The Research and Design of Branch Prediction based on Multicore Heterogeneous

Jin Yu_ping
Mudanjiang Normal University
Mudanjiang, Heilongjiang, China, 157012
jyphbeu@163.com

Sun Hai_jian
Mudanjiang Urban Planning
Mudanjiang, Heilongjiang, China
157000
shamohai137044@163.com

Jiao Ping
Harbin Engineering University
Harbin, Heilongjiang, China
150001
jiao.ping@hotmail.com

Abstract—Aiming at those problem that it was difficult to improve the processor performance only by improving the single core frequency, as well as superscalar pipeline stall when process a branch instruction, the architecture of heterogeneous multi-core processor which used B-Cache structure and C-Core processor controller was introduced in this paper. The new architecture avoided the pipeline flushed due to branch miss-predict, and improve overall efficiency of Multi-Core processor.

Keywords-Multi-Core; B-Cache; C-Core;E-Core

I. INTRODUCTION

With the continuous requirement of processor performance, the scholars and experts in the field of computer architecture, which have begun working to improve the performance of processor and reduce the power consumption. However, in a certain processor manufacturing process, just from the physical structure to improve processor performance for example improve the frequency of processor which tend to bring more disadvantages (such as power consumption issues). Beyond question, the traditional single-core processor architecture, technology has been facing a bottleneck [1] which we must take much time to research.

However, the recent years the Multi-Core processor technology solved the above problem very well. The Multi-Core processor integrated more than two cores which can improve processor computing capacity by parallel computing [2]. In the last two years, as the processor technology gradual maturity, there were some changes. One of the most significant was the processor architecture. That is, the traditional isomorphic Multi-Core architecture converted to heterogeneous architecture—“main processor-coprocessor”, which only had one or more common core tasks assigned functions, such as co-processor, accelerator and peripheral and so on, those functions can be realized by specialized hardware. This kind architecture can maximize the efficiency and performance of Multi-Core processor platform. However, nowadays the most Multi-Core processor commonly adopted symmetry isomorphic architecture [3], in traditional architecture of Multi-Core processor each core had its own first level Cache, and shared the second level Cache through bus.

The traditional architecture of Multi-Core processor, although it could reduce the complexity of processor, there was a performance limit. Because the traditional isomorphic architecture improved the processor performance only by increasing the core’s number of isomorphic processor instead of enhance the frequency. However, the performance can not be increased with the number of processor cores added [4].

In order to solve the above problems encountered in isomorphic architecture and improve performance, this paper introduced a new heterogeneous architecture of Multi-Core which mainly contains B-Cache structure, C-Core controller. The architecture not only optimized the process of loop instruction block, but also avoided the processors pipeline stall and flushed. So, the new kind architecture could improve processor implementation efficiency and took full advantage of resources.

II. MULTI-CORE PROCESSOR

Nowadays, the problem of Multi-Core processor was how to design the architecture and adopt what kind of communication mechanism. Those were both key technology which had great influence to processor performance.

In addition, at present the branch prediction accuracy of dynamic branch predictor could reach 84%-94%. Although, the GAs two dynamic branch predictor prediction accuracy was already very high, it would bring about 3 clock cycles consumption when miss-predict. With the pipeline and issue width of superscalar processor increase, the clock cycles of pipeline resume would be gradually increased.

A. The Reserch of Multi-Core Architecture

The architecture of Multi-Core processor can be divided into two types: isomorphic architecture and heterogeneous architecture. In the isomorphic architecture of Multi-Core, the cores were all in the same architecture, while the heterogeneous architecture was quite the contrary the cores are not same at all. Since the architecture of processor was correlation with the entire chip area, power consumption and performance, so select the propriety architecture of Multi-Core processor design was very important.

![Figure 1. Architecture of Cell Processor](image)

The Figure.1 was IBM Cell heterogeneous architecture of Multi-Core processor diagram [5]. It was basically a chip...
with one PPC hyper-threaded core called PPE and eight specialized cores called SPE which through a high-speed bus EIB connect one of PPE. The PPE included a 64-bit, dual-issue, dual-thread, the order of implementation of the core computing, it was responsible for running the operating system; The SPE was responsible for the completion of a specific floating-point operations, matrix computation, scientific computing, process of multimedia data.

The single-precision floating-point operations can be achieved peak 201GFLOPS work at 3.2GHz frequency in the Cell heterogeneous Multi-Core processor [6, 7], but at the same frequency, the Intel Pentium 4 processor peak was only 25.6GFLOPS. It indicated that the heterogeneous architecture was superior to isomorphic Multi-Core processor [8]. So it was one of reason, this paper adopted heterogeneous architecture of Multi-Core processor.

B. The Research of Data Reuse

Data Reuse which was come from a well-known program optimization mechanism. This mechanism can dynamically recording the information which was procedure in the implementation process of complex calculations, and in all possible cases, by reusing the results information to avoid more complex calculations [9].

The Figure.2 is a schematic diagram of data reuse mechanism to work, front-end parts of each take to obtain an instruction that will trigger the mechanism for data reuse, reuse mechanism to first determine whether the instructions are reusable candidates. If the candidate can be reused and instructions and historical examples of the implementation of the conditions match, then the reuse mechanism on the historical examples of reuse and discard already fetched instructions. Otherwise, the normal execution of instructions and the results recorded in the historical structure.

III. ONE KIND HETEROGENOUS ARCHITECTURE OF MULTI-CORE

Aiming at the branch prediction recovery in current Heterogeneous multi-core processors structure, this paper introduced a kind heterogeneous architecture of Multi-Core processor as the Figure.3 shown which was based on the research and analysis of Data Reuse technology. The new kind of architecture had five cores, one C-Core was used to instruction scheduling, and the other four E-Cores responsible for instruction implementation.

The program execution process was as follows:

The branch predictor would take part in i-Cache fetch instruction work [10]. The b-cache would fetch instruction from the opposition predicted path and sent those instructions into b-cache buffer.

b) The b-cache decoded those instructions.

c) Issue Station was used for dispatching instruction to C-Core processor controller.

d) After C-Core checked the marked instruction which from the Issue Station, considered the instruction implementation state in the each E-Core, used the C-Core scheduling algorithm select a proper E-Core, then sent instruction into the selected one and Wait processing.

e) E-Core was used for implementing the instruction which were sent by C-Core.

f) Ins reorder was used for reordering the out-of-order instruction sequence that was flowed into Ins reorder station, sorted those instruction accordance with the initial instruction sequence. At last, Ins reorder submitted instruction sequence by the Commit in the next clock cycle.

g) Commit submitted the sequence of instruction which have been process by Ins reorder.

IV. BRANCH PREDICTOR

At present, the prediction precise can be arrive at 83.4%-97.5%. Figure.4 shown was the classic branch predictor. Although those predictor precise were not the same, it would be consume three more clock cycle when arise miss prediction [11]. And because branch predict resume problem, which decrease the performance of processor. Aiming at the above problem, this paper introduce
B-Cache controller fetched the instruction from the path which was opposition of Branch prediction, and sent instruction into B-Cache instruction buffer, at the same time make mark to those instruction. For example tag1 (The tag1 was branch instruction mark, which marked the opposition instruction sequence), fetch width was definite n(the n was the width of superscalar pipeline). Then the Pseudo-decoder decoded those instructions, and store those decoded instructions in Pseudo-decoder buffer.

When select the one of E-Cores by the arithmetic of C-Core, if the execution of branch instruction brought the signal of miss-predict, the B-Cache would search the relevant “tag” mark, when hit the decoded instruction which were stored in Pseudo-decoder buffer would be sent into the E-Core. This mechanism reduce the clock cycle did not other like the past which need fetch and decode instruction.

V. C-CORE CONTROLLER

C-Core was the most important part of the entire heterogeneous architecture of Multi-Core processor. Since C-Core was not only responsible for receiving instruction from the Issue station, but also to deal with those instruction then make use of the C-Core scheduling algorithm sent those processed instruction to a proper E-Core. The C-Core scheduling algorithm would ensure E-Cores can efficient parallel working.

A. Dynamic Instruction issue

C-Core adopted the Dynamic Instruction issue technology. It accorded the instruction implement state of all E-Cores, then dynamic controlled the number of instruction flow to E-Core. If C-Core encountered branch instruction, C-Core would calculate burthen of E-Core (use of the follow two Equation), then select a smaller burthen E-Core to implement the sequence instruction which fetched from the branch instruction predicted opposite path, and the same time marked the selected E-Core.

When the E-Core processing the branch instruction, if this branch instruction was miss-predict, the instruction stream would be flushed in the next instruction clock cycle which were implemented in the marked E-Core, but because on the opposite path of the instruction sequence had been fetched into the marked E-Core in the beginning of implementation, so the miss-predicted didn’t stall the pipeline, if reduced clock cycle punishment. However, it would be still reduce the processor performance, so it required a more proper branch prediction mechanism in Multi-Core processor.

B. C-Core Algorithm

C-Core scheduling algorithm not only can co-ordinate and optimize the instruction steam implement in E-Cores. The algorithm consider server factors as follow: included E-Core’s implement saturation S, E-Core’s correlation R, E-Core’s data switch frequency F, The branch instruction miss-predicted rate M, and a branch instruction prediction accuracy rate T. C-Core controller calculated two results according to the following equations, then dynamic controlled the number of instruction sent to E-Cores.

The number of instruction calculating Equation:

\[
N = \frac{1}{S \times R \times F}
\]  

As Equation (1) shown, the factor S was the implement saturation of E-Core which was the percentage of all current implementing instruction, if the E-Core looked like a single pipeline, S reflects the speed of pipeline flow (the number of instruction in the pipeline).

The factor R was the implement instruction correlation of E-Cores. For example, if an instruction in the E-Core1 related with an instruction in the E-Core2 (The E-Core1 need the data which was in the E-Core2, the R was 1), then their correlation can be seen as one, and the R was the in-degree percentage of the total correlation in each E-Core (total correlation degree was 1/2 sum of in-degree and out-degree).

The factor F was the switch data frequency of E-Core which was used to improve processor performance when encounter the miss-predicted.

E-Core performance calculation Equation:

\[
P = \frac{T}{M} \times N
\]  

In the Equation (2), the factor P was represent processor performance, while T and M respectively represent the percentage of branch prediction right and wrong. The equation calculation results can be reflected one E-Core performance which considered the branch predictor accuracy.

C. C-Core Algorithm Implement

The Figure.5 and Figure.6 respectively was the instruction state and the correlation of E-Core at the time t. According to those two chart, C-Core calculated the results from Equation (1) and Equation (2), then dynamic changed
the number of instruction sent to E-Core, which avoid the low efficiency of busy uneven mode, and reduce the processor processing time, improve the E-Core work efficiency[12,13].

This paper introduce a new kind heterogeneous architecture of Multi-Core processor that adopt a B-Cache structure, which could avoid the pipeline stall that brought by miss-predicted, reduced clock cycle that the past stall and wait, improved the efficiency of the pipeline. The kind heterogeneous architecture of the Multi-Core processor not only take full advantage of Multi-Core processor resources, as well as improve the overall processor performance by C-Core scheduling algorithm. The research of heterogeneous structure of Multi-Core processor and branch prediction resume mechanism had high value in high-performance processor design.

REFERENCES
[9] Iliksen and N. Dutt, Data reuse driven energy-aware MPSoC co-synthesis of memory and communication architecture for streaming applications, in Proc. of CODES+ISSS, 2006:294-299P.

VI. CONCLUSION

Just as the much potential more than the past single core processor, the Multi-core processor architecture and technology had been widely adopt in the design of processor. The problem how improve the performance of multi-core is very important.