Research on FFT Algorithm Use SMP System

Bingfeng Qian¹,², Yize Sun¹, Qian Zhang³

¹School of Mechanical Engineering, Donghua University, Shanghai, China
²Machinery College, Shanghai Dianji University, Shanghai, China
³Department of Electronics and Electrical Engineering, Liverpool John Moores University, Liverpool, UK

Email address: qbf314403@126.com (Bingfeng Qian)

To cite this article:
Bingfeng Qian, Yize Sun, Qian Zhang. Research on FFT Algorithm Use SMP System. International Journal of Information and Communication Sciences. Vol. 5, No. 1, 2020, pp. 1-7. doi: 10.11648/j.ijics.20200501.11

Received: January 31, 2020; Accepted: February 18, 2020; Published: February 26, 2020

Abstract: Based on the theoretical research of a series of algorithms of adaptive beamforming and space-time adaptive processing, it is necessary to discuss the implementation of related algorithms on the machine hardware platform. Fast Fourier transform (FFT) is an essential process in this implementation. With the development of array antennas, the number of points to be calculated by the FFT has also increased significantly. Therefore, the ultra-large point FFT has become a key technology for current antenna signal processing. Most of the existing FFT algorithms are researched based on single-core processors, and there is very little literature on the research of super-large-point FFT algorithms suitable for Symmetric Multiprocessor (SMP). In this paper, through analyzing the characteristics of a symmetric multi-processor (SMP) parallel processing system, the very large FFT fast algorithm (VLFFT) is proposed. This algorithm significantly reduces the dependence on memory and improves the FFT's performance using the limited rules of the one-dimensional sequence split, changing the twiddle factor calculation method, and optimizing the data distribution and storage access. And the research's optimized very large FFT algorithms is also summarized in the paper. Experiment results show that the algorithm is suitable for the SMP platform and can effectively solve the problem of the very large FFT, which make single-core processors harder to realize.

Keywords: Radar Signal Processing, SMP, Storage Optimizing, Very Large FFT, Parallel Processing System

1. Introduction

The FFT is the key technology of radar signal processing and its execution efficiency determines system processing ability. The rapid development of high-resolution and large-bandwidth, phased-array radar systems has greatly increased the computational complexity of FFTs. It causes processor resource deficiency and low execution efficiency. These problems directly affect the development schedule of radar systems [1, 2].

Most existing FFT algorithms are based on single-core processors, but the literature about super-large FFT algorithms based on symmetric multi-processors (SMPs) are seldom [3, 4, 5]. The algorithm based on the GPU block FFT algorithm solves the memory leak problem caused by high-capacity images [6]. This study is helpful for SMP algorithms in radar systems; however, the method is hard to transfer to multi-processor platforms. An FFT accelerator was provided based on matrix transposition in DSP using the parallel strategy, mixed twiddle factor strategy, and cache-line pingponging space to improve FFT acceleration performance and calculations [7]. According to the modern super-scalar processor, Gao Lining provided an efficient FFT algorithm method through modeling and analysis that is based on cache optimization [8]. The method split the FFT, which provides full functionality of the cache and improves processing performance. The optimization strategy of the cache and splitting method provide good references. Multi-core processors realize large FFT algorithm splitting and parallel optimization in, which fully verify operational performance of multi-core DSP processors [9]. However, this method introduces a weighted rotation factor and dominant transposition, which decrease real-time performance of the FFT. By far, Moore’s Law and limited power have become obstacles to chip technology development. The ability of a single core processor is limited, so more SMPs have become the main processor of radar systems. It is necessary to research a large FFT suitable for SMP to improve the
performance of radar systems. This study analyzes the characteristics of SMP parallel processors and presents an algorithm suited for very large FFTs in SMPs. In order to reduce computation, the algorithm is limited to the one-dimensional series division rule and changes the factors to decrease dependence on storage resources. The algorithm is implemented on the TMS320C6678 multiprocessor DSP platform. The results of experiments show that the algorithm is suitable for the SMP platform and solves the large FFT problem. At the same time, it improves the utilization ratio of storage resources and execution performance.

2. The Parallel Processor System Based on SMP

To meet the demand of high resolution and large bandwidth applications of radar systems, it is necessary to have larger storage resources and lower processing delays in signal processing systems. Therefore, parallel computer theory is applied to real-time processing radar systems through enlarging parallel calculation scales and improving parallel efficiency to solve current problems [10, 11]. Multi-core processors have become mainstream and SMP is widely used in multi-core processor systems [12, 13]. There is no difference between each processor and the SMP assigns tasks to each processor. This means every processor can access memory, I/O, peripherals, and interrupts equally. Because the distance between each processor is closer (and the bandwidth is bigger), the processor delay is lower and performance is better. Figure 1 is the sketch map of a distributed storage structure parallel processing system based on the SMP.

3. Improved Large FFT Algorithm

The FFT function is split into a two-dimensional series as in, which results in smaller processing for FFT calculations [14, 15]. The main process is as follows: (1) the split one-dimensional series into a two-dimensional series; (2) address the FFT in the row direction; (3) multiply by the hinge factor; (4) store and transpose the result of step (3); (5) deal with the FFT in the columns; and (6) store and transpose the results. For the SMP system, there are three problems in the realization. The first is how to split to obtain the best calculation performance. The second is how to address the
problems caused by introducing hinge factors that cause extra memory and calculations. The third is dominant transpose needs for extra memory and performance will decrease sharply when single numbers are increased. This section provides a specific optimization method to address these needs.

3.1. The Rule of Sequence Division Optimization

We define the length of a one-dimensional series as $L$; the two-dimensional series is $L = M \times N$, $M$ is the number of rows, and $N$ is the number of columns. Fine-grained FFT calculation is based on the two-FFT algorithm. According to equation, realization FFT calculations are as follows [16]:

$$\begin{cases} C_{\text{mul}} = N \times M \log_2^M + M \times N \log_2^N + C_f \\ C_{\text{sum}} = N \times M \log_2^M + M \times N \log_2^N \end{cases}$$

(1)

where $C_f$ is the number of introducing hinge factors. According to the Cauchy inequality:

$$\begin{cases} C_{\text{mul}} \geq \frac{L^2}{4} \log_2^{2\tau} + C_f \\ C_{\text{sum}} \geq L^2 \times \log_2^{2\tau} \end{cases}$$

(2)

When $M = N = \sqrt{L}$, the minimum of (2) is:

$$\begin{cases} C_{\text{mul}} = \frac{L^2}{4} \log_2^{2\tau} + C_f \\ C_{\text{sum}} = L^2 \times \log_2^{2\tau} \end{cases}$$

(3)

The conclusions are deduced as follows. The minimum number of FFT complex multiplications and complex additions can be found in theory, when a one-dimensional series is split into row-equal columns. The length of the FFT calculation is $N = 2^n$, where in practice $M, N$ equals:

$$\begin{cases} M = N = 2^\frac{n}{2} \quad n \text{ is even number} \\ M = 2^\frac{n}{2}, N = 2^\frac{n}{2} \quad n \text{ is odd number} \end{cases}$$

(4)

3.2. Hinge Factors Optimization

The calculation of (3) is as follows:

$$Z(n_0, k_0) = e^{-\frac{j2\pi n_0 k_0}{L}}$$

(5)

where $n_0 = 0, 1, ..., M - 1, \ k_0 = 0, 1, ..., N - 1$.

To split (5) into a two-dimensional matrix of size $M \times N$, each row value comes from the first row. This saves a large amount of memory, which is important for realizing the large FFT algorithm. However, it also results in extra complex multiplication. Further analysis is required in order to know the influence caused by extra complex multiplication.

Using only complex multiplication to measure the operation time of the FFT (as in the principle of division mentioned in 3.1), the extra complex multiplication occupying the total multiplication is:

$$D = \frac{C_{\text{mul}} - C_{\text{sum}}}{N(M-1)}$$

(6)

Let $M = N = \sqrt{L}$, then (6) is

$$D < \frac{1}{4(L - \sqrt{L}) \log_2^{2\tau} + 2}$$

(7)

Figure 2. The graph of $D$ with an increasing FFT point.
According to (7) and Figure 2, we see that the proportion of extra complex multiplication to the total multiplication decreases with an increase in the point size. When the point size is larger than 256K, the ratio is close to 0. Therefore, for large FFT, the calculation of extra complex multiplication is little by means of hinge factor saves the outstanding memory source.

3.3. Data Distribution and Storage Access Optimization

Because the point size of very large FFT is large and three times the dominant transposition, it requires a minimum of twice the buffer space. It is impossible to place original FFT data, intermediate data, and results data all in internal storage. Off-chip memory used as big data storage is not fast enough. We discuss the relationship between FFT data distribution and the performance of storage.

Based on the features of the SMP system and the single SMP as a unit, large amount calculations were accomplished by many processors. If we place original FFT data, intermediate data, and result data all in off-chip memory, the processing ability must be decreased. Treatment points sizes become small because the FFT was split into a two-dimensional series. Obviously, we can place original data into off-chip memory then move the data into internal memory for calculations. The results are then sent back to the off-chip memory. Suppose the number of SMP processors is Q and the length of the series L = M × N. The SMP process is a large FFT as shown in Figure 3:

Data distribution and data access optimization place the large amount of original data in off-chip memory by using the Q access granularity to send the corresponding rows or columns to MSM. Each SMP processor unit accomplishes the corresponding row or column FFT calculations. Row or column FFT calculations are divided into four parts: reading data, transposition, FFT, and writing data. Figure 4 shows the circular timing of the FFT.

According to Figures 4 and 5, it is clear that reading and writing data and transposition are all hidden in off-chip memory. Reducing extra off-chip memory to dominant the transposition saves the storage and efficiency of matrix transposition. Table 1 shows the source demand of the measurements.
By optimizing very large FFT data and matrix storage access, using the core of the SMP as particle size, and internal temporary cache-line pingponging space, the access speed is faster than the directory access of off-chip memory. At the same time, rational parallel processing hides the data access and matrix transposition in the FFT processing, which saves system resources and decreases large FFT processing delays. Therefore, distribution and matrix access make the realization of large FFTs possible.

Based on the above analysis, optimized, very large FFT algorithms are shown in Figure 6.

4. Experimental Verification and Performance Comparison

Experiments were based on the AS-R robot project. The results obtained were from the DSP radar signal board. The real actual equipment is shown as in Figure 7 [17]. Synthetic system consists of one Virtex7 FPGA and four TMS320C6678 chips. Each DSP chip supports a maximum of 8GB of off-chip memory with the high-speed Rapid IO serial bus. The verification conditions and test methods are as follows: (1) The DSP kernel main frequency is 1GHz; and (2) the execution time is based on the difference between per- and post-large FFT function execution time. The results based on the SMP large FFT are shown in Table 2.

![Figure 6. The flow chart of a VLFFT.](image)

![Figure 7. The physical map of the system.](image)

![Figure 8. The comparison of the runtime of FFT.](image)
Comparing the above results illustrates that the improved algorithm is suitable for the SMP processing architecture, with the point size increasing the ability to save memory and improve performance is significantly.

5. Conclusion

For the problem of existing FFT algorithms being unsuitable for SMP parallel processing platforms, we put forward an improved large FFT algorithm. The measure is based on optimizing row and column splitting rules to reduce complex operations and change the hinge factors algorithm to optimize data distribution and reading, which decreased memory demands. The results of the experiments show that this algorithm saved 50% of the memory resources, which is suitable for the SMP system and markedly improved the FFT processing speed.

Acknowledgements

This paper is supported by Shanghai Municipal Education Commission, The construction of experimental technical team in Colleges and Universities (No. 14SY08).

References

[1] Peng Yu, Jiang Honglan, Yang Zhiming, et al. Design of general digital signal processing system based on DSP and FPGA [J]. Theory and Method, 2013, 32 (01): 17-21.
[2] J. Yang, T. Xiong, Y. Peng. A fuzzy approach to filtering interferometric SAR data [J]. International Journal of Remote Sensing, 2007, 28 (6): 1375-1382.
[3] Yang Lin, Zhong Sheng, Zhang Jia-tian. Research on Data Parallel Computation Method of FFT [J]. Computer Technology and Development, 2017, 27 (10): 91-95.
[4] Cheng Dong, Wang Weihong. Application of OpenMP in SAR Image Processing [J]. Computer Science, 2017, A1 (44): 161-163, 187.
[5] Feng Qiao. A Concurrent Collections Based Approach for Parallel Data Compress on Multi-core System [J]. Electronic Science & Technology, 2015, 2 (03): 295-301.
[6] YANG Xue, LI Xue-you, LI Jia-guo, et al. Research on Fast Fourier Transforms Algorithm of Huge Remote Sensing Image Technology with GPU and Partitioning Technology [J]. Spectroscopy and Spectral Analysis, 2014, (2): 498-504.
[7] LEI Yuanwu, CHEN Xiaowen, PEN Yuanyi. A High Energy Efficiency FFT Accelerator on DSP Chip [J]. Journal of Computer Research and Development, 2016, 07: 1438-1446.
[8] GAO Li-ning, ZHU Liang, LIU Teng-fei, LIU Feng. An Efficient FFT-Mapping Method Based on Superscalar Processor [J]. Transactions of Beijing Institute of Technology, 2016, 09: 940-946.
[9] YUAN Q, YANG Kang, ZHOU Jian-jiang. Implementation of large points FFT based on C6678 multi-core DSP [J]. Electronic Measurement Technology, 2015, 38 (2): 74-80.
[10] Vinogradov, V. I. Advanced high-performance computer system architectures [J]. Nuclear Instruments & Methods in Physics Research, 2007, 571 (1-2): 429-432.
[11] IET. High-performance computing system architectures: design and performance [Editorial] [J]. Iet Computers & Digital Techniques, 2012, 6 (5): 257-258.
[12] Zhu Yong-zhi, Zhang Dan-dan, Cao Bao-xiang, et al. Research of parallel programming techniques of hierarchical model based on SMP clusters [J]. Acta Electronica Sinica, 2012, 40 (11): 2206-2210.
[13] Tam David, Aaimi Reza, Stumm Michael. Thread clustering: sharing-aware scheduling on SMP-CMP-SMT multiprocessors [J]. Acm Sigops Operating Systems Review, 2007, 41 (3): 47-58.
[14] Zhu Y Z, Zhang D D, Cao B X, et al. Research of parallel programming techniques of hierarchical model based on SMP clusters [J]. Acta Electronica Sinica, 2012, 40 (11): 2206-2210.
[15] Tam D, Azimi R, Stumm M. Thread clustering: sharing-aware scheduling on SMP-CMP-SMT multiprocessors [J]. Acm Sigops Operating Systems Review, 2007, 41 (3): 47-58.
[16] Wang Shi-Yi. Digital signal processing [M]. Beijing: Beijing institute of technology press, 1997: 123-131.
[17] Texas Instruments. TMS320C6678 Multicore Fixed and Floating-Point Digital Signal Processor [OL].

Biography

Bingfeng Qian. B. 1982, Donghua University, Shanghai, China. PhD student, Research: soccer robot and radar signal processing.
Yize Sun is currently a Professor at the College of Mechanical Engineering, Donghua University, Shanghai, China. He has published more than 100 technical papers, more than 20 patents for invention. His current research interests include Technology and system of high-end textile equipment, intelligent equipment and robots, solar photovoltaic system integration and inverter technology.

Qian Zhang is a Senior Lecturer in Engineering Systems at the Department of Electronics and Electrical Engineering, Faculty of Engineering and Technology, Liverpool John Moores University, UK. His research interests lie in Data-driven Modelling, Multi-objective Optimal Design, Robotics and Control, and their Applications in Engineering and Transport Systems.