

# Parallel Optimization Technique of Multi-channel Input Signals' Simulation

N.Pang

*Beijing Institute of Technology, Beijing, China*

P.J. Xu

*Beijing Institute of Power Design, Beijing, China*

**ABSTRACT:** Multi-channel input signals are often used in Multiple-input Multiple-out systems. This paper first introduces HPC (High-powered Computation) optimized principle, then discusses some key techniques of generating multiple input signals. The implementation based on multicore DSP is given then. In the end the performance of this method is analyzed and evaluated. The efficiency is testified on a test system.

**KEYWORD:** Multi-channel radar echo; Parallel optimization; Parallel computation

## 1 INTRODUCTION

Multi-channel input signals are often used in Multiple-input Multiple-out radar or communication system and distributed aperture coherence synthetic radar or communication system. Building this kind of real-time simulating system not only can save a amount of manpower and material resources, but also can solve many problems much better than other methods.

For example, there are several ways to simulate echo of multiple input radar system [1]. The first way is using DDS (Direct Digital Synthesizer) chip. But it only can generate several simple echo, and can not meet the need of complicated radar echo. The second way is using FPGA(Field Programmable Gate Array) chip. But it lacks flexibility [2]. The third way is using CPU/DSP(Digital Signal Processing) chip [3]. This way has the characters of flexibility and nicety, can save more money and time, but it depends on high-powered chip, programming model and efficient algorithms.

## 2 PRINCIPLE ON HPC OPTIMIZATOION

### 2.1 Introduction of symmetric multi-processor

Parallel system structure, parallel programming model, parallel optimization are three key factors of implementing parallel computation. Fig. 1 shows the relation of the three factors [4].

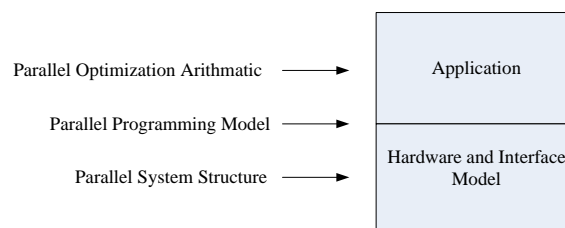


Figure 1. Three elements of parallel computation

Parallel system structure are the foundation of parallel computation. SMP (Symmetric Multi-Processor) is a typical processor with this structure. Its structure is symmetrical, and used parallel identification technique and circulation technique. In accessing memory, there are the characteristic of operating system image, low accessing delay, low communication delay. SMP is a include multiple processors which can run a independent instruction stream respectively. The representative parallel processors are IBM R50,YH-2 based on CPU and,TMS320C6678、TMS320C6609 based on DSP. Figure 2 shows elementary structure of SMP[4].

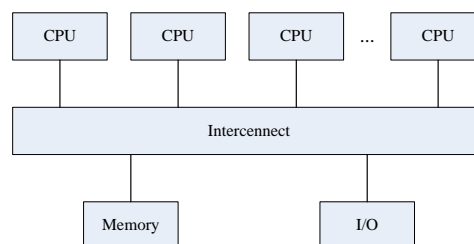


Figure 2. SMP structure sketch map

## 2.2 Shared memory parallel programming model

Parallel Programming Model provided normative programming regulation, self-contained programming interface and systemic implementation frame. These are in favor of hold the characteristic of system structure and application program synchronously. It can be realized in ways. The first is adopting bran-new programming language. The second is adopting mature parallel programming language and extending the language by paralism. The latter is better programmable and transplantable. OpenMP is a kind of Shared Memory Parallel Programming Model. It is a parallel frame based on serial language frame. In general, OpenMP API can combine with C / C++ and Fortran easily. Fig.3 is the sketch map of OpenMP model flow. In main thread, the parts that can be executed with parallel child threads fork and join alternately[5].

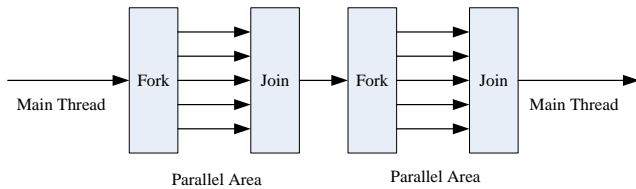


Figure 3. OpenMP model flow sketch map

## 3 DESIGN OF HIGH-POWE RED SIMULATION SYSTEM

### 3.1 Analyzation of multi-channel intercurrency

Computation parallelism usually includes three parts. They are task parallel, data parallel, pipeline parallel. Figure 4 shows all tasks of multi-channel signal simulation. Each kind of signal has identical type. There is not dependency among them. So task parallel can be realized. In the end all kinds signals adds together.

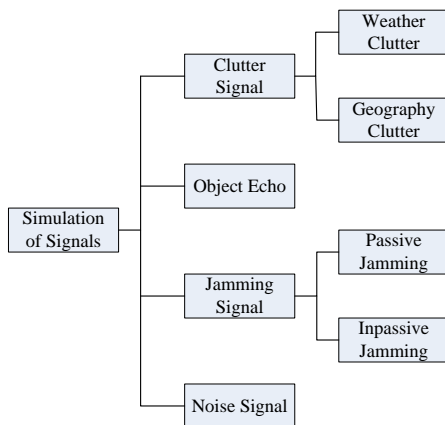


Figure 4. Constitution of multi-channel radar echo

The arithmetic can be divided into memory accessing, communication and computation. These tasks form a pipeline. The output of the first step is the input of the next step. They are the relationship of producer and consumer. So pipeline parallel can be adopted to conceal data I/O delay and communication transmission delay.

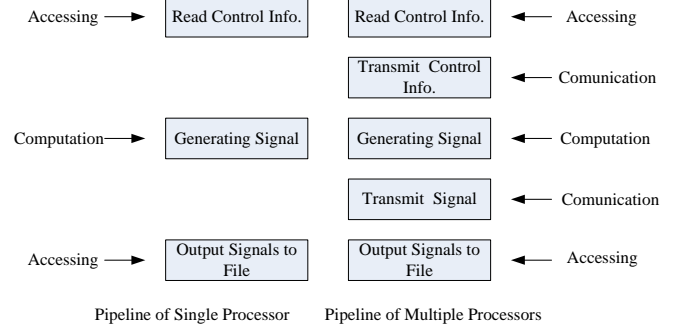


Figure 5. Pipeline Model of Single/Multiple processors

Multi-channel signals are needed to output in the end. The signals of all channels have the same generating method. So data parallel can be realized.

### 3.2 Applicability of three elements

TMS320C6678 DSP is selected because of its SMP structure. It is a highest-performance fixed/floating-point DSP that is based on TI's KeyStone multicore architecture. Incorporating the new and innovative C66x DSP core, this device can run at a core speed of up to 1.25 GHz. For developers of a broad range of applications, such as mission critical, medical imaging, test and automation, and other applications requiring high performance, TI's TMS320C6678 DSP offers 10 GHz cumulative DSP and enables a platform that is power-efficient and easy to use. TI's KeyStone architecture provides a programmable platform integrating various subsystems and uses several innovative components and techniques to maximize intra-device and inter-device communication that allows the various DSP resources to operate efficiently and seamlessly. The multicore shared memory controller allows access to shared and external memory directly without drawing from switch fabric capacity. The C6678 DSP integrates a large amount of on-chip memory. In addition to 32KB of L1 program and data cache, there is 512KB of dedicated memory per core that can be configured as mapped RAM or cache. The device also integrates 4096KB of Multicore Shared Memory that can be used as a shared L2 SRAM and/or shared L3 SRAM. All L2 memories incorporate error detection and error correction. For fast access to external memory, this device includes

a 64-bit DDR-3 external memory interface (EMIF) running at 1600 MHz and has ECC DRAM support. The high-powered program optimization and OpenMP API provide efficient support for real-time application based on the SYS/BIOS on it[6].

Multi-channel radar echo can be decomposed according to Fig.6. Task A is pattern control and synchronization; Task B is simulation of all kinds of signals; Task C is signal collection and synchronization; Task D is collecting of multiple channel signal and synchronization. Task E is signal pre-processing which is out of range of this paper. Fig.6 is a simulating system of 8 DSP chips which shows arithmetic mapping. Task A is completed by DSP0~6-core0. Task B is completed by DSP0~6-core1~6. Task C is completed by DSP0~6-core7. Task D is completed by DSP7-core0. Task E is completed by DSP7-core1~7.

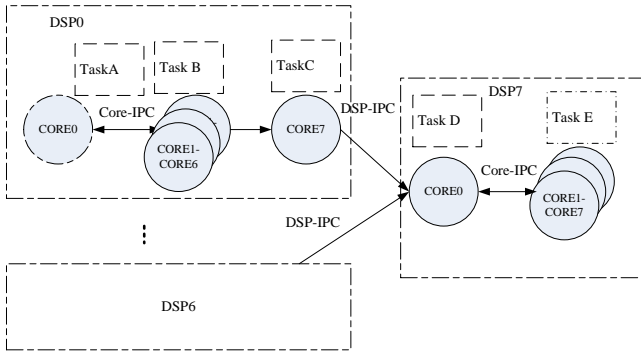


Figure 6. Task Allocation on Multiple cores of Multiple DSPs

### 3.3 High-powered parallel optimization technique

#### 3.3.1 Circulation conversion

It is supposed that radar transmitting signal is below in type (1):

$$s_T(t) = u(t) \exp\{j[2\pi f_0 t + \varphi_0]\} \quad (1)$$

$f_0$  is emission signal carrier frequency,  $\varphi_0$  is emission signal starting phase,  $u(t)$  is emission signal complex envelope. For chirp signal, the complex envelope is below in type (2).

$$u(t) = \text{rect}(t/T) \exp(j\pi\mu t^2) \quad (2)$$

After mixing, complex envelope of LFM(Linear Frequency Modulation) signal is below in type(3).

$$s(t) = A \text{rect}((t-\tau)/T) \exp(j\pi\mu(t-\tau)^2) \exp(-j2\pi f_0 \tau) \quad (3)$$

$A$  is Amplitude modulation factor,  $\tau$  is echo delay time,  $T$  is pulse width,  $\mu$  is for FM slope,  $\tau = (2R_0 - 2vt)/c$ ,  $c$  is velocity of light,  $R_0$  is range of the object,  $v$  is the speed of the object.  $T_s$

is sample period. Type (4) shows the LFM signal after discretization.

$$y(n) = \exp(\pi\mu(1+2v/c)^2 n^2 T_s^2 + (2\pi\mu(1+2v/c)(\tau_0 - 2R_0/c) + 4\pi f_0 v/c)nT_s + \pi\mu(\tau_0 - 2R_0/c)^2 - 4\pi f_0 R_0/c) \quad (4)$$

In general, the method is using function cos and sin without consideration of efficiency. But it can not meet the need of real-time simulation.

The phase of the signal can be arranged to polynomial, and FM slope and scope of frequency variation are considered.

if  $\tau_0 = -T/2 + T_s/2$ , the frequency range is  $[-B/2, B/2]$ ,  $\mu = -B/T$ .

Assumed type (5) is  $a_0, a_1, a_2$ .

$$\begin{aligned} a_0 &= \pi\mu(\tau_0 - 2R_0/c)^2 - 4\pi f_0 R_0/c, \\ a_1 &= (2\pi\mu(1+2v/c)(\tau_0 - 2R_0/c) + 4\pi f_0 v/c)T_s, \\ a_2 &= \pi\mu(1+2v/c)^2 T_s^2, \end{aligned} \quad (5)$$

Then type (6) is obtained.

$$\theta_n = a_2 n^2 + a_1 n + a_0 \quad (6)$$

The polynomial is adjusted to monopulse signal, phase coded pulse and pulse train signal. But  $a_0, a_1, a_2$  are different.

$$\begin{aligned} \Delta\theta_n &= \theta_n - \theta_{n-1} = 2a_2 n - a_2 + a_1, \\ \Delta\zeta_n &= \Delta\theta_n - \Delta\theta_{n-1} = 2a_2 \end{aligned} \quad (7)$$

If  $B = e^{j2a_2}$ , then the overlapping type can be obtained in type(8).

$$A_n = BA_{n-1}, y_n = A_n y_{n-1} \quad (8)$$

$$A_0 = e^{j(a_1 - a_2)}, y_0 = e^{ja_0} \quad (9)$$

The original value is in type (9). For object echo as monopulse signal, phase coded pulse and pulse train signal or some impulsive jamming, the complex computation can be transformed into overlapping forms which can use circulation optimization technique.

#### 3.3.2 Memory accessing spending optimization

DMA is used in this design for data moving between memory, which is not needed CPU participation. Three high-speed EDMA controllers are integrated in C6678 processor. EDMA transmitting speed between L2 or MCSR and DDR3 is showed in table 1. The result is the average value of 100 times by inquiring.

Table 1. EDMA transmission performance of concurrent execution with 8 cores

| Core number | Source address | Destination address | Size of data block | Transmit speed |
|-------------|----------------|---------------------|--------------------|----------------|
| Croe0~7     | L2             | DDR3                | 16KB               | 1212           |
| Croe0~7     | MSM            | DDR3                | 16KB               | 1311           |

### 3.3.3 Communication spending optimization

TI provides various mature IPC (Inter-processor communication) API including Notify module and MessageQ module and so on, which are based on zero-copy semaphore technique. Notify module are used in the design. For example, DSP0~6-Core0 receive control commands periodically, then transmit a Notify message to DSP0~6-Core1~Core6. DSP0~6-Core1~Core6 begin to generating signals after receiving controlling commands. Then return message to DSP0~6-Core0 to reporting completion of computation. Notify module does not copy data between cores. It transfers ownership of data in MSM, which saves much transmitting time. TI does not provide IPC between DSP chips. So DSP-IPC is developed by users based on Serial RapidIO Router configuration.

### 3.3.4 Delay concealing optimization

When there are memory accessing and communication with long-time delay, some measures must be adopted. The design uses two methods: overlapping of memory accessing and communication, overlapping of memory accessing and computation.

– *Overlapping of memory accessing and communication*

The best effective way is adopting double buffers if accessing and communication are considered. Two segments data store into two different memory. When one buffer fetches data, the other buffer starts up communication. This mechanism increase efficiency effectively.

– *Overlapping of memory accessing and computation*

The first is overlapping of EDMA(Enhanced DMA) transmission and computation. Because EDMA does not use DSP core time. After EDMA starting up transmitting, pending is not needed. The computation can execute simultaneously. Irrelevance and balance must be considered here. The second is circulation block. The efficiency of Cache lies on reuse of space and time. Spending of loading data to cache should avoid.

## 4 TEST AND ANALYZITION

Accelerating ratio testing is done in case of optimization adopted or not. With the increase of data, the accelerating ratio increase distinctly. This indicates that the parallel optimization technique are fit for the occasion of massive computation.

Table 2. Comparison between with and without optimization

| Data Length   | Without Optimization | With Optimization | Accelerating Ratio |
|---------------|----------------------|-------------------|--------------------|
| Complex 32bit | us                   | us                |                    |
| 2048          | 114.3                | 24.5              | 4.6                |
| 4096          | 212.1                | 42.8              | 4.9                |
| 8192          | 458.6                | 89.4              | 5.1                |
| 16384         | 1070.9               | 189.2             | 5.6                |
| 32768         | 1953.2               | 320.7             | 6.1                |

## 5 CONCLUSION

An attempt has been made to optimize the parallelization of programs by considering the multicore architecture, parallel programming model and parallel programming skills. These skills are composed of circulation conversion, accessing optimization, communication optimization and delay hiding, which resulted in not only considerable speedup but also much smaller scale hardware system.

## REFERENCES

- [1] Shujie Zhao. 2010. *Radar signal processing technique*. Peking: Tsinghua university press.
- [2] Liu Yi, Su Shao-ying, Chen Zeng-ping. 2010. Design of Wideband Radar Echo Simulator. *Radar Science and Technology*. 8(2):125-128.
- [3] Xu Jingli, Wang Xuegang. 2006. Digital Simulation of Radar Target Echo Based on TMS320C6201. *Modern electronic technology*. 7:60-62.
- [4] Wang Landong, Du Jing. 2012. *Parallel Optimization Technique of Imaging Radar*. Peking: Electronic Industries press.
- [5] Dheeraj D. Nitish B. Shruti Ramesh. 2012. Optimization of Automatic Conversion of Serial C to Parallel OpenMP. *2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discover*.
- [6] Texas Instruments Incorporated. 2012. TMS320C6678 multicore Fixed and Floating-Point Digital Signal Processor Data Manual.