8-node HBBS executes the CBP and FIM algorithms about 100 times faster, compared to the execution time of CBP and FIM algorithms on an IBM 340 series Workstation.

1. Introduction

Studies [2] have shown that, for the shared memory multiprocessor system, the common bus used to connect the processors to memory is by far the most limiting resource. The performance of a shared bus multiprocessor system can be improved considerably, by increasing the the number of buses. Another way of improving the performance is to restrict the number of processors sharing a bus to 5-8, thereby reducing the congestion on the bus. Mugde et.al [11] have proposed an architecture to improve the bus capacity. In their design, N processors are connected to M memory modules through B parallel buses. Each processor is connected to every bus and so is each memory bank. John and Lie [5] have studied the performance of a prioritized multiple-bus multiprocessor system.

A hierarchical bus-based system is shown in Fig. 1. The system is built around a cluster consisting of four ADSP 21062 DSP chips [1], sharing a common bus. The building block of the HBBS, the processor cluster, is shown in Fig. 2. Four nodes share a common bus, and a common memory. The common bus comprises of address, data, control and arbitration signals. Each node has a dual-ported internal memory of 2 Mbits. The internal memory and input/output processor (IOP) registers of the DSP node are called *multiprocessor memory space*. Multiprocessor memory space is mapped into the unified address space of each of the nodes. In a cluster, any one of the four nodes can become the bus master. Bus arbitration can be configured as prioritized or round-robin with rotating priority. Once a node becomes a bus master, it can directly read and write the internal memory space of any other node in the cluster and also access the shared memory. The address map of the processors in a cluster is shown in Fig 3.

A host processor is connected to the common bus of a cluster through a set of buffers. The host can access the shared memory module of the cluster, and the internal memories and IOP registers of each of the nodes in the cluster.

A bottleneck still exists in the nodes sharing a common bus. It has to use the common bus to communicate with any other node in the cluster. Only two nodes can communicate through the bus at any time. In order to circumvent such a drawback, the nodes in a cluster are connected in a hypercube structure using the processor link ports. All the nodes that form a cluster are connected in a hypercube structure

0-8186-7901-8/97 $10.00

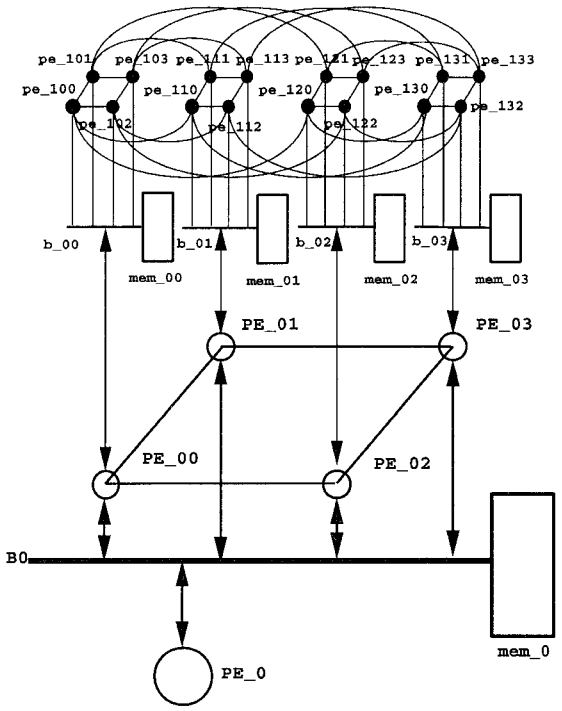

Fig 1. Hierarchical bus-based system

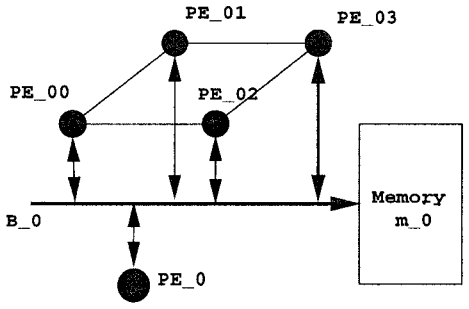

Fig 2. Processor Cluster

using link ports of ADSP 21062. The DSP chip has 6 dedicated 4-bit link ports. Each link port consists of four bi-directional data lines. All the six link ports can operate concurrently and each port can transmit/receive data at a speed of 40 Mbytes/sec. So each node in a cluster has two communication media: bus and point-to-point link port.

The scheme can be extended to build large networks of computers. Fig. 1. shows a system with 2 hierarchical levels. The node PE_0, at level 0, is the host node. Four nodes at level 1 share a common bus B_0. These for nodes PE_{00}, PE_{01}, PE_{02} and PE_{03} form a cluster, and are connected together in a hypercube topology. Each node at level 1 is connected to a cluster consisting of four nodes in level 2. All the 16 nodes at level 2 are linked together in a 4-dimensional hypercube topology using link ports.

In this paper, we describe how the CBP and FIM image reconstruction algorithms can be parallelized on a HBBS.

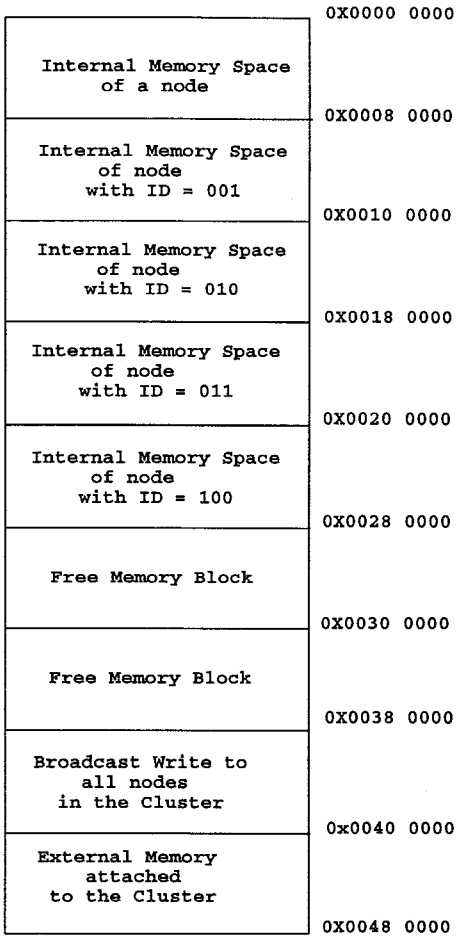

Fig 3. Cluster Address Map

In the first part of this paper, we have discussed the implementation of the convolution backprojection (CBP) algorithm on the HBBS. In the second part of this paper, we have investigated the implementation of the Fourier inversion method (FIM) algorithm. All the algorithms have been executed on two conventional systems to compare the performance of our system to execute the CBP and FIM algorithms. Computerized Tomography (CT) plays an important role in diagnostic medicine. The typical reconstruction algorithms are CBP [6] and FIM [13]. The CBP algorithm gives excellent reconstructed images when the number of projections is large, and the projection data are equi-spaced. However, the time required to reconstruct an image has been one of the major drawbacks associated with the CBP technique. To reconstruct an image of the size N*N, the computational complexity of CBP algorithm is $O(N^3)$. However, FIM technique, in which an image is reconstructed by taking the inverse Fourier transform of the spectrum of the projection data, is found to give good quality pictures at a much lower computational cost. The computational complexity of the FIM algorithm is $O(N^2 log N)$.

Algorithm improvement, dedicated hardware , and parallel processing are the three general methods to speed up the image reconstruction. A parallel implementation of a 3-D reconstruction based on CBP is reported in [3]. In [12] Shieh et al have proposed an expandable multiprocessor architecture based on digital signal processors to execute fast Radon transform and backprojection.

2. Convolution Backprojection (CBP) Algorithm

Let f(x,y) be the linear attenuation coefficient at (x,y) in one fixed plane section of an object (Fig. 4.). The projection p(s,θ) of an image f(x,y) is defined as line integral through f along the line t inclined at an angle θ from y-axis.

$$p(s,\theta) = \int_{t(s,\theta)} f du = \sum_{t(s,\theta)} f(x,y)\Delta u \qquad (1)$$

where t(s,θ) is the line s= x cosθ + y sin θ. inclined at an angle θ from the y-axis at a distance s from the origin , and u is the distance along t. In practice, values of p(s,θ) are computed for a finite number of angles and for discrete values of s.

For discrete data, the reconstruction algorithm required to approximate $f(k\Delta x, l\Delta y)$, where $0 \leq k \leq (N-1)$, and $0 \leq l \leq (N-1)$ from $p(m\Delta s, \theta_n)$ where $0 \leq m \leq (M-1)$ and $1 \leq n \leq N_{angle}$, is

$$f(k\Delta x, l\Delta y) = \Delta\theta \sum_{n=1}^{N_{angle}} \tilde{p}(k\Delta x cos\theta_n + l\Delta y sin\theta_n, \theta_n) \qquad (2)$$

where $\tilde{p}$ is the convoluted projection data at an angle θ_n, N*N is the size of the image to be reconstructed, N_{angle} is the number of views and M is the number of parallel rays in a view . The convolution operation is done in frequency domain. In the backprojection stage, the contributions from the convoluted projections of all the angles θ_n are to be back-projected on each of the image pixels $f(k\Delta x, l\Delta y)$. Since we have a discrete convoluted data, we need to go through a stage of interpolation. The linear interpolation method gives good results at a lesser computational cost. A 6-point Lagrange interpolation relation gives much better images, but at a higher computational cost. We have used the linear interpolation and the 6-point Lagrange interpolation in our studies.

3. Implementation of a Hierarchical Bus-Based System (HBBS)

A two level bus-based system has been implemented using ADSP 21062 DSP chips (Fig. 1). Eight DSP chips

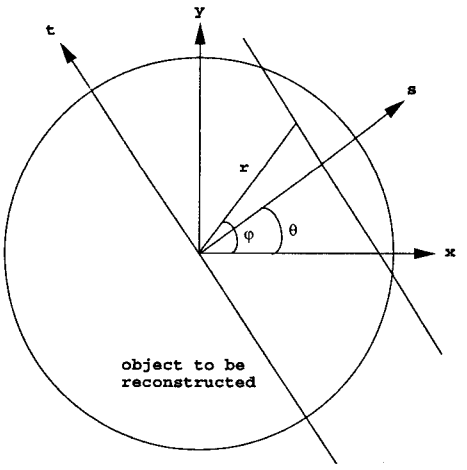

Fig 4. Parallel beam geometry

($PE_{100} - PE_{103}$ on a shared bus B_{00} and $PE_{110} - PE_{113}$ on shared bus B_{01}) form two clusters of 4 nodes each at the leaf level, and two DSPs chip are configured at the level 1 (PE_{00} and PE_{01}) and one DSP chip form the level 0 node PE_0. The two clusters at the leaf level, are connected together in a hypercube topology using the built-in links of the DSP chips. The node PE_0 can access cluster bus B_0 and memory M_0.

The node at level 0, PE_0 is connected to a Pentium PC to use the resources such as memory, keyboard, display and disk storage. The ADSP 21062 operates at 40 MHz with a cycle time of 25 ns.

4. Execution Time

PE_0 collects the projection data $p(m\Delta s, \theta_n)$ for each of the angles θ_n, n=1 to N_{angle}, from the measurement system. The N_{angle} projection data are decomposed among the P leaf level nodes cyclically. Node P_{00} gets $p(m\Delta s, \theta_0)$, node P_{01} gets $p(m\Delta s, \theta_1)$ and so on. PE_0 transfers the projection data, cyclically, to the nodes at level 1 after taking over the bus using the bus request and bus grant protocols. The nodes at level 1 transfer the projection data to the nodes at the leaf level. This pipelined communication helps in the efficient utilization of the buses at hierarchical level 1 and 2. The total communication time for all the PEs at level 2 to get one row of projection data is approximately 125 μ seconds (9 * 512 clock cycles). The nodes at level 0 and 1 execute the following steps in sequence to distribute the projection data to all leaf level nodes.

step 1. PE_0 sends proj[0] to PE_{00}.
step 2. PE_0 sends proj[1] to PE_{01}
PE_{00} sends proj[0] to PE_{100}
step 3. PE_0 sends proj[2] to PE_{00}
PE_{01} sends proj[1] to PE_{110}

step 4. PE_0 sends proj[3] to PE_{01}
PE_{00} sends proj[2] to PE_{101}
step 5. PE_0 sends proj[4] to PE_{00}
PE_{01} sends proj[3] to PE_{111}
step 6. PE_0 sends proj[5] to PE_{01}
PE_{00} sends proj[4] to PE_{102}
step 7. PE_0 sends proj[6] to PE_{00}
PE_{01} sends proj[5] to PE_{112}
step 8. PE_0 sends proj[7] to PE_{01}
PE_{00} sends proj[6] to PE_{103}
step 9. PE_{01} sends proj[7] to PE_{113}

As soon as each node at the leaf level gets one projection data $p(s, \theta_n)$ it starts executing the convolution algorithm. The convolution algorithm is implemented in Fourier domain using the following equation.

$$\tilde{p} = IFT(FT(p(m\Delta s, \theta_n) \times FT(filterfunction)) \quad (3)$$

where IFT denotes inverse Fourier transform. Meanwhile, PE_0 retrieves the next set of projection data for transfer to the nodes at the higher level.

The image data of size N*N is decomposed among P processors as follows. Each processor P_i is assigned N/P rows of image data. Processor P_i gets rows (i-1)*N/P to (i*N/P - 1). The image block assigned to PE_i is I_i. The processor P_i has been assigned N_{angle}/P rows of projection data. Node P_i computes equation (2) and updates the pixels of the image block I_i using the M/P projection data. Each image block I_i has to be updated using all the N_{angle} projection data. But N_{angle} projection data are distributed amongst the N nodes. So the updated image block at node P_i has to be circulated to P_{i+1} and the to P_{i+2} and so on to all other processors. We have used the ring topology embedded in hypercube to circulate the image block amongst all PEs. Node PE_i configures a link port as output to send the updated image block to P_{i+1}. Node PE_i configures another link as input to connect to P_{i-1} to read the image block updated by P_{i-1}. In order to overlap the communication with the computation, P_i sends an image row to P_{i+1} as soon as it completes the update task on that image row. The transfer of updated image row from PE_i to PE_{i+1}, transfer of image row from P_{i-1} to PE_i and the processing of the next image row are done concurrently. The dual-ported internal memory of the DSP chip with an independent IO processor make it possible to schedule all three operations concurrently.

Table 1 gives the execution times for CBP algorithm on a HBBS with 8 PEs at level2, 2 PEs at level 1 and one PE at level 0, on an IBM RS 6000 workstation, and on a Silicon Graphics Indigo 2 workstation. The results show that a HBBS having 8 PEs at level2, 2 PEs at level 1 and one PE at level 0 executes the CBP algorithm, about 35 times faster than a Silicon Graphics Indigo 2. Compared to an IBM 6000/340 workstation, the DSP based multiprocessor system gives 100 times better speed performance. The DSPs run at 40 MHz with zero 'wait' states. The execution times given for the multiprocessor system include the communication overhead. The Indigo 2 workstation consisted of R4400 CPU, 4410 FPU running at 150 MHz, 16 KB instruction/ 16 KB data cache, 1 MB level 2 cache and 64 MB memory. The Indigo 2 runs under IRIX 5.3 OS. The IBM RS 6000/340 consisted of RS 6000/340 CPU running at 33 MHz, 8 KB instruction/ 32 KB data cache,and 32 MB memory.The system runs under IBM AIX 3.5 OS. In both cases, the executables are created using C compiler with O2 level optimization.

5. Fourier Inversion Method (FIM)

Fourier Inversion is an efficient method for image reconstruction in a variety of applications, for example, in computed tomography and magnetic resonance imaging. The interpolation from the polar coordinate to cartesian is the most demanding operation in FIM. Many interpolations schemes such as nearest neighbor, 2-D linear interpolation and bilinear interpolations have been proposed in literature [9], [14], [10]. Noll [4] has suggested a gridding algorithm for interpolation from polar to cartesian based on interpolation by the scan line method.

The steps involved in image reconstruction based on FIM are:

1. projection measurement $p(m\Delta s, \theta_n)$, n=1 to N, m= 1 to M
2. 1-D FFT of $p(m\Delta s, \theta_n)$
3. interpolation (polar to cartesian) and
4. 2-D IFFT

Stark et.al [14] have provided the following interpolation relation to compute F(R,θ) at the cartesian point (u,v),

$$F(R,\theta) = \sum_{n=-N/2}^{+N/2} \sum_{k=0}^{N_{angle}-1} F(n/2A, 2\pi k/N_{angle}) \cdot sinc[A(R/\pi - n/A)]\sigma(\theta - 2\pi k/N_{angle}) \quad (4)$$

$$\sigma(\theta) = \frac{sin(N_{angle}/2)\phi}{N_{angle} sin(\theta/2)} \quad (5)$$

where $R = \sqrt{u^2 + v^2}, \theta = cos^{-1}(u/R)$, and A is the radius of the circle of support of the object f(x,y).

PE_0 sends first half of the projection data rows, proj[0]-proj[(N_{ang}/2)-1] to PE_{00} and the other half, proj[N_{ang}/2]-proj[N_{ang}-1] to PE_{01}. The PEs PE_{00} and PE_{01} distribute the projection data equally to all PEs at the leaf level. As shown in the section on CBP,most of the communication time overlaps with the 1-D FFT computation time.

Each PE at the highest level also gets a copy of the coordinate addresses of the pixel that lie within the projection data it has to process. Each PE computes the 1-D FFT of the projection rows. The total communication time for all the PEs at level 2 to get one row of projection data is approximately 125 μ seconds (9 * 512 clock cycles).

The second step in FIM algorithm is the interpolation from r-θ to cartesian coordinate frame. We have used equation 3 given by Stark et.al for interpolation. Fifteen neighboring pixels of each cartesian pixel (5 neighboring pixels on each projection angle, and 3 angles of projection) have been used for interpolation. When we compute the interpolation for the pixels that lie in the boundary between two PEs, a communication takes place between PE_i and PE_{i+1} to exchange one row of projection data. Similar communication takes place between PE_{i-1} and PE_i. The 2-D FFT samples are stored in the cluster memories, with the first and third quadrants stored in cluster memory M0, and the second and fourth quadrants stored in M1. A 2-D IFFT of the 2-D FFT samples give us the image function f(x,y).

Because of the aliasing problem, the FFT has to be calculated for a number of points much larger than the size of the image N*N. In our algorithm, we have doubled the sampled 2-D FFT domain to 512*512, to reconstruct an image of size 256*256.

Table 1 gives the execution times for Fourier Inversion algorithm on a HBBS with 8 PEs at level 2, 2 PEs at level 1, and one PE at level 0, on an IBM 6000/340 series workstation, and on a Silicon Graphics Indigo 2 workstation. The HBBS executes the FIM algorithm about 100 times faster compared to the time of execution of FIM algorithm on an IBM 6000 Workstation. The reconstruction time based on the FIM algorithm is much less than the reconstruction time using the CBP algorithm.

5.1 Performance

We have executed the sequential version of the CBP and FIM programs on a single ASDP 2106x processor. The results are shown in Table 1.The following performance metrics [8], [7] were computed:

1. Workload, W measures the amount of computation work performed in an application. For scientific computation and signal processing applications, a natural metric is the number of floating point operations that need to be performed.

2. Measured Computation Speed C = W /T(p), where W is the *workload* and T(p) is the total execution time of the parallel program running on p nodes, including all communication overhead.

3. Utilization, U = C/(peak performance of the chip * p) is the ratio of the measured performance to the peak performance of a p-node system.

4. Speedup, S = T(1)/T(p), where T(1) is the execution time of the best sequential program on a single node.

The component algorithms and the workloads of the CBP and FIM programs are shown in Table 2. The workload is obtained through code inspection, counting only the dominant terms. For instance, an n point complex FFT has a workload of 5*n*log n floating point operations flops.

5.2 Component algorithms of FIM program

1. 512 point 1-D complex FFT of 128 views:
2. 15 point interpolation based on Stark's [14] formula,
3. 2-D IFFT of 512*512 matrix, which consists of 512 numbers of 512 point IFFT followed by a matrix transpose and 512 numbers of 512 point IFFT.

5.3 Component algorithms of CBP

1. Convolution through frequency domain computation, which consists of the following three operations:

(a) 512 point FFT of 128 views,
(b) Complex multiplication of filter function and FT of projection data,
(c) 512 point IFFT of 128 views.

2. Backprojection operation based on linear interpolation.

The Speedup of the HBBS with 8 nodes at leaf level and 2 nodes at level 1 is found to be around 7.85.

For the FIM, we get the best performance for the 1-D FFT and 2-D FFT. However, the interpolation steps give us a poor response with utilization falling to 25%. Out of a total of 65 MFlops workload, 38 MFlops are executed at a lower utilization rate of 25 %, which gives an overall utilization of 35 % for the FIM program. The DSP chips are not very efficient in executing Stark's [14] interpolation equation (3).

In the case of CBP, since the convolution operation is implemented in the frequency domain, we get a good utilization factor of 65%. The code for FFT routine uses multi-function instructions extensively, and hence utilization factor is relatively high.

6. Conclusions

A HBBS supports larger number of processors, and still retains the simplicity of a bus-based system. The HBBS solves the bus congestion problem inherent in bus-based systems. The HBBS is found to execute the image reconstruction algorithms based on CBP and Fourier inversion very efficiently. The better performance is not only because of multiple processors in the system, but also because of the highly specialized instructions and architecture of the DSP devices In addition, the parallel algorithm developed in this paper for the CBP and FIM algorithm uses overlapped communication and computations to a large extend.

Table 1
Execution time for various reconstruction algorithms

Image size: 256 * 256, Number of detector elements: 512

Algorithms	HBBS		IBM	Indigo 2
	1 PE	8 PEs	RS 6000	
Parallel beam CBP based on linear interpolation	2.6 s	332 ms	33 s	14 s
Parallel beam CBP based on 6-point Lagrange interpolation	12 s	1.6 s	186 s	68 s
Fourier Inversion based on Stark interpolation	1.54 s	200 ms	25 s	12.8 s

Table 2
Performance of the processor node

Program	Component Algorithms	Workload (MFlop)	Execution time	Measured Speed (MFlops/s)	Utilization (%)
FIM		65.34	1.54	42	35
	1-D FFT	2.94	32 ms	92	77
	Interpolation	38.4	1.28 s	30	74
	2-D IFFT	24.1	269 ms	89	25
CBP		203.4	2.6 s	78	65
	Convolution	6.4	74 ms	86	71
	Backprojection	197	2.52 s	78	65

References

[1] Analog Device's Users Manual, ADSP 2106x SHARC (1993).

[2] J. Archibald, and J. L. Baer, " Cache Coherent Protocols: Evaluation Using a Multiprocessor Simulation Model', *ACM Trans. Computer Systems.*, pp. 273-298, Nov 1996.

[3] C. M. Chen, S. Y. Lee, and Z. H. Cho, " A Parallel Implementation of 3-D CT Image Reconstruction on Hypercube Multiprocessor," *IEEE Trans. on Nuclear Science.*, Vol. 37, No. 3, June 1990, pp. 1333-1346.

[4] Douglas C Noll, J. A. Webb, and T. E. Warfel, " Parallel Data Resampling and Fourier Inversion by Scan-Line Method," *IEEE Trans. on Medical Imaging.*, Vol. 14, No. 3, Sept. 1995, pp. 454-463.

[5] L. K. John, and Yu-cheng Liu, " Performance Model for a Prioritized Multiple-Bus Multiprocessor System," *IEEE Trans. on Computers.*, Vol. 45, No. 5, May 1996.

[6] G. T. Herman , *Image Reconstruction from Projections.*, New York, Academic, 1980.

[7] Hockney, " Performance Parameters for Benchmarking of Supercomputers," *Parallel Computing.*, Vol. 17, pp. 1111-1130, 1991.

[8] Kai Hwang, Zhiwei Xu and M. Arakawa, " Benchmark Evaluation of the IBM SP2 for Parallel Signal Processing," *IEEE Trans. on Parallel and Distributed Systems.*, Vol. 7, No. 5, May, 1996, pp. 522-536.

[9] S. Matej, and I. Bajla, " A High-Speed Reconstruction from Projections Using Direct Fourier Method and Optimized Parameters," *IEEE Trans. on Medical Imaging.*, Vol. 9, No. 4, Dec 1990, pp. 421-429.

[10] R. M. Mersereau, and A. V. Oppenheim, " Digital Reconstruction of Multidimensional Signals from Their Projections," *Proceedings of the IEEE.*, Vol. 62, No. 10, Oct 1974, pp. 1319-1338.

[11] T. N. Mudge, J. P. Hayes, and D.C. Winsor, " Multiple Bus Architectures," *Computer.*, Vol. 20, No. 6, pp. 42-48, June 1987.

[12] E. Shieh et.al, " High-Speed computation of the Radon Transform and back-projection using an expandable multi-processor architecture," *IEEE Trans. Circ. Syst. on Video Technology.*, Vol. 2, No. 4, pp. 347-359, Dec 1992.

[13] L. A. Shepp, and B. F. Logan, " The Fourier Reconstruction of the Head Section," *IEEE Trans. on Nuclear Science.*, Vol. 21, pp. 21-43, 1974.

[14] H. Stark, J. W. Woods, I. Paul, and R. Hingorani, " An investigation of Computerized Tomography by Direct Fourier Inversion and Optimum Interpolation," *IEEE Trans. on Biomedical Engineering.*, Vol. 28, No. 7, July 1981, pp. 496-505.