Analysis and Characterization of Ultra Low Power Branch Predictors

Athanasios Chatzidimitriou, George Papadimitriou, Dimitris Gizopoulos
Dept. of Informatics and Telecommunications
University of Athens
Athens, Greece
{achatz, georgepap, dgizop}@di.uoa.gr

Shrikanth Ganapathy, John Kalamatianos
AMD Research
Advanced Micro Devices, Inc.
{Santa Clara CA, Boxborough MA}, USA
{shrikanth.ganapathy, john.kalamatianos}@amd.com

Abstract—**Branch predictors are widely used to boost the performance of microprocessors. However, this comes at the expense of power because accurate branch prediction requires simultaneous access to several large tables on every fetch. Consumed power can be drastically reduced by operating the predictor under sub-nomimal voltage levels (undervolting) using a separate voltage domain. Faulty behavior resulting from undervolting the predictor arrays impacts performance due to additional mispredictions but does not compromise system reliability or functional correctness. In this work, we explore how two well established branch predictors (Tournament and L-Tage) behave when aggressively undervolted below minimum fault-free supply voltage (V_{min}). Our results based on fault injection and performance simulations show that both predictors significantly reduce their power consumption by more than 63% and can deliver a peak 6.4% energy savings in the overall system, without observable performance degradation. However, energy consumption can increase for both predictors due to extra mispredictions, if undervolting becomes too aggressive.[1]**

Keywords—Branch predictors, energy efficiency, gem5, microarchitectural simulation, power, voltage scaling

I. Introduction

As modern microprocessors require high performance, sophisticated branch predictors are used to deliver high fetch bandwidth and therefore better IPC. However, accurate branch predictors employ large SRAM tables that are being accessed on every fetch cycle [21], leading to high power consumption. This property is undesirable in processors operating under limited power budgets such as those used in embedded systems. The most power consuming components of a branch predictor are the branch target buffer (BTB) and branch direction prediction arrays [23], [24], [25]. Recent work has estimated that the front end of a superscalar processor contributes up to 33% of the overall processor power [23]. In order to improve the chip's energy efficiency, we decide to undervolt the branch predictors, assuming a voltage domain dedicated to the predictor circuits.

Scaling down the supply voltage (V_{DD}) can imply significant improvements in energy efficiency, given that voltage has a quadratic and linear relationship with dynamic power and frequency, respectively. However, as supply voltage is reduced, many circuits, including SRAMS begin to fail. Given that branch predictor operation does not affect functional correctness, processor reliability is not compromised due to faults in the predictor tables. Malfunctions in these units can only compromise performance which provides an opportunity for significant energy savings.

In this work, we make the following contributions: (a) improve energy efficiency of embedded processors by undervolting their branch direction predictors while operating at peak operating frequency (1.0GHz), (b) evaluate the performance impact and energy savings of undervolting using fault rates obtained from silicon measurements in a 14nm FinFET technology and simulating an out-of-order high performance embedded processor, (c) compare the tolerance of two well established branch predictors to undervolting faults. Our results show that the power consumption of a BP can be reduced more than 63% without noticeably compromising performance. L-Tage branch predictor proves to be more tolerant in undervolting. Peak energy savings for a processor with L-Tage is 6.4% when V_{DD} is set to 0.55 V_{nom} (nominal) and 6.3% for Tournament at 0.575 V_{nom} but starts dropping beyond those points.

II. Background & Related Work

Branch predictors are components that can deliver large performance boost to a processor. To increase their efficiency, designers invest significant amounts of chip area to build large and sophisticated predictors. As a result, the actual predictors end up consuming around 10% of the overall chip power [25], [26]. This significant amount of power has attracted a lot of attention, with many studies focusing on reducing BP power consumption [23], [24], [27], [28]. There are many studies that focus on the effects and malfunctions of reduced supply voltage beyond nominal conditions in SRAM arrays of the microprocessor. [1], [2], [3], explore the behavior of several undervolted systems, focusing on cache failures as well as voltage margins of x86 and ARM systems, while the authors of [4] and [5] propose techniques for accelerated margin identification and methodologies for voltage margin prediction.

Moreover, the studies [6], [7], [8], and [12] focus on protection methodologies that allow aggressive undervolting of caches. Agarwal *et al.* [13] focused on yield improvements tolerating process variations. Ganapathy *et al.* [14] presented experimental results for near-threshold voltage operation of SRAM arrays using measurements taken from several wafers at 14nm FinFET process. Zimmer *et al.* [15] proposed a method to analyze the effects of different organization, and timing on V_{min} at design time. Apart from studies that focus on the effects of undervolting in SRAM-based caches, there are also works that evaluate the performance impact of speculative

[1] This work is supported by the H2020 Framework Program of the European Union through the UniServer Project under Grant Agreement 688540.

components. Hsieh *et al.* in [9] evaluate the importance of identifying hard faults that lead to performance degradation for yield improvement. Foutris *et al.* in [10] and [11] and Filippou *et al.* in [16] evaluate the performance impact of permanent faults in several performance components.

III. Methodology

A. Microarchitecture-level modeling

In this study, we use Gem5 [17], a full-system microarchitecture-level simulator to model a system with an independent voltage domain dedicated to the branch predictors. Gem5 offers an out-of-order CPU core with detailed description of several branch predictors. The component's logic is implemented functionally while the storage elements are modeled as variables (memory bits). In this abstraction level, SRAM failures can be modeled as stuck-at faults (SRAM bitcells that are permanently stuck at 0 or 1, regardless of the written value) with equivalent functional result.

GeFIN framework [18] was used to inject stuck-at faults in the simulation for the different voltage levels. The framework can effectively inject multiple long-duration faults on all branch predictor arrays, allowing us to observe the impact of faults inside the branch predictor. These faults correspond to faulty cells due to the voltage reduction of the branch predictors below V_{min}. McPAT tool [19] was then used to collect power consumption measurements for our models.

B. Branch Predictors & Workloads

The studied branch predictors in this work are the Tournament branch predictor [20] and the L-Tage branch predictor [21] which are embedded in the fetch pipeline stage of an out-of-order CPU core. We allocate a budget of approximately 256K bits to be used in the branch predictors of our simulated system in order to support a high performance embedded processor. The organization of L-Tage tables has been selected based on the findings in [21].

For the experimental workloads, we used 20 benchmarks from the MiBench suite [22]. The benchmarks are representative programs from the embedded domain and every benchmark is executed till completion, hence they are applicable for our study as they allow experimentation on all execution phases. On every simulation, a warm-up period of 4M instructions was used during which, the predictors were considered to operate in off-nominal voltage conditions and their stuck-at faults were present.

Table I: Number of faulty cells per predictor across different voltage levels.

Voltage	Tournament	L-Tage
0.600 V_{nom}	1.2	1.5
0.575 V_{nom}	113	113
0.550 V_{nom}	5,826	5,775
0.525 V_{nom}	81,495	80,865
0.500 V_{nom}	159,382	169,094

C. Voltage levels & SRAM maps

Undervolted operation affects the switching time of transistors and is usually accompanied with underclocking to cover this mismatch. In this study, the predictors' arrays operate in the same frequency as the embedded processor (1.0 GHz), which remains unchanged. Using the compound Poisson distribution that was proposed in [14], we have developed a tool that generates maps of SRAMs, defining the operating threshold of each cell. Each cell is operational if the supply voltage (V_{DD}) is above its threshold and is not operational if the V_{DD} is set below the threshold (modeled as stuck-at faults). It is important to note that the generated fault maps are directly derived from 14 nm silicon measurements and are the closest representation to manufactured SRAM arrays.

The granularity of voltage levels is 0.025 V_{nom} (nominal voltage). For every SRAM array, we generated 10,000 SRAM maps and randomly selected 50 of each, representing 50 different "test chips" for our experiments. The generated maps have a yield that matches what is presented in [14] for each voltage level. In every one of these chips, we executed the same workloads in all 6 different operating voltages (including V_{nom}). Table I presents the average number of stuck-at faults that were present at each voltage level for the two predictors.

IV. Experimental Results

A. Prediction Accuracy

As the number of faulty cells increases, the accuracy of both predictors drops. Fig. 1 presents the average (across all the test chips) misprediction rates for each benchmark. In all cases, L-Tage misprediction rate never exceeds 25 *Miss-Predictions per Kilo Instuctions* (MPKI), while the tournament predictor reports a rate as high as 50 MPKI on *dijkstra* benchmark. Considering that the fault-free misprediction rate is less than 2,

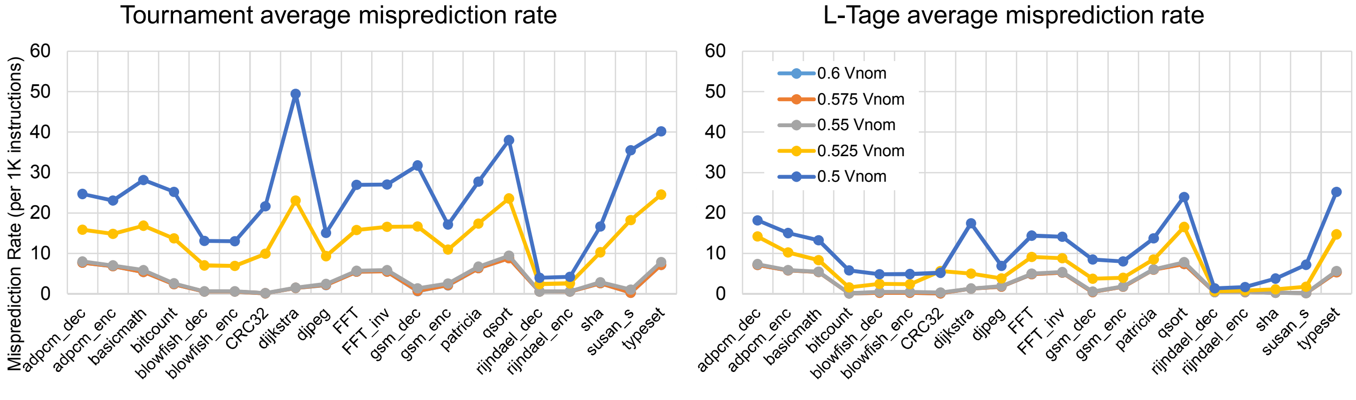

Fig. 1: Misprediction rates (per 1K instructions) for Tournament (left) and L-Tage (right).

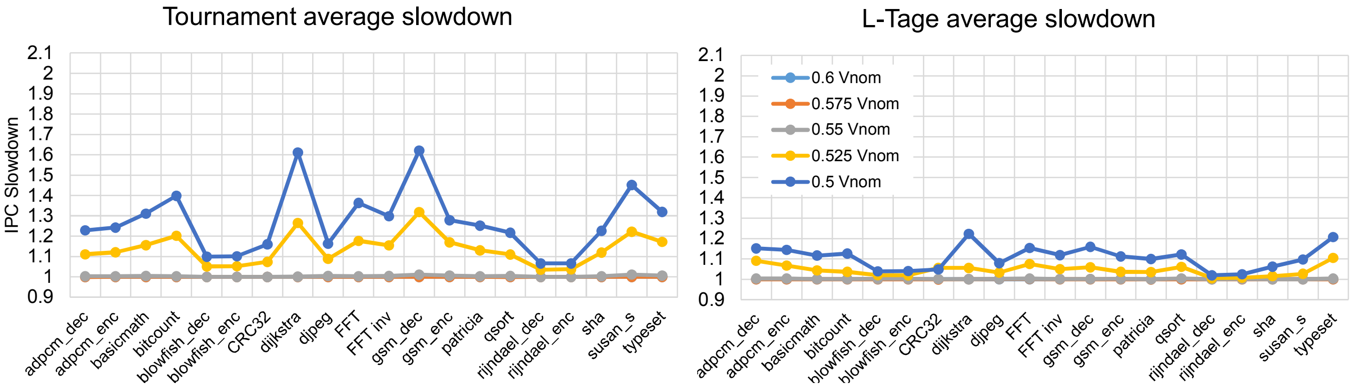

Fig. 2: Average IPC slowdown (normalized) for Tournament (left) and L-Tage (right).

a difference of 20x is reported, which is expected to deliver significant performance degradation on the system. The increased misprediction rate can lead to poor performance and negatively affect the energy consumption of the chip.

B. Performance

Both predictors appear to suffer incrementally as the supply voltage is reduced and more faulty cells are present in the SRAM arrays. However, the level of performance degradation is significantly smaller than the one of the accuracy loss. Both predictors are hybrid and dynamically select the source table to provide a prediction. In practice, this means that they can adapt and avoid using faulty cells for prediction. In order to cause observable impact on performance, multiple failures must occur (*faulty sets*), forcing successive decisions based on faulty elements. As the size of these arrays is quite large, and the predictor entries are usually selected using different criteria, faulty sets are unlikely to happen with small numbers of faulty cells. This can be observed in Fig. 2, where the average IPC differences for the Tournament predictor are small (less than 1%) for the first two levels of undervolting, but quite large at lower voltages (average 27% IPC loss at 0.5 V_{nom}). In contrast, the L-Tage predictor exhibits higher tolerance and the average IPC loss at 0.5 V_{nom} is 10%. The worst slowdown is observed in benchmark *dijkstra*, which reports an average slowdown of 2.01X on the Tournament predictor and 1.22X on the L-Tage. The higher tolerance of L-Tage can be attributed to the fact that the predictor employs 12 different tables for a prediction and the largest portion of its arrays is used as tag. As a result, the probability of having a fault-free tag-hit in one of the tables is higher than that of Tournament. We can see that the predictor can still deliver high performance even under very low voltage levels, as the average slowdown is 4.5% for the 0.525 V_{nom} while for the Tournament predictor, the slowdown is 13.85%.

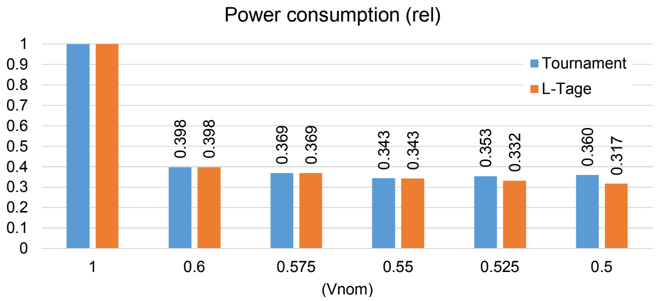

Fig. 3: Branch predictor power consumption on different voltage levels.

C. Power & Energy

To measure the power consumption of each predictor, we used McPAT [19]. The tool was manually modified to use the lower V_{DD} for the undervolting configurations. To calculate the power consumption, McPAT uses the array sizes and event triggers (from Gem5 simulations) for read and write operations on the arrays. Fig. 3 shows how the power consumption is affected in the two predictors on the different voltage levels. In lower voltage levels we see that the two predictors achieve different power savings. Tournament predictor's power consumption lowers as the voltage drops until 0.55 V_{nom}. Interestingly, the high misprediction rate causes higher consumption in lower V_{DD}. In contrast, L-Tage predictor power consumption drops along with the supply voltage.

By considering both the power consumption and the different execution time due to the low voltage operation, we have calculated the total energy consumption of the CPU for the execution of the benchmarks. Only the branch predictors (BP) were configured to operate in off-nominal voltage levels (independent voltage domain) and Fig. 4 illustrates how the overall energy consumption is affected by the undervolted branch predictors. For the energy estimation we assume that that the BP power consumption is 10% of the overall CPU power, when operated at nominal voltage [25],[26],[28]. The presented values are calculated as the average energy consumption for all of our benchmarks and test chips, for each voltage level.

Tournament achieves the best energy savings (6.3%) when operated at 0.575 V_{nom} voltage level, while L-Tage manages to deliver 6.4% at 0.55 V_{nom}. At 0.575 V_{nom}, the predictors have approximately 100 faulty cells, which lead to marginal average performance losses (in the scale of 0.1%). Similar level of efficiency is also observed at 0.55 V_{nom}. Both predictors still provide high performance (<1% degradation) at 0.55 V_{nom}, where more than 5,000 faulty cells exist in the predictor arrays. This voltage level marks the point where energy consumption begins to worsen compared to the nominal operation for the Tournament predictor, while L-Tage continues to deliver higher energy efficiency even at 0.525 V_{nom}. The overall chip's energy consumption at 0.5 V_{nom} with the 10.7% slowdown of the L-Tage is 3.1% more than the nominal consumption and the benefits of the undervolting are offset by the performance

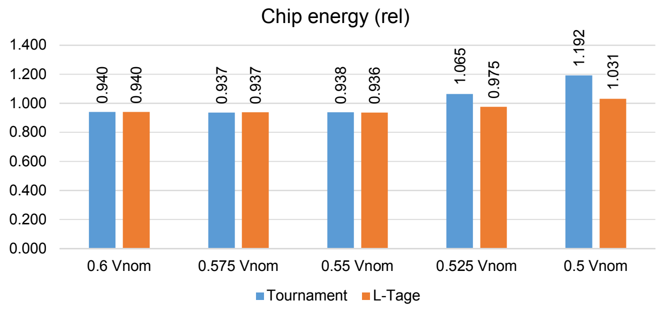

Fig. 4: Average energy consumption for the different levels of undervolting, normalized to nominal energy consumption.

degradation. This is even worse for the Tournament predictor, which consumes almost 19% more energy at that point. In the previous step of 0.525 V_{nom}, L-Tage manages to consume 2.5% less energy than the nominal while the Tournament predictor's massive failing causes 6.5% increase in the energy consumption.

V. Conclusions

We have explored the behavior of two widely used branch predictors, Tournament and L-Tage, when operated in near threshold voltage on a 14nm FinFET fabrication node. Given the availability of a separate voltage domain, aggressive lowering of the V_{DD} can lead to average energy savings up to 6.4% in both cases, without affecting the system reliability or performance. The lowering of supply voltage starts to have detrimental impact in the overall energy consumption when reaching 0.55 V_{nom} or below. Our study shows that the performance/power tradeoff for speculative components of a microprocessor can be exploited to reduce the overall power consumption without compromising the system reliability.

References

[1] A. Bacha and R. Teodorescu, "Dynamic reduction of voltage margins by leveraging on-chip ECC in Itanium II processors," International Symposium on Computer Architecture (ISCA), 2013.

[2] A. Bacha and R. Teodorescu. "Using ECC Feedback to Guide Voltage Speculation in Low-Voltage Processors," IEEE/ACM International Symposium on Microarchitecture (MICRO), 2014.

[3] G. Papadimitriou, M. Kaliorakis, A. Chatzidimitriou, D. Gizopoulos, P. Lawthers, and S. Das, "Harnessing Voltage Margins for Energy Efficiency in Multicore CPUs," International Symposium on Microarchitecture (MICRO), 2017.

[4] M. Kaliorakis, A. Chatzidimitriou, G. Papadimitriou, D. Gizopoulos, "Statistical Analysis of Multicore CPUs Operation in Scaled Voltage Conditions," IEEE Computer Architecture Letters (CAL), 2018.

[5] G. Papadimitriou, A. Chatzidimitriou M. Kaliorakis, Y. Vastakis, D. Gizopoulos, "Micro-Viruses for Fast System-Level Voltage Margins Characterization in Multicore CPUs," International Symposium on Performance Analysis of Systems and Software (ISPASS), 2018.

[6] C. Wilkerson, H. Gao, A. R. Alameldeen, Z. Chishti, M. Khellah, and S.-L. Lu, "Trading off Cache Capacity for Reliability to Enable Low Voltage Operation," International Symposium on Computer Architecture (ISCA), 2008.

[7] Z. Chishti, A. R. Alameldeen, C. Wilkerson, W. Wu, and S.-L. Lu, "Improving cache lifetime reliability at ultra-low voltages," International Symposium on Microarchitecture (MICRO), 2009.

[8] H. Duwe, X. Jian, D. Petrisko, and R. Kumar, "Rescuing uncorrectable fault patterns in on-chip memories through error pattern transformation," International Symposium on Computer Architecture (ISCA), 2016.

[9] T.Y. Hsieh, M.A. Breuer et al., "Tolerance of Performance Degrading Faults for Effective Yield Improvement," International Test Conference (ITC), 2009.

[10] N. Foutris, D. Gizopoulos, J. Kalamatianos, and V. Shridharan, "Assessing the Impact of Hard Faults in Performance Components of Modern Microprocessors," International Conference on Computer Design (ICCD), 2013.

[11] N. Foutris, A. Chatzidimitriou, D. Gizopoulos, J. Kalamatianos, V. Sridharan, "Faults in data prefetchers: Performance degradation and variability," VLSI Test Symposium (VTS), 2016.

[12] J. Abella, J. Carretero, P. Chaparro, X. Vera, A. González, "Low Vccmin fault-tolerant cache with highly predictable performance," International Symposium on Microarchitecture (MICRO), 2009.

[13] A. Agarwal, B. C. Paul, H. Mahmoodi, A. Datta, K. Roy, "A process-tolerant cache architecture for improved yield in nanoscale technologies," IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, Vol.: 13, No.: 1, 2005.

[14] S. Ganapathy, J. Kalamatianos, K. Kasprak, and S. Raasch, "On Characterizing Near-Threshold SRAM Failures in FinFET Technology," Design Automation Conference (DAC), 2017.

[15] B. Zimmer, S. O. Toh, H. Vo, Y. Lee, O. Thomas, K. Asanovic, B. Nikolic, "SRAM Assist Techniques for Operation in a Wide Voltage Range in 28-nm CMOS," IEEE Transactions on Circuits and Systems, Vol.: 59, No.: 12, 2012

[16] F. Filippou, G. Keramidas, M. Mavropoulos, D. Nikolos, "Recovery of Performance Degradation in Defective Branch Target Buffers," International Symposium on On-Line Testing and Robust System Design (IOLTS), 2016.

[17] N. Binkert, B. Beckmann, G. Black, S. K.Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R.Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D.Hill, D. A.Wood, "The Gem5 simulator," ACM SIGARCH Computer Architecture News, vol. 39, no. 2, May, 2011.

[18] A. Chatzidimitriou, D. Gizopoulos, "Anatomy of microarchitecture-level reliability assessment: Throughput and accuracy," International Symposium on Performance Analysis of Systems and Software (ISPASS), 2016.

[19] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen , N. P. Joupp, "McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures," International Symposium on Microarchitecture (MICRO), 2009.

[20] R. E. Kessler, "The Alpha 21264 Microprocessor," IEEE Micro, vol. 19, no. 2, 1999.

[21] A. Seznec, "A 256 kbits l-tage branch predictor," Journal of Instruction-Level Parallelism (JILP) Special Issue: The Second Championship Branch Prediction Competition (CBP-2), vol. 9, 2007.

[22] M. Guthaus, J. Ringenberg, D. Ernst, T. Austin, T. Mudge, R. Brown, "MiBench: A free, commercially representative embedded benchmark suite," Intl. Workshop on Workload Characterization (WWC), 2001.

[23] S.Kim, E.Jo, H.Kim, "Low Power Branch Predictor for Embedded Processors," International Conference on Computer and Information Technology, 2010.

[24] J. Haj-Yihia, A. Yasin, Y. Ben Asher, A. Mendelson, "Fine-Grain Power Breakdown of Modern Out-of-Order Cores and Its Implications on Skylake-Based Systems," ACM Transactions on Architecture and Code Optimization (TACO), Volume 13 Issue 4, 2016.

[25] D. Parikh, K. Skadron, Y. Zhang, M. Barcella, M. Stan, "Power Issues Related to Branch Prediction," International Symposium on High-Performance Computer Architecture (HPCA), 2002.

[26] D.Chaver, L.Pinuel, M.Prieto, F. Tirado, M. C. Huang, "Branch prediction on demand: an energy-efficient solution," International Symposium on Low Power Electronics and Design (ISLPED), 2003.

[27] Y. Chang, "Lazy BTB:reduce BTB energy Consumption using Dynamic Profiling," Design Automation Asia & South Pacific Conference, 2006.

[28] M.C.Huang, D.Chaver, L.Pinuel, M.Prieto, F. Tirado, "Customizing the Branch Predictor to reduce Complexity and Energy Consumption," IEEE Micro Sept/Oct 2003.