Load Value Prediction Method to Utilize Prediction Miss Bias

Nobuhiro Moriwaki, Lin Meng, Shigeru Oyanagi

***Abstract*—Data hazard caused by load instruction is a serious problem for superscalar processors to improve instruction level parallelism. Various load value prediction methods have been proposed to reduce the data hazard. If the prediction is correct, successive instructions can be executed in parallel without waiting for memory access. However, prediction miss causes performance loss by squashing and re-executing successive instructions. Hence, load value prediction must balance the number of predictions and accuracy of predictions.**

This paper proposes a new load value predictor by forcussing prediction miss bias. We found that load value prediction miss tends to be biased to specific instructions. The proposed method provides a specific mechanism for predicting miss biased instructions, By adding this mechanism to a baseline predictor, accuracy of prediction can be increased. The result of experiment shows the effectivemess of the proposed method.

***Index Terms*—superscalar processor, Load prediction, instruction level parallelism, architecture**

I. Introduction

Current processors use deeper pipeline and wider instruction issue width for exploiting instruction level parallelism. It causes heavy performance loss by pipeline hazard. In addition, the gap between CPU cycle time and memory access time is getting greater. Hence, data hazard caused by load instruction is a serious problem to improve instruction level parallelism.

Load value prediction is an effective method to reduce data hazard when the successive instructions depend on a load instruction. In load value prediction, the predictor predicts the loaded value and the dependent instructions can be executed by using the predicted value without hazard.

On the other hand, load value prediction causes serious performance loss when the prediction is missed. In this case, successive instructions to the missed load instruction which have already been executing must be squashed and re-executed again. Therefore, the goal of the load value prediction is to increase prediction rate and to decrease prediction miss.

In this article, we propose a new load value prediction method which utilizes the bias of prediction miss. This paper shows that the prediction miss is biased to a small set of load instructions. By preparing dedicated mechanism for biased load instructions, accuracy of load value prediction can be increased. Since the size of the miss biased instructions is small, increase of hardware amount is small.

N.Moriwaki is with the Guraduated School of Information Science and Engineering, Ritsumeikan University 1-1-1 Nojihigashi, Kusatsu, Shiga, 525-8577 Japan (e-mail:is0043vp@ed.ritsumei.ac.jp

L.Meng is with the Department of Electronic and Computer Engineering, Ritsumeikan University 1-1-1 Nojihagashi, Kusatsu, Shiga, 525-8577 Japan (e-mail:menglin@fc.ritsumei.ac.jp)

S.Oyanagi is with the College of Informaiton Science and Engineering, Ritsumeikan University 1-1-1 Nojihigashi, Kusatsu, Shiga, 525-8577 Japan (e-mail:oyanagi@cs.ritsumei.ac.jp)

The rest of this paper is organized as follows. Section 2 explains related works on load value prediction. Section 3 shows performance of load value prediction. Section 4 discusses the bias of load value predictions miss. Section 5 proposes a mechanism for utilizing miss prediction. Section 6 shows the result of evaluation for proposed mechanism. Section 7 gives the conclusion and further problems.

II. Load Value Prediction

Various load value predictors have been proposed. Figure 1 shows the basic load value predictor, which utilizes a table consisted of tag, confidence information, and load value information. Tag keeps the load instruction address, and load value can be predicted by the load value information. When a load instruction is fetched, the predictor accesses to the table. If the tag matches to the instruction address, then the matched entry is accessed and load value is obtained by the information stored in the entry.

When load instruction is committed, the predictor updates the load value information by the committed value.

Since miss prediction penalty is large, the mechanism to reduce miss prediction is necessary. Confidence information is included in the load value information. Confidence information consists of 2-bit saturating counter, which is incremented when the prediction is correct, and is reset to zero when the prediction is missed (which is called miss resetting counter) as shown in Fig,2. If the confidence value is greater than 1, then the predicted value is used, otherwise predicted value is not used.

Several load value prediction methods have been proposed. They are as follows.

- Last value predictor[1],[2] : it is the simplest predictor which uses the last load value as the prediction value[1]. If the same value is always loaded, then this method is useful.
- Stride predictor[3] : it uses a history of load values to find a stride, and predicts load value by the stride pattern. The predicted load value can be obtained by addition of last value and stride.
- Differential Finite Context Method (DFCM)[4]: DFCM is an extended version of stride predictor which uses several stride histories to find the changing pattern to predict the load value [2]. If the latest history of the load values matches to one of the stride patterns, then the predicted load value can be obtained by applying the last value to the pattern.

III. performance of load value prediction

At first, SimpleScalar Tool Set[6] is used to evaluate the precision of load value prediction. The organization of the processor is given in Table I. The SimpleScalar PISA is used

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

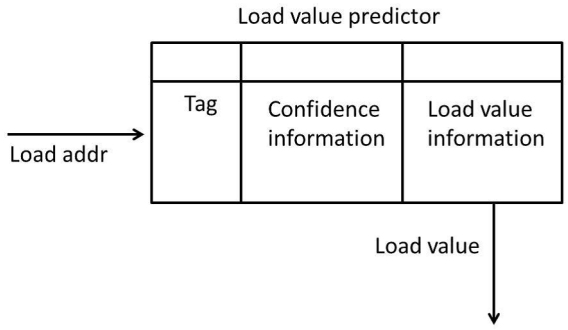

Fig. 1. Load value prediction diagram.

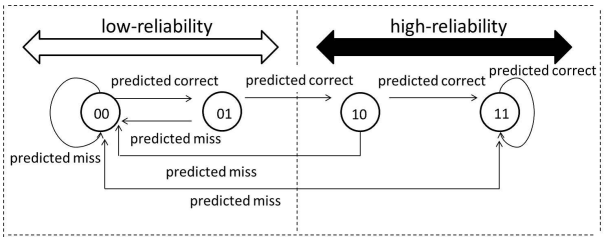

Fig. 2. miss resetting 2bit counter

for instruction set. The benchmark programs of bzip, gcc, gzip, mcf, parser, vortex, and vpr from SPECint2000 are used. Last value predictor is used as the baseline mechanism.

Here, we define evaluation items of prediction rate and prediction accuracy for load value prediction. They are defined as the following expressions.

$$\text{Prediction rate} = \frac{\text{number of prediction}}{\text{number of load instruction execution}}$$

$$\text{Prediction accuracy} = \frac{\text{number of successful prediction}}{\text{number of prediction}}$$

Fig.3 shows the evaluation of load value prediction without confidence mechanism for each benchmark program. Each bar shows the ratio of correct prediction, incorrect prediction, and no prediction from the bottom. No prediction means that the instruction is not existed in the predictor. It shows that the average prediction rate is more than 90%, and average prediction accuracy is less than 50%. It can be thought that accuracy is too low for practical use because of large miss penalty.

On the other hand, Fig.4 shows the evaluation of load value prediction with confidence mechanism. Each bar shows the ratio of correct prediction of high confidence, incorrect prediction of high confidence, correct prediction of low confidence, incorrect prediction of low confidence, and no prediction from the bottom. It shows that the average prediction rate is 40%, and average prediction accuracy is more than 85%.

By comparing Fig.3 and Fig,4, it can be seen that the confidence mechanism works well to reduce the ratio of miss prediction by preventing the miss prediction by low confidence situation. However, Fig,4 shows that the ratio of miss prediction of high confidence is about 10%, which is not enough low.

Now we consider the performance of load value prediction. The following parameters are used. F_{load} shows frequency of load instruction, R_{pred} shows ratio of load value prediction, A_{pred} shows accuracy of load value prediction, $reduce$ shows average decrease of latency at load instruction Decrease of CPI by load value prediction is represented as the following expression.

$$\Delta CPI_{pred}$$

$$= F_{load} \times R_{pred} \times A_{pred} \times reduce$$

Assume that $(1 - A_{pred})$ shows prediction miss rate, and $penalty$ shows prediction miss penalty. Increase of execution cycle $\Delta CPI_{mispred}$ is represented as the following expression.

$$\Delta CPI_{mispred}$$

$$= F_{load} \times R_{pred} \times (1 - A_{pred}) \times penalty$$

$CPUtime_{ideal}$ shows the execution time without load value prediction. $CPUtime_{real}$ shows the exeution time with load value prediction.

Ratio of preformance is represented as the following expression.

$$\frac{CPUtime_{ideal}}{CPUtime_{real}} = \frac{CPI_{ideal}}{CPU_{ideal} - \Delta CPI_{pred} + \Delta CPI_{mispred}}$$

Here, we consider the performance of load value prediction by using parameter values of experiment. Parameter values are defined as follows.

- width of instruction issue : 4,
- decrease of latency by load value prediction : 3 cycles,
- load value pediction miss penalty : 10 cycles,
- fequency of load instruction : 25%
- ratio of load value prediction : 35%
- accuracy of load value prediction : 85%
- miss of load vaue prediction : 15%
- $CPUtime_{ideal}$: 0.25

The ratio of performance is

$$\frac{CPUtime_{ideal}}{CPUtime_{real}}$$

$$= \frac{0.25}{0.25 - 0.25 \times 0.35 \times 0.85 \times 3 + 0.25 \times 0.35 \times 0.15 \times 10}$$

$$= 1.58$$

Hence, ratio of performance enhancement is 58%. It is shown that performance increases by decreasing $CPI_{mispred}$. $CPI_{mispred}$ can be decreased by increase A_{pred} and R_{pred}. This paper aims to improve A_{pred} and R_{pred}.

IV. MISS PREDICTION BIAS

In the case of branch prediction, we have found that prediction miss is biased to a small set of instructions[5]. We think that the same phenomenon might be seen in the case of load value prediction. Hence, we study the bias of miss prediction on the load value prediction.

Bias of miss prediction is investigated. Fig.5 shows the ratio of prediction miss caused by miss biased instructions to the total miss predictions. The result shows that about 40% of miss is occupied by the top-16 instructions. The result

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

shows that about 80% of miss is occupied by the top-32 instructions.

Especially, bzip and mcf has a strong miss prediction bias on the small set of instructions.

TABLE I
PROCESSOR CONFIGURATION

Pipeline	5 stages: 1 Fetch, 1 Decode, 1 Execute 1 Memory Access, 1 Commit
Fetch, Decode, Dispatch	4 instructions
Issue	Int: 4, fp: 2, mem: 2
Window	Dispatch queue: 256, Issue queue: 256,
BTB	2K-entry 4-way associative BTB, 32-entry RAS
Memory	64KB, 4-way associative, 1-cycle instruction and date caches 2MB, 8-way associative, 10-cycle L2

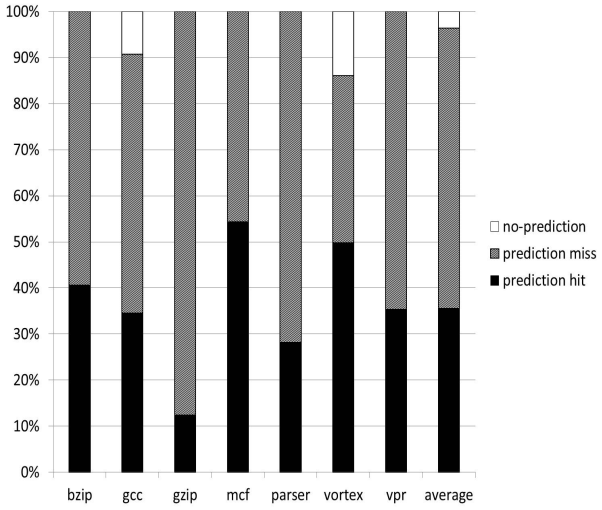

Fig. 3. result of last value prediction without confidence

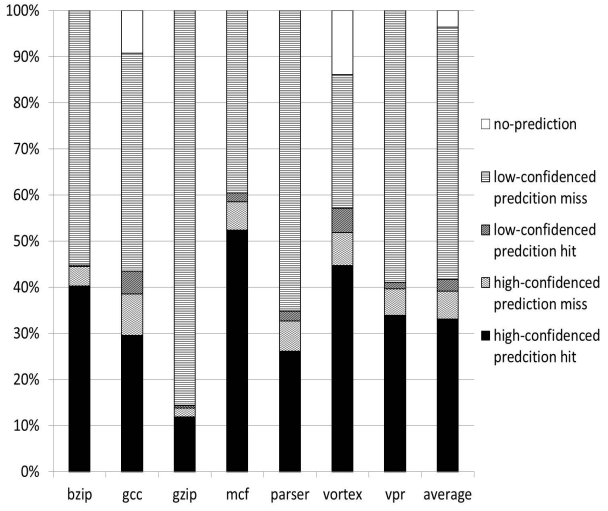

Fig. 4. result of last value prediction with confidence

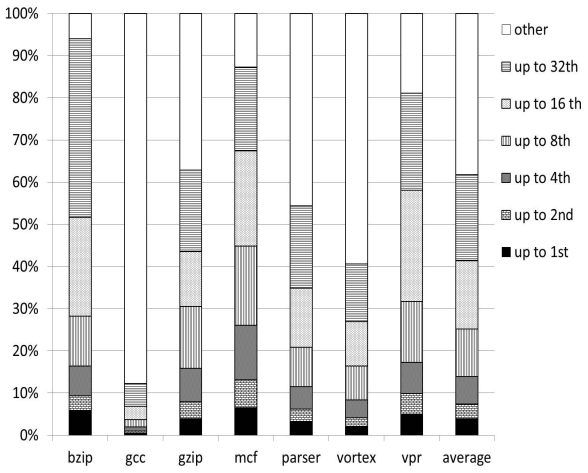

Fig. 5. Ratio of prediction miss on biased 32 instructions

V. PATTERN OF VALUES ON MISS BIAS INSTRUCTIONS

In the precious section, we have seen the existence of prediction miss bias[5]. This section analyzes the detailed pattern of the value on the miss biased instructions. The patterns are extracted by the simulation on the SimpleScalar Tool Set. The following four patterns are extracted.

- Counter type: the pattern of values with the same stride. The example is the sequence of 598, 599, 600, 601, and 602 with stride value 1. This pattern cannot be predicted by the last value prediction.
- Iteration of two values: The example is 272, 368, 272, and 368. This pattern can be predicted by remembering two values.
- Iteration of multiple occurrences of two values: The example is 231, 231, 0, 0, 233, 233, 4, and 4. It is difficult to predict the first occurence of each value, but it is easy to predict the second occurence.
- Iteration of long sequence of values: of values. It is difficult to predict this pattern because of remembering a long sequence of values.

Table II shows the pattern of values on miss bias instructions. And Table III shows the ratio of miss pattern on top-8 biased instructions of bzip benchmark. Table III shows that three patterns occupy most of prediction miss on top-8 biased instructions.

VI. LOAD VALUE PREDICTOR TO UTILIZE PREDICTION MISS BIAS

In section 4 and 5, we have shown that load value prediction miss is biased to a set of instructions. In this section, we propose an innovative load value predictor for a set of miss biased instructions attached to the baseline predictor.

Fig.6 shows the block diagram of the proposed hardware mechanism. It consists of ELVP (Extended Load Value Predictor) and MBB (Miss Bias Buffer). ELVP predicts load value based on the last value, and detects miss biased load instructions. When ELVP detects miss biased load instructions, MBB takes over the load value prediction of the miss biased instructions by local history. The detailed behavior of this mechanism is explained below.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE II
THE EXAMPLE OF FREQUENCE MISS PATTERN

counter type	598-599-600-601-602-...
Iteration of two values:	272-368-272-368-272-368...
Iteration of multiple occurrences of two values	231-231-0-0-233-233-4-4...

TABLE III
MISS PATTERN ON BZIP BENCHMARK

up to 8th(addr)	pattern			
	counter type	Iteration of two values	Iteration of multiple occurrences of two values	other
1st(4270584)		100%		
2nd(4270256)			100%	
3rd(4254408)	100%			
4th(4270424)			100%	
5th(4255400)	100%			
6th(4270496)			100%	
7th(4270560)			100%	
8th(4255496)				100%

A. Detection of miss biased instruction by ELVP

At first, ELVP works as the last value predictor by using tag and last value stored in the table. ELVP also counts the miss prediction and the value is stored into MCT(Miss Counter). MCT is incremented if the prediction is missed, and MCT is decremented if the prediction is hit. When miss count of MCT exceeds the threshold, then its tag is transferred into MBB.

B. Registration of miss biased instruction to MBB

MBB stores information of miss biased instruction, and takes over the load value prediction of miss biased instructions by using the stored information. The stored information is as follows.

- Addr: address of the miss biased instruction
- V1∼V4: history of the load value (last four values).It is organized by shift register.
- Pb (pattern Bit): type of patterns. It consists of 2-bit to represent four patterns.
- CCT (Correct Counter): saturating counter that holds history of prediction. It is incremented on prediction hit, and decremented on prediction miss. CCT is used to determine the effectiveness of prediction by MBB.

In the previous section, the required size of MBB is 16 or 32 in the benchmark programs. In order to use the small size of MBB effectively, MBB is replaced during execution based on LRU (Least Recently Used)[7] logic.

When miss count of ELVP exceeds the threshold, the instruction address is checked whether it already exists in MBB. If the address is not in MBB, the entry of MBB is determined by LRU logic. Namely, the least recently used MBB entry is replaced by the instruction.

C. Update of MBB

When a load instruction which is registered in MBB is committed, the corresponding entry of MBB is updated. V1, V2, V3, and V4 are shifted, and new load value is stored into V1. Pb is updated by comparing V1, V2, V3, and V4. CCT is updated based on the result of prediction.

D. Load Value Prediction by MBB

MBB stores history of miss biased load instructions. When a load instruction is fetched, the instruction address is matched with both ELVP and MBB. ELVP works as a last value predictor. If the instruction address is matched with MBB, then MBB calculates predicted load value as follows.

- If Pb = 1, MBB recognizes that it is an iterative pattern. V1 is used as a predicted value.
- If Pb = 2, MBB recognizes that it is a counter pattern. The predicted value is calculated by V4 + (V3 - V2) ,where V3-V2 is a stride.
- If Pb = 3, MBB recognizes that it is a iteration of multiple occurrences of two values. Prediction values is calculated once by twice iteration.
- If Pb = 0, MBB does not work.

The predicted value is sent to selector, and the final prediction value is selected between the result of baseline predictor and that of MBB. CCT is used to determine the confidence of MBB prediction.

VII. EVALUATION

The proposed method is compared with conventional method on SimpleScalar Tool Set. Last value predictor is used as the conventional method, and three types of MBB organization by proposed method of 8 , 16 , and 32 entries are evaluated.

Fig.7 shows prediction rate, and Fig.8 shows prediction accuracy. From Fig.7, the proposed method can increase prediction rate for all MBB organizations. The better prediction rate is obtained by the bigger MBB organizations. In bzip and mcf, more than 20% prediction rate is increased by the 32 entry. In gcc, prediction rate is not increased by proposed method.

From Fig.8, the proposed method can increase predction accuracy. The better prediction accuracy is obtained by the bigger MBB organizations. Prediction accuracy is increased for all benchmarks, but the a mount of increase is small.

Table IV shows average value of prediction ratio and accuracy for conventional method and proposed method. From Table IV, the proposed method can improve both prediction rate and prediction accuracy. From Table IV, 13% prediction rate in average is increased by the proposed

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

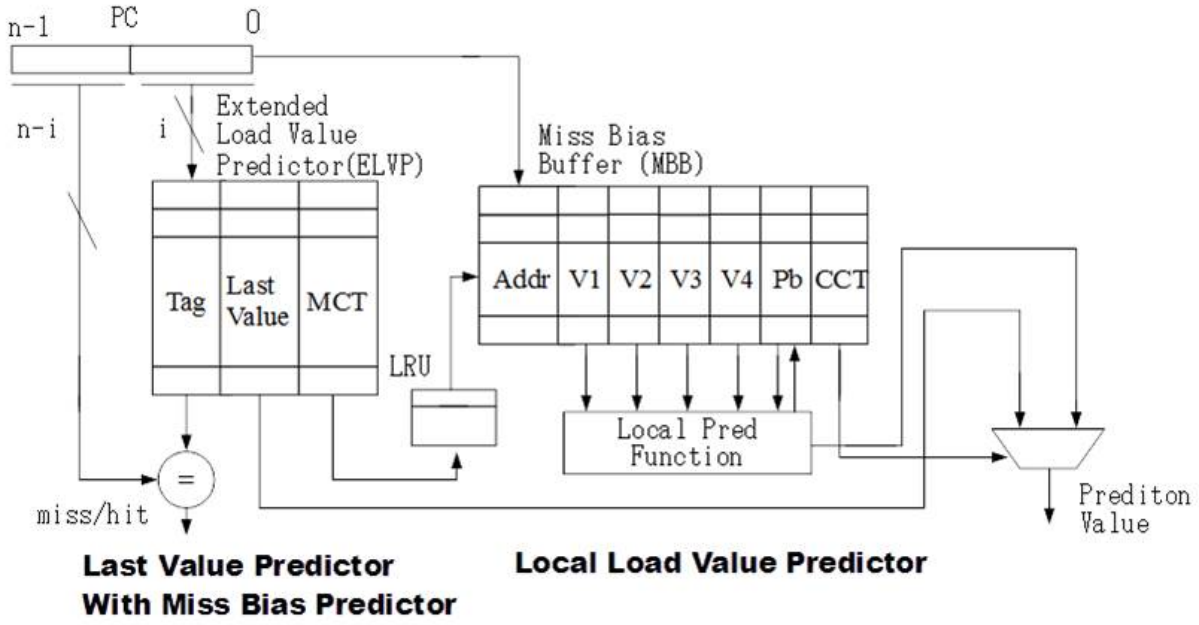

Fig. 6. Block diagram of proposed load value predictor

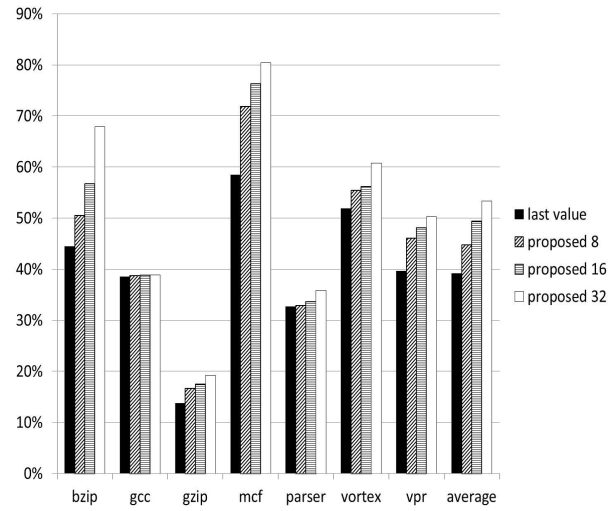

Fig. 7. prediction rate

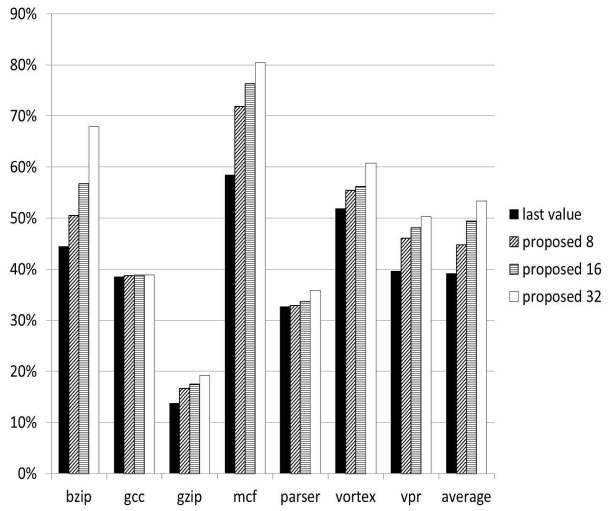

Fig. 8. prediction accuracy

TABLE IV
RESULT OF EXPERIMENT

	prediction rate	prediction accuracy
last value prediction	40%	84%
proposed method(up to 8entry)	44%	86%
proposed method(up to 16entry)	49%	87%
proposed method(up to 32entry)	53%	88%

method. 4% prediction accuracy in average is increased by the proposed method.

VIII. CONCLUSION

Decreasing miss prediction is important for enhancing processor performance of instruction level parallelism. This paper proposed a new mechanism for load value prediction that utilizes the miss prediction bias. The proposed method provides dedicated mechanism for predicting miss biased instructions.

The proposed method is organized by adding a new mechanism to the last value prediction. The proposed method equips small size of buffer for miss biased instructions, which is dynamically replaced. New mechanism provides a dedicated mechanism for three patterns which are frequently appeared.

By experiment, prediction rate and prediction accuracy can be improved. 13% prediction rate in average is increased by the proposed method, and 4% prediction accuracy in average is increased by the proposed method. In order to further improve the prediction rate ,more pattern of values on miss bias instructions must be found.

REFERENCES

[1] Mikko H. Lipasti, Christopher B. Wilkerson, and John Paul Shen, ”Value Locality and Load Value Prediction”, the Proceedings of the 7th International Conference on Architectural Support for Programming Languages and Operating Systems, pp.138-147, 1996.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[2] Martin Burtscher and Benjamin G.Zorn, ”Exploring last n value prediction”, the International Conference on Parallel Architectures and Compilation Techniques, pp.66-76, 1999.
[3] Yiannakis Sazeides and James E. Smith, ”The predictability of data values”, In the 30th International Symposium on Microarchitecture, pp.248-258, 1997.
[4] Bart Goeman, Hans Vandierendonck and Koen De Bosschere, ”Differential FCM: Increasing Value Prediction Accuracy by Improving Table Usage Efficiency”, the Seventh international Symposium on High Performance Computer Architecture (HPCA ’01), pp.207-216, 2001.
[5] L.Meng, K.Yamazaki, and S.Oyanagi,“ A Novel Branch Predictor Using Local History for Miss-Prediction Bias ”, Proceedings of 2012 International Conference on Computer Design (CDES’12), pp.77-83, Jul.2012.
[6] D. Burger and T. M. Austin, ”The SimpleScalar Tool Set Version2.0”, Technical Report, University of Wisconsin-Madison Computer Sciences 1997.
[7] C.C.Kavar and S.S.Paramar, “ Performance Analysis of LRU Page Replacement Algorithm with Reference to different Data Structure ”, International Journal of Engineering Research and Application, Vol.3, No.1, pp.2070-2076, 2013.

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)