Microarchitectures for Managing Chip Revenues under Process Variations

Abhishek Das, Serkan Ozdemir, Gokhan Memik, Joseph Zambreno, and Alok Choudhary
Electrical Engineering and Computer Science Department, Northwestern University

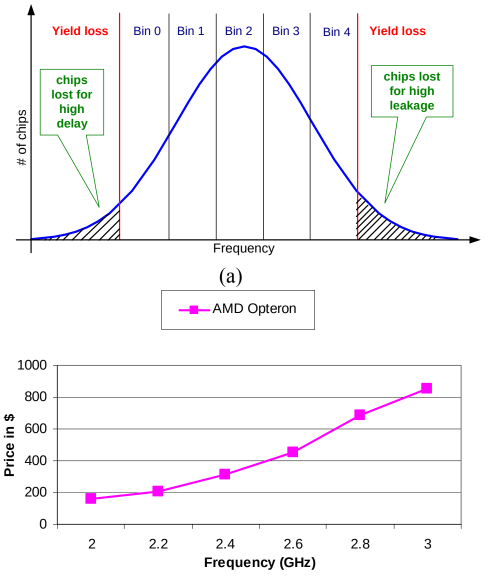
***Abstract*—As transistor feature sizes continue to shrink into the sub-90nm range and beyond, the effects of process variations on critical path delay and chip yields have amplified. A common concept to remedy the effects of variation is speed-binning, by which chips from a single batch are rated by a discrete range of frequencies and sold at different prices. In this paper, we discuss strategies to modify the number of chips in different bins and hence enhance the profits obtained from them. Particularly, we propose a scheme that introduces a small Substitute Cache associated with each cache way to replicate the data elements that will be stored in the high latency lines. Assuming a fixed pricing model, this method increases the revenue by as much as 13.8% without any impact on the performance of the chips.**

***Index Terms*— Computer Architecture, Cache Memories, Process Variations, Fault-tolerant Computing.**

I. Introduction

The computing research literature is filled with design techniques and architectural optimizations that seek to improve performance, power consumption, reliability, and security among others. However, the evaluation of these concepts tends to neglect one of the key factors driving any chip manufacturing decision: a company's bottom-line of revenue and profit. This shortcoming is understandable, as the relationship between any of the standard design metrics and profit is not well understood. For example, an optimization that improves performance by 10% will increase profit only if a) the cost of re-engineering can be amortized over the lifespan of the chip, b) the per-unit testing cost stays constant, and c) consequent changes in other design factors do not decrease the value of the new chip to consumers. In this work, we aim to fill this important gap and show how architectural decisions can be made considering the revenue/profit from a batch of chips.

Chip yield is one obvious scope for optimization, as the continuing downward scaling of transistor feature sizes has made fabrication considerably more difficult and expensive [10]. However, an approach that optimizes solely for yield would not take into account the fact that CPUs concurrently manufactured using a single process are routinely sold at different speed ratings and prices. This practice of speed-binning (Figure 1a) is usually performed by testing each manufactured chip separately over a range of frequency levels until it fails. As a result of the inherent process variations, the different processors fall into separate speed bins, where they are rated and marketed differently. Figure 1b shows this distribution for AMD Opteron family processors [2]. Similar superlinear correlation between frequency and price of a chip can be observed for various processor families by vendors such as Intel, IBM, and Texas Instruments. Speed-binning thus helps chip manufacturer create a complete product line from a single design. Assuming a simplified supply and demand model, total chip revenue would be the sum of the segmented areas under the yield curve. Consequently, one way of increasing revenue would be to shift the binning distribution in such a way that more processors are able to fall into higher-priced bins.



(a)

(b)

Fig. 1 (a) Frequency binning in modern microprocessors, (b) Price vs. frequency of AMD Opteron family [2].

In order to improve the binning of individual chips at the circuit and architectural level, the effects of process variations on fabrication must be masked. Previous studies have shown that in a relatively mature technology like 130nm, these variations are known to result in as much as a 30% decrease in maximum frequency and 500% increase in leakage power [5]. For newer technologies, these variations can be even higher: 20-fold increases in leakage have been reported for 90nm [4]. An expected continuation of this trend will increase the impact of speed-binning on a manufacturer's bottom-line.

In this paper, we propose a scheme in which the level 1 (L1) cache is augmented with a small Substitute Cache (SC) storing the most critical cache words. We concentrate on

Manuscript submitted: 9-May-2007. Manuscript accepted: 01-Jun-2007. Final manuscript received: 13-Jun-2007.

caches because caches are known to be the critical component under process variations [7]. Also, our analysis (described in Section II) reveals that the critical path lies on the cache 58% of the chips we model. With the help of minimal control logic, the processor can fetch data from SC instead of the main data array whenever a read/write access is made to these critical words. Hence access latency is minimized with no extra cache misses. The SC technique registers 12.4% and 13.8% increases in revenue for the average and best cases, respectively. These gains are achieved mostly through an increase in the number of chips in the higher-priced bins.

The rest of this paper is organized as follows. Section II describes the modeling framework. Revenue estimation techniques are discussed in Section III. In Section IV, we present our proposed architecture in detail. Experimental results are presented in Section V. Related work and conclusions are presented in Sections VI and VII.

II. Modeling Framework

A. *Architecture Modeling*

To model a processor core, we have taken into account the 7-stage pipeline in Alpha-21364 (EV7) architecture. The main components of our processor are the Issue Queue, the Integer Execution Unit, the Register File, and the L1 Data cache. All these components are modeled in SPICE using the 45nm BPTM technology models. The issue queue is based on that of EV7 and has 20 entries. The register file is an 80-entry structure with 4 read and 2 write ports. The integer execution unit is modeled using the netlist generated after synthesizing the corresponding component in the Sun OpenSPARC code [12]. The L1 cache is a 32 KB 4-way set associative cache and our model is based on the architecture described by Amrutur and Horowitz [3].

B. *Modeling Process Variations*

Process variations are statistical variations in circuit parameters like gate-oxide thickness, channel length, Random Doping Effects (RDE), due to shrinking process geometries [4]. They can be of two types: with-in-die (WID) and die-to-die (D2D) variations. WID variations consist of spatially correlated or systematic variations, and random variations.

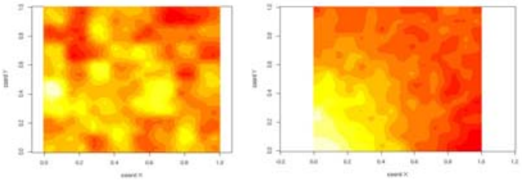


Fig. 2. Maps showing the variation of threshold voltage for different range parameters: φ = 0.3 (left) and φ = 0.5 (right).

We model both spatially correlated and random process variations for five different process parameters (metal thickness, inter-layer dielectric thickness, and line-width on interconnects; gate length and threshold voltage for the MOS devices). The statistical distributions of these parameters are based on limits given by Nassif [10]. To take into account the spatial correlation we use a range factor (φ) in the two dimensional layout of the chip, which gives a measure of randomness in variations. The values in two different points separated by a distance d are correlated by $1/d$ if $d < \varphi$ and are not correlated if $d > \varphi$. Since spatially correlated process variations are found to be the dominating factor [6], we set the random variations to constitute 30% of the total variation. A spatial map of parameter values is generated using the 'R' tool for statistical computing. Example maps with varying φ values are shown in Figure 2. We then use the floorplan of Alpha EV7 processor to extract the parameter values corresponding to the different units we model in this work.

C. *Binning Methodology*

In order to estimate the binning and demonstrate the effect of process variations on it, we chose a set of 1000 chips for our analysis. The delay and leakage current values obtained from SPICE simulations are used in estimating the total delay and leakage power, which in turn is used to determine the binning distribution and yield loss. The cut-off for delay has been set to be the sum of mean and standard deviation of the delay of the simulated chips (μ + σ) and the leakage cut-off has been set to be three times the mean leakage value.

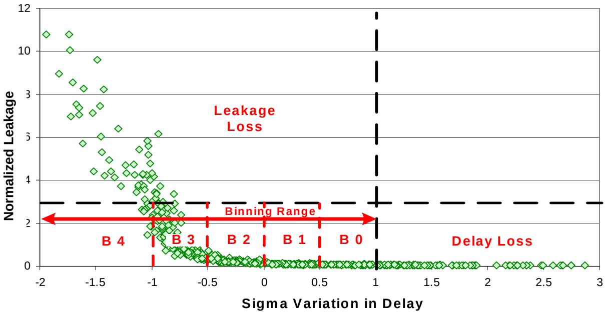


Fig. 3. Normalized leakage and delay distribution scatter plot for simulated chips showing the binning for 5-bin strategy.

Most processor families are available in discrete frequency intervals. Likewise, our binning methodology also assumes equal binning intervals depending on the number of bins to be generated. Chips that satisfy delay and leakage cut-off, contribute to total yield and are placed into discrete bins starting from the slowest to the fastest bin. Figure 3 shows the distribution of the normalized leakage power consumption versus the distribution of chip latencies for the base case (i.e., without any architectural optimizations) for the 1000 simulated chips. The chips are distributed in 5 distinct bins; those falling within 'μ + σ' and 'μ + 0.5σ' delay values are put into Bin0 (denoted by B0). These correspond to the slowest chips. Similarly, chips with latencies within 'μ + 0.5σ' and 'μ' are assigned to Bin1, between 'μ' and 'μ - 0.5σ' to Bin2, and likewise. Note that the highest bin consists of the chips with delay values less than 'μ – σ'. Using a similar methodology, we model a strategy that generates 6 or more bins. In this case, the bin intervals are set to 0.4μ.

III. Revenue estimation Methodology

To devise a simplistic revenue estimation methodology, we analyze the prices of the available chips from representative families and use this information to estimate a change in revenue when the distribution of the chips in the bins is changed. Pricing data was collected from Intel, AMD, and Texas Instruments (TI), for representative families having 5 and 6 speed ratings [2, 8]. The majority of the processors manufactured by the above companies have 5 or 6 distinct frequency ratings matching with our binning methodology.

Instead of using the absolute prices, we use normalized prices within each family. In order to make the model independent of any particular processor family, we use maximum, average, and minimum normalized price values for each of the 5-bin and 6-bin families. To calculate the revenue earned from a bin, we multiply the yield of that bin with its normalized price. The revenues obtained from each of these bins are then summed up to get the total revenue for the whole processor family manufactured.

Note that our cost model is easy to develop, since it based on the current market prices. However, it assumes the existence of previously determined pricing data, and hence cannot be used to estimate revenue for a new processor family before manufacturing and marketing of the same. Moreover, significant economic fluctuations like change in demand and supply may affect this simplified cost model.

IV. Revenue-Aware Architecture

Our goal in this work is to control the critical paths in a set of manufactured set of chips and move the chips to the faster bins to increase the revenue. We achieve this with the aid of a novel scheme called Substitute Cache (SC). At a high level, the idea is to augment each cache way with extra storage that will be used if certain locations in the main cache exhibit long latencies. In such cases, the data will be read from the substitute cache, and chips from the lower frequency bins can now be placed in higher frequency bins, because the high latency lines are not used. Moreover, some of the chips, which could have failed due to high access latencies, will be added to the overall yield.

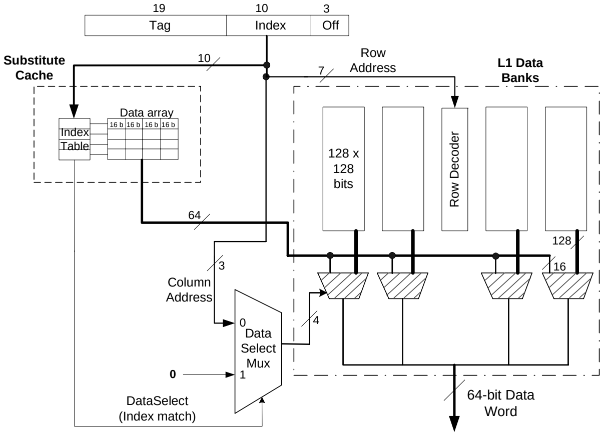


Fig. 4. One cache way of a 32KB 4-way set associative L1 cache augmented with Substitute Cache. Column muxes are shaded as they select data from 9 inputs as opposed to 8 inputs.

The anatomy of the proposed cache architecture is shown in Figure 4. The substitute cache (SC) for a single cache way is highlighted within the dashed block. SC is similar to a fully associative cache structure. In our study, its size is either 4 or 8 entries. As opposed to the L1 cache, SC has smaller line sizes. Particularly, it consists of only 64-bit entries, because it stores words of the main data array. Instead of storing the whole cache line, only the critical word in the line is stored in the SC, because our study reveals that the words with maximum access latency are always the ones that are furthest from the decoder. As a result, by storing these words, we obtain the same improvement in frequency while keeping the SC size small.

An SC is divided into 2 components: an index table and a data array. Whenever a cache word is placed in the data array of the SC, index bits of its address, which is equal to the concatenation of the row and column addresses (10 bits in our architecture) are placed in the index table of the SC. In case of a data access, the index table is checked with the index bits of the address. A match implies that the data will be read from the SC instead of the main array. Specifically, if the index of the address is found in the SC index table, the contents of the corresponding data array row are forwarded to the column multiplexers of the main array. If the index of the address does not match any index table entries, the main array will be accessed. Note that, even if there is a match in the index table, the access can still miss in the cache if the corresponding tag does not match.

During a typical read operation the row address part of the index field selects the appropriate row in the data array through the row decoder. The requested word is then chosen by the column multiplexers with the help of the column address bits of the index. One of the key observations is the difference between the times taken by each of these steps. Particularly, the inputs to the column multiplexers are available at the same time the decoder is accessed. However, the signals provided to the decoders will traverse through the decoder logic, the word lines, the memory cell, the bit lines, and the sense amplifiers before it will reach the column multiplexers. We utilize this imbalance to operate our SC structure. As soon as the address is available, we start accessing the SC index table. If they record a hit, we change the input to the column multiplexers to 0, and forward the output of the SC as the output of the cache. If, on the other hand, there is no match in the index table, we will set the column multiplexer to the original position indicated by the column address. Using CACTI 3.2 we found the total access latency for an 8-entry SC to be 0.24 nanoseconds; whereas the latency for the main array (one set of the 32KB 4-way set associative cache) is 0.40 nanoseconds. Therefore, the SC access can be completely overlapped with the main array access and will not cause an increase in the cache access latency. The only change in the latency of the main array is due to the changes in the column multiplexers. Because of the data forwarding from the SC, the column multiplexers (straddled in Figure 4) have an additional input coming from the SC data array. Using SPICE we found this overhead to be

0.34% of the overall cache access latency. Note that there is no performance loss in terms of CPI for the SC scheme, as the effective cache size remains unchanged.

One of the key components during the operation of SC is the index table. After the chip is manufactured, a Built-In-Self-Test (BIST) is performed where n most critical cache indices are chosen and placed in the SC index table. It should be noted that the size of the SC dictates the area and power overhead of this approach. With the help of SPICE and CACTI, we found the total power overhead to be 6.0% and 6.5%, and the area overhead to be 3.7%, and 4.1% of the main array for the 4-, and 8-entry SCs, respectively.

V. Experimental Results

To find how the chips are placed into different bins, we first analyze the base architecture and find the ‘μ’ and ‘σ’ of critical path latencies of the 1000 chips. The bin boundaries are set based on these values. We then apply the SC scheme to find the new latencies and the corresponding bin distribution.

The chips that fall in the higher/faster bins are sold with higher prices than those falling in the lower/slower bins. The number of chips in different bins for the base case (without any resizing) is multiplied with their respective maximum (max), average (avg), and minimum (min) prices to calculate the maximum, average, and minimum revenue for the base case. Using the same methodology, the revenues for different SC schemes are calculated based on their new binning distribution. The relative change in revenue is then calculated with respect to the revenue of the base case.

TABLE I

Binning Strategy	Price Curve	Increase in revenue [%]			
		$\Phi = 0.3$		$\Phi = 0.5$	
		SC-4	SC-8	SC-4	SC-8
5 bins	*Max*	6.99	11.42	6.16	13.01
	Avg	6.57	10.90	6.07	12.32
	Min	5.83	9.71	5.78	11.43
6 bins	*Max*	7.29	13.75	5.83	12.14
	Avg	6.49	12.41	5.61	11.51
	Min	5.59	10.44	5.51	10.39

Table 1 presents the percentage increase in revenue obtained using two different SC schemes. The SC-4 scheme increases the revenue by up to 7.3 % for 6-bin case. Note that SC has power consumption overhead because of the additional structures it uses, and hence causes some power-related chip losses. However, despite this loss, we observe that the SC tends to provide better revenues than base case, because it is able to generate elevated number of chips in higher bins. The SC-8 scheme improves the revenue by up to 13.8%. Generally, we see that the improvements are higher when $\varphi = 0.3$. This is expected because when the process variation parameters are more random, there are few extreme cases and SC can eliminate them, making a significant change on the overall critical path latency.

VI. Related Work

Several circuit-level techniques have been adopted to counter the negative effects of process variations. Datta et al. [5] propose a novel approach of changing the effective speed-binning by gate sizing, and thus increasing the profit. Ozdemir et al. [11] present cache architectures which improve the overall yield of a batch of chips. The approach in this paper is significantly different from [11] since the total chip yield is divided into bins that have unequal weights. Liang et al. [9] target at mitigating the effects of process variations by introducing variable latency structures. Agarwal et al. [1] propose a scheme that prevents yield loss due to failures in the SRAM cells of the cache. In comparison to the abovementioned works, our efforts have been directed towards effective binning and revenue optimization. Besides the novelty of the SC scheme, to the best of our knowledge, this is the first work in revenue/profit-aware architectures.

VII. Conclusion

Efficient binning under process variations is becoming a significant challenge for chip manufacturers. A considerable amount of effort is being made to save chips from excessive delay and market them properly to increase the profit margin. In this paper, we proposed an architecture that aims at maximizing the revenue obtained from a particular line of chips. This technique, called Substitute Cache (SC), has no performance overhead and works by storing critical words of the data array in a separate structure. Overall, the most aggressive SC scheme increases the revenues by up to 13.8%, with no performance overhead.

References

[1] A. Agarwal, B. C. Paul, H. Mahmoodi, A. Datta, and K. Roy, "A Process-Tolerant Cache Architecture for Improved Yield in Nanoscale Technologies," *IEEE Trans. Very Large Scale Integrated Systems*, vol. 13, pp. 27-38, 2005.

[2] AMD, "AMD Processor Pricing", May 2006, http://www.amd.com/pricing

[3] B. S. Amrutur and M. A. Horowitz, "Speed and Power Scaling of SRAM's," *IEEE Trans. on Solid-State Circuits*, vol. 35, pp. 175-185, Feb. 2000.

[4] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter Variations and Impact on Circuits and Microarchitectures," In Proc. of *Design Automation Conference*, 2003.

[5] A. Datta, S. Bhunia, J. H. Choi, S. Mukhopadhyay, and K. Roy, "Speed Binning Aware Design Methodology to Improve Profit Under Parameter Variations," In Proc. of *ASP-DAC*, 2006.

[6] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos, "Modeling Within-Die Spatial Correlation Effects for Process-Design Co-Optimization," In Proc. of *ISQED*, 2005.

[7] E. Humenay, D. Tarjan, and K. Skadron, "Impact of Parameter Variations on Multi-Core Chips," In Proc. of *Workshop on Architectural Support for Gigascale Integration*, June 2006.

[8] Intel, "Intel Processor Pricing", 2006, http://www.intel.com/intel/finance/pricelist/processor_price_list.pdf

[9] X. Liang and D. Brooks, "Mitigating the Impact of Process Variations on CPU Register File and Execution Units," In Proc. of *International Symposium on Microarchitecture*, 2006.

[10] S. R. Nassif, "Modeling and Analysis of Manufacturing Variations," In Proc. of *IEEE Conference on Custom Integrated Circuits*, May 2001.

[11] S. Ozdemir, D. Sinha, G. Memik, J. Adams, and H. Zhou, "Yield-Aware Cache Architectures," In Proc. of *Intl. Symp. on Microarchitecture*, 2006.

[12] Sun, "OpenSPARC T1", http://opensparc-t1.sunsource.net/index.html