UDC 004.254

COMPARISON OF NEHALEM AND IVY BRIDGE TLB ARCHITECTURES

M. ASKARI[1], N.N. IVANOV[2]

Belarusian State University of Informatics and Radioelectronics
6, P. Brovki Str., Minsk, Republic of Belarus
[1]mahask2004@yahoo.com; [2]ivanovnn@gmail.com

The speed gap between processor and main memory is reduced using a cache. Cache role in multi-core parallel systems is detected here. Cache memories without Translation Lookaside Buffer (TLB), will not show their capabilities and their impacts on performance will be lower. The performance comparison of these buffers in the different architectures helps to discover impact of architectures change on their work. In this paper, two different types of Intel microarchitecture have been compared by miss ration of their TLBs. Nehalem and Ivy–Bridge. For this purpose, the statistical techniques used to compare the measured values by Intel–Vtune2013 on SPEC CPU2000 benchmarks.

Keywords: Nehalem, Ivy–Bridge, Miss Rate, Translation Lookaside Buffer, TLB, Spec CPU2000, Vtune2013.

The developers of the system are trying to increase system performance. To support this goal, the Performance Monitoring Unit (PMU) offers a wide range of events to help investigate the processor components. Observe the counted events require special software and understanding presented numbers need statistical analysis. In addition, it is necessary to run specific and standard applications in order to compare the two systems together. For the calculation of Miss Rates, counted event is divided by the total of events which is called INST_RETIRED.ANY. Event containing misses of data TLB (DTLB) is DTLB_MISSES.ANY and event containing hits of Second level of TLB (STLB) is DTLB_LOAD_MISSES.STLB_HIT. Miss rate in this case is obtained by subtracting the Hit rate of value 1.

SPEC company, has introduced some program that can be used as standard applications to calculating performance and miss ratios of TLB, caches and etc. In this paper, it has been used all 12 benchmarks of CINT2000 package included: BZIP2, CRAFTY, EON, GAP, GCC, GZIP, MCF, PARSER, PERLBMK, TWOLF and VORTEX. Each benchmark is executed 50 times in 3 phases. A total of 1800 times randomly run on each system. All experiments have performed on 2 user mode and a copy of the each benchmark is executed in parallel on each core.

The first system under test with architecture Nehalem is Intel Core i5-460M with 2 cores. It has a two-level TLB for each core. On the first level it is an Instruction TLB (ITLB) and a Data TLB (DTLB). On the second level it is a TLB (STLB) unified Instruction and Data. Posted Virtual Address (VA) by core, according to its type is converted to a Physical Address (PA) in the ITLB or DTLB. If a match not happened in level one, it is required to search in level two of TLB and if the address is not found here, it is need to refer main memory that it has a significant time to update. If at any stage it is happened a match, it will be referred to a hierarchy cache and main memory to obtain data relating to physical address conversion. This system has three levels of caches which are called: L1, L2 and L3. The first level (L1) has two parts for Instruction and Data included: L1I and L1D. Each TLB in each level have some parts related to page size included 4Kbyte size, Large and Huge size. In this table, not paying attention to these cases and other cases including prefetching and nature of bench-

marks for discussion and only it want compare the results of two factors together: DTLB and STLB miss rates. ITLB is not counted by event counter in these experiments.

The second system under test with architecture Ivy-Bridge is Intel Core i5-3317U with 2 cores. Assuming the lack of detail, the two systems will have similar model for study their TLBs. The results have been statistically analyzed. Techniques used for the analysis is T-test method. A null hypothesis assumes that there is no significant relationship between the values of experiments. Then the result of statistical analysis is compared with a value considered. If statistical analysis was performed, have a result less than that assumed value, the null hypothesis is rejected. Usually, the assumed value of 0.05 is chosen. In case of rejection of the null hypothesis, say 95% chance of a significant relationship is between the values.

Table 1 shows the results of the analysis for DTLB and STLB by Regression. “R2” is acronym for Standard deviation, “Ttest” for T-test coefficient and “Bnch” for Benchmark.

Table 1. Result of analysis for DTLB and STLB in 460M and 3317U.

	R2	Ttest	R2	Ttest	R2	Ttest	R2	Ttest
Bnch	BZIP2		CRAFTY		EON		GAP	
460M	0.32718	3.28e-05	0.38111	3.80e-06	0.00274	0.72366	0.04302	0.15719
3317U	0.00698	0.58072	0.20229	0.00134	0.15856	0.00556	0.08589	0.04321
Bnch	GCC		GZIP		MCF		PARSER	
460M	0.35563	1.22e-05	0.00669	0.58875	0.31324	3.53e-05	0.19757	0.00196
3317U	0.14059	0.00940	0.04669	0.13588	0.07384	0.05892	0.81317	9.67e-19
Bnch	PERLBMK		TWOLF		VORTEX		VPR	
460M	0.00222	0.74772	3.05e-05	0.97024	0.13154	0.01324	0.24352	0.00042
3317U	0.11244	0.01981	0.58495	6.10e-10	0.28944	8.02e-05	0.76267	5.71e-16

Considering that, each miss on DTLB is an input for STLB so it is assumed that STLB miss is dependent to miss on DTLB. In this regard, the null hypothesis is stated as STLB miss is not dependent to miss on DTLB. Looking at Table 1, where the value in “Ttest” column is less than 0.05 then null hypotheses in this benchmark is rejected and assumption is acceptable with 95 %.

In relation to the i5-460M, the null hypothesis for EON, GAP, GZIP, PERLBMK and TWOLF is acceptable and is not significant correlation between the results. In relation to the i5-3317U, the null hypothesis for BZIP2, GZIP and MCF is acceptable and is not significant correlation between the results. Therefore, according to the architectural details of these two CPUs that was ignored and prefetching behavior and nature of benchmarks like multithreading that were not considered, so it is probability that the algorithms related to above benchmarks be affected by above behavior. As future work, the real effectiveness of the above behavior on these benchmarks can be studied.

References

1. *Tanenbaum, A.* Modern Operating Systems / A. Tanenbaum. // New Jersey: Prentice Hall, 2001.

2. *Hennessy, J.* Computer Architecture / J. Hennessy, D. Patterson // Philadelphia: Elsevier, 2011.

3. *Kandiraju, B. K.* Characterizing the dTLB Behavior of SPEC CPU2000 Benchmarks. / B. K. Kandiraju, A. Sivasubramaniam. // international conference on Measurement and modeling of computer systems. –2002. –Vol.30, p. 129–139.