ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

Analysis of Features Affecting the Time Predictability

Tanushree Bhilare[1], Kranti Wagle[2]

PG Student, Dept. of Electronics, FRCRCE, Mumbai, Maharashtra, India[1]

Assistant Professor, Dept. of Electronics, FRCRCE, Mumbai, Maharashtra, India[2]

ABSTRACT: The time predictability in real time systems is very important. An ideal real time system must provide good time predictability along with good average-case performance which is difficult. The effect of micro-architectural features such as caches and pipelines on time predictability has already been analyzed. In this paper, we analyze the effect of out-of-order execution and superscalar pipelining on the architectural time predictability. To the best of our knowledge this work has not been done previously. The results help us in understanding the exact effect as the analysis performed is quantitative.

KEYWORDS: Architectural Time Predictability (ATP), Out-of-order Execution, Superscalar Pipelining, Real Time Systems.

I.INTRODUCTION

In real time systems, time predictability is a very important design parameter like average-case performance. This is due to the fact that unpredictability in hard and safety-critical real time systems may lead to a deadline miss causing fatal damage to the user or the system.

The micro-architectural features used in advanced processor such as caches, pipelines etc. improve the average-case performance but degrade the time predictability. On the other hand, the measures taken to improve time predictability, such as cache locking etc. degrade the average-case performance. Hence a system giving very good average case performance can lead to very poor time-predictability and vice versa. An ideal real time system must provide good time-predictability with good performance which is very difficult to achieve given the conflicting requirements of both the parameters.

There is a wide range of factors which affect the time predictability of a system. This includes both hardware and software features. The micro-architectural features such as caches, pipelines, out-of-order execution etc. affect the time predictability. In software, the instruction set architecture, the run-time inputs and the operating system used effect the time predictability.

The time predictability can be defined in many ways. It can be categorized as the predictability due to hardware and that due to software. In this paper, we focus on the architectural time predictability (time predictability due to hardware). We analyse the effect of out-of-order execution and superscalar pipelining on the architectural time predictability. The effect of micro-architectural features such as caches and pipelines has already been evaluated.

II.RELATED WORK

The related work includes various approaches to define the time-predictability and to measure the same and various papers which propose modified architecture designs to provide greater time predictability. Thiele and Wilhelm[9] discuss the threats to the time predictability and classify them into several categories such as architectural features, software features, task level features and distribution operation.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

They also define the time-predictability as the pessimism in the WCET analysis and that in BCET analysis. Grund et al[3] propose a template for defining the time predictability and classify the time predictability as state-induced predictability (SIP) and input-induced predictability (IIP).

Kirner and Puschner[4] propose a universal definition of time predictability and provide a formula to calculate the time predictability. Ding and Zhang[2] propose a definition of the architectural time predictability (ATP) and propose a metric, architectural time predictability factor (ATF), to measure ATP quantitatively. Ding and Zhang[1] define of the architectural time predictability and propose standard deviation of CPI as a metric to measure the same quantitatively. Edwards and Lee[8] propose a new type of processors which they call Precision Timed Machine (PRET) whose temporal behaviour can be predicted more easily.

Paolieri et al[7] propose a new multi-core architecture in which a request to a shared resource from a Hard Real-time Task (HRT) can be delayed by any other task only for a fixed time interval. Witham and Audsley[5] modify a superscalar out-of-order CPU core to exploit instruction-level parallelism to achieve better time predictability and evaluate the same for WCET analysis.

III.PROPOSED WORK

Our work is based on the previous work by Zhang and Ding[1]. We use their definition of the architectural time predictability which can be given as follows:

Definition 1: Given a number of instructions, architectural time predictability indicates the degree that the architecture under study can provide predictable execution time.

In above definition, the number of instructions is assumed to be known. Hence for a given ISA, the above definition separates the time predictability due to hardware from that due to software. The metric used to measure the architectural time predictability quantitatively is the standard deviation of CPI as proposed by Zhang and Ding[1]. The formula for standard deviation of CPI is given as follows:

$$standard\ deviation\ of\ CPI = \sqrt{\frac{\sum_{i=1}^{n}(e_i - e_{avg})^2}{n}} \quad (1)$$

Where,

e_i = Execution cycles taken by the ith instruction (CPI)

e_{avg} = Averaged execution time for total number of instructions

n = Total number of instructions in given benchmark

Note that here the CPI accounts for cycles per individual instruction i.e. the execution time in cycles for individual instruction. The standard deviation of CPI denotes the variation in the execution times of instructions. Hence it indicates the amount of architectural time predictability for a given architecture.

We use the standard deviation of CPI as a metric to investigate the effects of out-of-order execution and superscalar pipelining on the architectural time predictability. It is predicted that the use of above measures which improve performance are detrimental to the time predictability. But to the best of our knowledge, the quantitative analysis of these micro-architectural features has not been done in the past.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

IV.SIMULATION

We simulate a superscalar processor using the SimpleScalar[10] architecture. The simulator is modified to measure the execution cycles of individual instructions. Once these values are obtained standard deviation is calculated. The architectures evaluated can be given as follows:

- A superscalar processor with out-of-order execution
- A superscalar processor with in-order execution
- A scalar processor with in-order execution

The values of standard deviation of CPI obtained using the processor of first type are compared with that using the second and third type of processors. Note that the configuration used for evaluating the effect of out-of-order execution is different from that for superscalar pipelining.

TABLE I

REAL-TIME BENCHMARKS USED

Name	Description	Multi-path	Inputs
fibcall	Simple iterative Fibonacci calculation	No	Single
sqrt	Square root function	No	Single
bsort100	Bubble sort program	Yes	Two
insertsort	Insertion sort	Yes	Three
qsort-exam	Non-recursive quick sort	Yes	Three

TABLE II

(a) THE PROCESSOR CONFIGURATION FOR OUT-OF-ORDER EXECUTION

Parameter	Value
Pipeline	2-IFQ, 32-RUU, 32-LSQ Fetch speed 2, Decode width 8, Issue width 8, Commit width 8
L1 I-cache	direct-map, 32-byte block, 16K bytes, 1 cycle latency
L1 D-cache	4-associativity, 32-byte block, 16K bytes, 1 cycle latency
Unified L2 cache	4-associativity, 64-byte block, 256K bytes, 6 cycle latency
Memory	Unlimited, 100 cycle latency

TABLE II

(b) THE PROCESSOR CONFIGURATION FOR SUPERSCALAR PIPELINING

Parameter	Value
Pipeline	4-IFQ, 16-RUU, 8-LSQ Fetch speed 1, Decode width 4, Issue width 4, Commit width 4
L1 I-cache	direct-map, 32-byte block, 16K bytes, 1 cycle latency
L1 D-cache	4-associativity, 32-byte block, 16K bytes, 1 cycle latency
Unified L2 cache	4-associativity, 64-byte block, 256K bytes, 6 cycle latency
Memory	Unlimited, 100 cycle latency

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

We use five benchmarks from the Mälardalen benchmark suit[6] as mentioned below. Two of our benchmarks are single path whereas the remaining three are multi-path. The multi-path benchmarks are evaluated using different inputs. We calculate the standard deviation of CPI for all the instructions in a given benchmark as well as for different instruction types such as Load, Store and control instructions.

V.RESULTS

A. EFFECT OF OUT-OF-ORDER EXECUTION ON THE ATP AND THE PERFORMANCE

As mentioned earlier the simulated superscalar processor is evaluated using both in-order and out-of-order execution. Fig. 1 demonstrates the comparison between the values of standard deviation of CPI for the in-order and out-of-order execution.

It can be seen in Fig. 1(a), that for load instructions, the standard deviation of CPI increases by -0.09%. This percent increase is calculated for the average standard deviation of CPI for all the given benchmarks. The negative sign appears due to the fact that for benchmark bsort100_worst, the standard deviation of CPI for in-order execution is greater than that of out-of-order execution. In case of Store instructions, the standard deviation of CPI increases by 16.32% when switched from in-order to out-of-order execution (Fig. 1(b)). The control instructions show 3.53% increase in standard deviation of CPI as can be seen in Fig. 1(c). The positive value is percent increase indicates decrease in the architectural time predictability.

Fig. 1(d) shows the overall impact of out-of-order execution as it takes all instructions in a given benchmark into account. The use of out-of-order execution decreases the architectural time predictability for overall system as the standard deviation of CPI increases by 3.61% for out-of-order execution when considering all instructions.

The effect on the performance is also evaluated and can be seen in Fig. 2. The average-case performance is measured in terms of the total number of cycles required by the benchmark to complete its execution. The execution cycles decrease by 22.66% when using out-of-order execution. This indicates significant improvement in the average-case performance of the system.

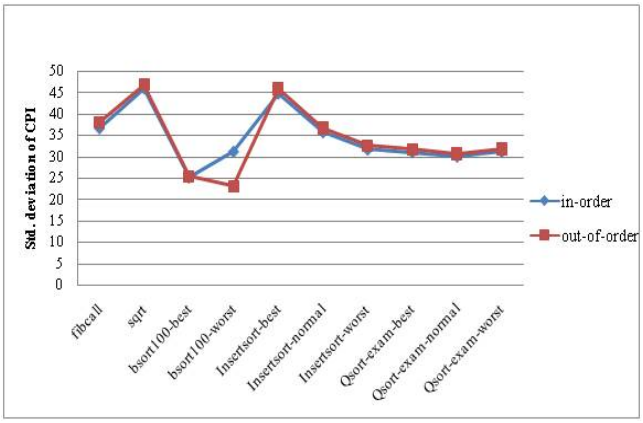

Fig. 1(a) Standard deviation of CPI for Load instructions

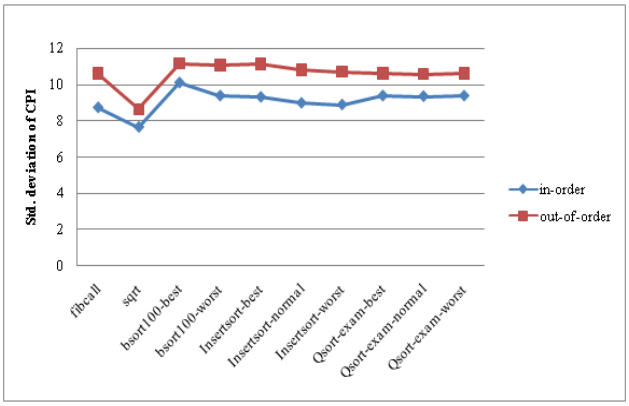

Fig. 1(b) Standard deviation of CPI for Store instructions

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

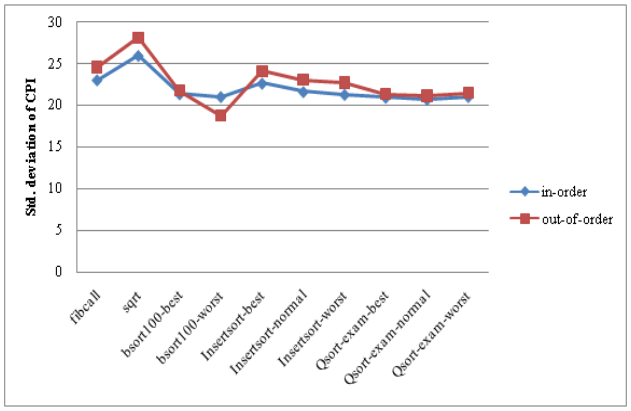

Fig. 1(c) Standard deviation of CPI for Control instructions

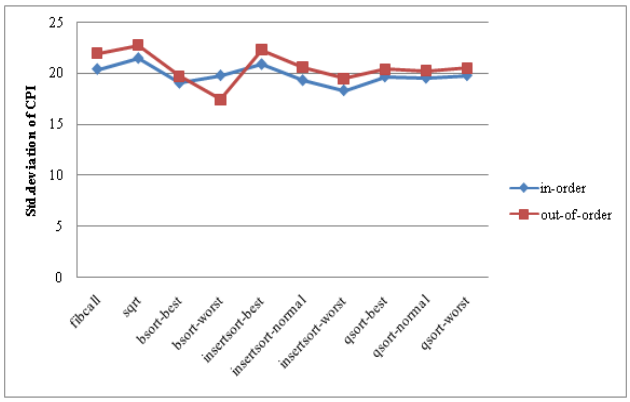

Fig. 1(d) Standard deviation of CPI for all instructions

Fig. 1 The comparison between the ATP when using in-order and out-of-order execution.

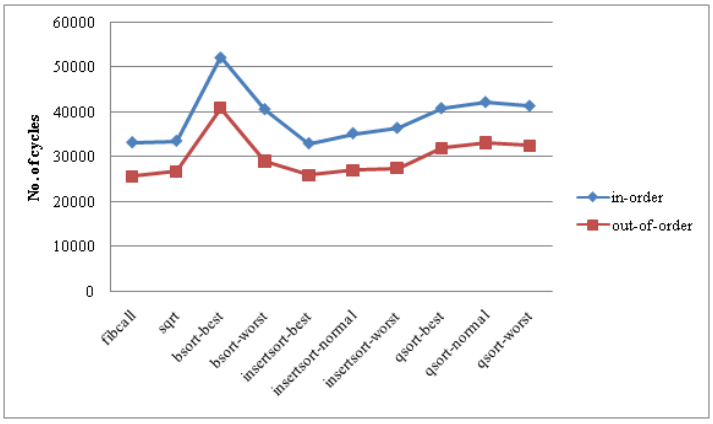

Fig. 2 The comparison between performances when using in-order and out-of-order execution

B. THE EFFECT OF SUPERSCALAR PIPELINES ON THE ATP AND THE PERFORMANCE

Fig. 3 demonstrates the effect of superscalar pipelines by comparing it with a processor having single pipeline (a scalar processor). The standard deviation of CPI increases by 4.92% for Load instructions when using superscalar pipelines (Fig. 3(a)). For Store instructions, the standard deviation of CPI increases by 6.39% for superscalar processor as compared to scalar processor (Fig. 3(b)). The control instructions show 1.48% increase in the standard deviation of CPI for superscalar processor (Fig. 3(c)). Note that the percent increase is calculated for the average value of standard deviation of CPI for all the given benchmarks. The increase in the standard deviation of CPI indicates degradation of the architectural time predictability.

Fig. 3(d) shows the overall impact on the system due to use of superscalar pipelines. The standard deviation of CPI increases by 6.82% for a superscalar processor when considering all the instructions in a given benchmark. This indicates that the overall time predictability degrades due to the use of superscalar pipelines.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

Fig. 4 shows the effect of superscalar pipelines on the performance. The execution cycles decrease by 24.02% for superscalar processor as compared to a scalar processor. Hence the average-case performance increases significantly when using a superscalar processor.

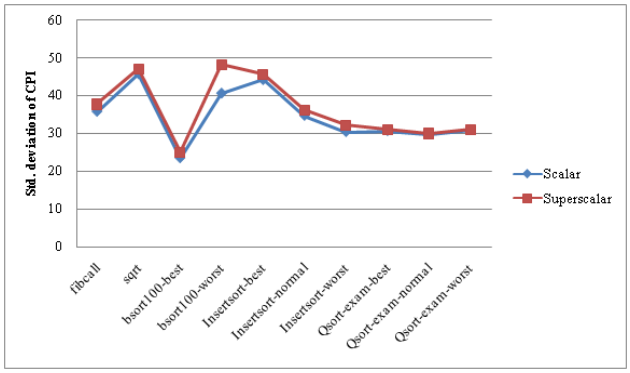

Fig. 3(a) Standard deviation of CPI for Load Instructions

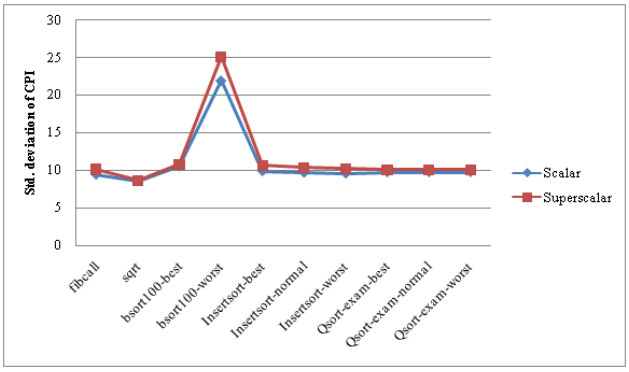

Fig. 3(b) Standard deviation of CPI for Store instructions

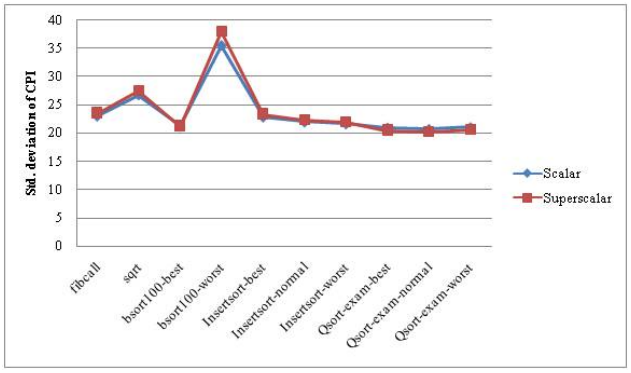

Fig. 3(c) Standard deviation of CPI for Control instructions

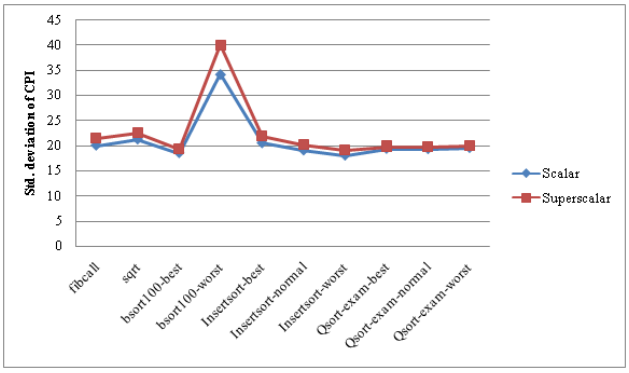

Fig. 3(d) Standard deviation of CPI for all instructions

Fig. 3 The comparison between the ATP due to scalar and superscalar pipelines

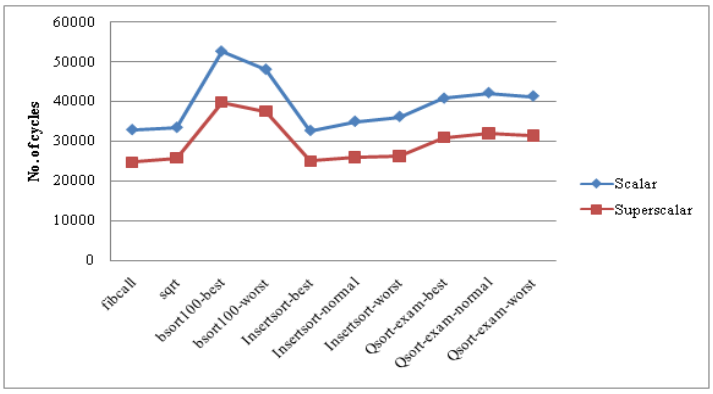

Fig. 4 The comparison between the performance when using scalar and superscalar pipelines

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

VI.CONCLUSION

The superscalar pipelines and the out-of-order execution are important measures to improve the average-case performance of a system. But their effect on time-predictability was only guessed in the previous work. This paper analyses the effect on the architectural time predictability due to the use of superscalar pipelines and the out-of-order execution quantitatively. For this, we have used the standard deviation of CPI as a metric.

The results of our experiments show that the use of superscalar pipelines and out-of-order execution degrades the architectural time predictability by a small margin whereas it improves the performance significantly. Hence it can be concluded that in case of hard and safety-critical real-time systems the use of above two features must be avoided. Instead alternative measures such as [5] can be used to improve the performance. In case of soft real-time systems, the above mentioned features can be used as the ATP degradation is very small.

In future work, the alternative measures which can improve performance without sacrificing time predictability can be found. Also the existing measures can be evaluated to analyse their effect on the time predictability.

REFERENCES

[1] Y. Ding and W. Zhang, "Exploiting Standard Deviation of CPI to Evaluate Architectural Time-Predictability", Journal of Computing Science and Engineering, Vol. 8, No. 1, pp. 34-42 , 2014

[2] Y. Ding and W. Zhang, "Architectural time-predictability factor (ATF): a metric to evaluate time predictability of processors," ACM SIGBED *Review*, vol. 9, no. 4, pp. 6-15, 2012.

[3] D. Grund, J. Reineke, and R. Wilhelm, "A template for predictability definitions with supporting evidence," in Bringing Theory to Practice: Predictability and Performance in Embedded Systems, Wadern, Germany: Schloss Dagstuhl Leibniz-Zentrum für Informatik, pp. 22-31, 2011.

[4] R. Kirner and P. Puschner, "Time-predictable computing," in Proceedings of the 8th IFIP WG 10.2 International Workshop on Software Technologies for Embedded and Ubiquitous Systems, Waidhofen/Ybbs, Austria, 2010, pp. 23-24.

[5] J. Witham, N. Audsley, "Time-Predictable Out-of-Order Execution for Hard Real-Time Systems" IEEE Transactions on Computers, Vol. 59, No. 9, September 2010

[6] J. Gustafsson, A. Betts, A. Ermedahl, and B. Lisper, "The Malardalen WCET benchmarks: past, present and future," in Proceedings of the 10th International Workshop on Worst-Case Execution Time Analysis, Brussels, Belgium, 2010, pp. 136-146.

[7] M. Paolieri, E. Quinones, F. J. Cazorla, G. Bernat, and M. Valero, "Hardware support for WCET analysis of hard realtime multicore systems," in Proceedings of the 36th Annual International Symposium on Computer Architecture, Austin, TX, 2009, pp. 57-68.

[8] S. A. Edwards and E. A. Lee, "The case for the precision timed (PRET) machine," in Proceedings of the 44th annual Design Automation Conference, San Diego, CA, 2007, pp. 264-265.

[9] L. Thiele and R. Wilhelm, "Design for time-predictability," in Design of Systems with Predictable Behaviour, Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik, 2004.

[10] SimpleScalar, http://www.simplescalar.com

BIOGRAPHY

Tanushree N. Bhilare is currently a M.E. student in the Electronics Department of Fr. Conceicao Rodrigues College of Engineering (University of Mumbai). She has completed her B.E. degree in Electronics and Telecommunication from Watumull Institute (University of Mumbai) in 2014. Her research interests are in Real-time Systems, Embedded Systems and Computer Architecture.

Prof. Kranti K. Wagle is an assistant professor in the Electronics Department at Fr. Conceicao Rodrigues College of Engineering (University of Mumbai). She has received M.E. degree in Embedded and Real Time Systems from Anna University, PSG Tech, Koimbatore. Her research interests are in Embedded and Real Time Systems, Computer Architecture. She is a member of IEEE.