Three-Dimensional Cache Design Exploration Using 3DCacti

Yuh-Fang Tsai, Yuan Xie, N. Vijaykrishnan, and Mary Jane Irwin
Department of Computer Science and Engineering, Penn State University
{ytsai,yuanxie,vijay,mji}@cse.psu.edu

Abstract

As technology scales, interconnects dominate the performance and power behavior of deep submicron designs. Three-dimensional integrated circuits (3D ICs) have been proposed as a way to mitigate the interconnect challenges. In this paper, we explore the architectural design of cache memories using 3D circuits. We present a delay and energy model, 3DCacti, to explore different 3D design options of partitioning a cache. The tool allows partitioning of the cache across different device layers at various levels of granularity. The tool has been validated by comparing its results with those obtained from circuit simulation of custom 3D layouts. We also explore the effects of various cache partitioning parameters and 3D technology parameters on delay and energy to demonstrate the utility of the tool.

1. Introduction

Interconnects dominate the performance and power behavior of deep submicron designs. Consequently, interconnect centric design methods and technology improvements are critical to the chip industry. While there have been significant interconnect technology improvements over the last few years, such as the use of copper and low-K dielectric, the industry is striving for additional improvements. The various technologies being actively explored to address the interconnect problem include the use of packet-based on chip communication networks [1], the use of angular wires instead of Manhattan routing [2] and the use of three-dimensional chips [3]. Three-dimensional chips are also attractive for additional reasons such as support for realization of monolithic mixed-technology chips.

A three dimensional (3D) chip is a stack of multiple device layers with direct vertical interconnects tunneling through them. A key benefit of this approach over a traditional two dimensional chip is the ability to reduce the length of long interconnects. Prior efforts have focused on developing different fabrication techniques involved in stacking multiple device layers and in forming the vertical interconnects. The size and density of the vertical interconnects that can tunnel between the different device layers varies based on the underlying technology used to fabricate the 3D chips.

To efficiently exploit the benefits of 3D technologies, design techniques and methodologies for supporting 3D designs are imperative. Recent efforts have focused on developing tools for supporting custom 3D layouts and placement tools [3]. In [4], the technology and testing issues are surveyed and the 3D integration framework is presented. However, the investigation of benefits at the architectural level for 3D designs is in its infancy. The authors in [5] study the energy and thermal performance of 3D designs under a supplied time constraint. However, their tool is based on a standard-cell circuit layout and the 3D design space is not fully explored. A recent paper provides an overview of the potential benefits of designing an IA32 processor in 3D technology [6]. However, it does not provide details of the design of the individual components. In [14], a 3D shared memory is fabricated. In this shared memory system design, six memory modules are distributed into three device layers and high data bandwidth is achieved by connecting broadcast bus in both horizontal and vertical directions (3D vertical interconnects). Another effort that is closely related is [7], in which the authors show an overview of benefits expected from the design of a carry look-ahead adder in 3D technology.

In this paper, we explore the architectural design of cache memories using 3D structures. Since interconnects dominate the delay of cache accesses, which determines the critical path of a microprocessor, the exploration of benefits from advanced technologies is particularly important. The regular structure and long wires in a cache make it one of the best candidates for 3D designs. A tool to predict the delay and energy of a cache is crucial as the timing profile and the optimized configurations of cache depend on the number of active device level available as well as the way the cache is partitioned into different active device layers. This paper examines possible partitioning techniques for caches designed using 3D structures and presents a delay and energy model to explore different options for partitioning a cache across different device layers.

The paper is organized as follows. Section 2 presents a background on 3D technology and cache structure; Section 3 discusses the possible 3D partitioning techniques for a cache; Section 4 presents a 3D cache delay-energy estimator called 3DCacti, and the results exploring the design space for 3D cache are presented in Section 5; Section 6 concludes the paper.

2. Background

In this section, we first give a brief introduction on different 3D technologies and the influence on 3D cache design space exploration. We then provide an overview of the basic structure of a conventional cache.

2.1 3D Technology Basics

There are various 3D technologies that have been explored in the literature. For the exploration of 3D cache designs we consider two most promising styles of 3D technologies: technologies similar to *wafer-bonding technology* [3] and those similar to *Multi-layer Buried Structures* (MLBS) technology [8]. Figure 1 shows the basic structure of these

[*] Acknowledgements: This work is supported in part by NSF 0093085, NSF 0202007, and MARCO/DARPA-GSRC, and SRC 00541.

two technologies: Wafer-bonding technology connects active device layers after processing each active device layer separately while MLBS sequentially processes layers of active devices (transistors) before processing all the metal routing layers. Only face-to-back type of wafer-bonding is shown given two active device layers will be discussed in this paper. Note that face-to-face wafer-bonding provides higher via density while it allows only two active device layers in a 3D stack.

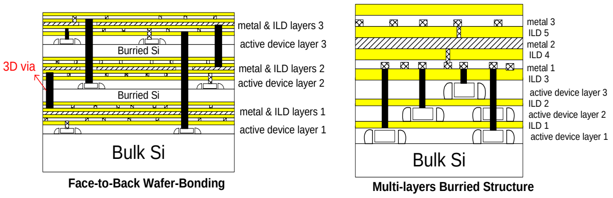

Figure 1: The structures of face-to-back wafer-bonding and MLBS. Note that drawings are not to scale and MLBS is enlarged for illustrating purpose.

A key difference between these two technologies that can influence the 3D cache partitioning strategy is the size of the vertical 3D vias, which provide connections between the different active device layers. In wafer-bonding, the dimensions of the 3D via are not expected to scale at the same rate as feature size, because wafer-to-wafer alignment tolerances during bonding pose limitations on the scaling of the vias [3]. Current dimensions of 3D via sizes vary from 1µm-by-1µm to 10µm-by-10µm. The relatively large size of via can hinder partitioning the cache at very fine granularities across multiple device layers.

The MLBS provides more flexibility in vertical 3D connection because the vertical 3D via can potentially scale down with feature size due to the use of local poly-Si wires for connection [3]. Availability of such technologies makes it possible to partition the cache at the granularity of individual cache cells [18]. However, wafer-bonding requires fewer changes in the manufacturing process and is more popular in industry [6] [13] than MLBS technology. Therefore, our 3D cache design space exploration is mainly focused on using wafer-bonding technology.

2.2 Cache Structure

The conceptual structure of a cache contains a tag array and a data array. A portion of the address bits are used to index the corresponding set in the tag and data array. Next, the tags and data of the different blocks belonging to a set are read. The tags read from all the blocks are compared against the tag portion of the incoming address. The indication of a match from the comparator output is used to enable the output driver of the corresponding block's data from the data array.

Neither the tag nor the data arrays are monolithic structures; the wordlines and bitlines of the memory array are divided into Ndwl and Ndbl parts resulting in Ndwl*Ndbl subarrays. This partitioning is effective in reducing the access times and power consumption. Since the dimensions of the tag and data arrays are different, they are typically partitioned differently. In a 3D structure, we can extend this partitioning approach to divide bit lines and word lines across different device layers. We will refer to this methodology as intra-subarray partitioning in Section 3.

In addition to influencing the design of individual arrays, the use of 3D structures can also help to reduce the delays due to global interconnects in the cache. One of the global interconnects are the incoming address inputs to the cache that are sent to a predecoder, which is placed in the center of the subarrays. The predecoded address signals then traverse in an H-tree format to the local decoders of the subarrays. The local decoders in turn drive the corresponding word line drivers. Other global signals include the select signals for driving the output buffers of the data array, the wires from output driver to the edge of the array, and the select lines for write and multiplexer control. All these global signals should benefit from a smaller footprint through the use of 3D technology. Global clock wiring will also benefit from 3D cache design as it travels shorter distance, even though in our evaluation we do not account for the benefit of the clock network.

3. 3D Cache Partitioning Strategies

In this section, we illustrate different granularities at which the cache can be partitioned to utilize the multiple device layers. A combination of partitioning at the different granularities discussed in this section is also possible.

3.1 SRAM cell level partitioning

The finest granularity of partitioning is at the SRAM cell level. At this level of partitioning, any of the six transistors of a SRAM cell can be assigned to any layers. For example, the pull-up PMOS transistors can be in one device layer and the access transistors and the pull-down NMOS transistors can be in another layer. The benefits of cell level partitioning include the reduction in footprint of the cache arrays and, consequently, the routing distance of the global signals discussed in section 2.2. The number and complexity of the logic gates remain the same as the conventional 2D designs.

However, the feasibility of partitioning at this level is constrained by the 3D via size as compared to the SRAM cell size. When the 3D via size does not scale with feature size as currently in wafer-bonding, partitioning at the cell level is difficult in future technology nodes. In contrast, partitioning at SRAM cell level will continue to be feasible in technologies such as MLBS, because no limitations are imposed on via scaling with feature size. However, it should be noted that even if the size of 3D via can be scaled to as small as a nominal contact in a given technology, the total SRAM cell area reduction (as compared to a 2D design) due to the use of additional layers is limited, because metal routing and contacts occupy a significant portion of the 2D SRAM cell area [13]. Consequently, partitioning at higher levels need to be explored.

3.2 Intra-sub-array partitioning

At this level of partitioning, the individual sub-arrays in the 2D cache are partitioned across multiple device layers. The partitioning at this granularity reduces the footprint of cache array and routing length of global signals. However, it also changes the complexity of the peripheral circuits. In our research, we consider two options of partitioning the subarray into multiple layers: 3D divided wordline (3DWL) strategy and 3D divided bit line strategy (3DBL)

3D Divided Wordline (3DWL) - In this partitioning strategy, the wordlines in a sub-array are divided and mapped onto different active device layers (See Figure 2). The corresponding local wordline decoder of the original wordline in 2D subarray is placed on one layer and is used to feed the wordline drivers on different layers through the 3D vias. Instead of a single wordline driver as in the 2D case, we have multiple word line drivers in the new design for each layer. The duplication overhead is offset by the resized drivers for a smaller capacitive load on the partitioned word line. Further, the delay time of pulling a wordline decreases as the number of pass transistors connected to a wordline driver is smaller. The delay calculation of the 3DWL also accounts for the 3D via area utilization. The area overhead due to 3D vias is small compared to the number of cells on a word line and is far smaller than the gains obtained due to partitioning.

Another benefit from 3DWL is that the distance of the address line from periphery of the core to the wordline decoder decreases proportional to the number of device layers. Similarly, the routing distance between the output of pre-decoder to the local decoder is reduced. The select lines for the writes and muxes as well as the wires from the output drivers to the periphery also reduce.

3D Divided Bitline (3DBL) - This approach is akin to the 3DWL strategy and applies partitioning to the bitlines of a subarray (See Figure 3). The bitline length in the subarray as well as the pass transistors connected to a single bitline is reduced. In the 3DBL approach, the sense amplifiers can either be duplicated across the different device layers or shared between the partitioned subarrays in the different layers. The former approach is more suitable for reducing access times while the latter is preferred for reducing number of transistors and leakage. In latter approach, the sharing increases complexity of multiplexing of bitlines and reduces performance as compared to the former. Similar to 3DWL, the length of the global lines are reduced in this scheme.

4. 3D cache delay-energy estimator (3DCacti)

In order to explore the 3D cache design space, we developed a 3D cache delay-energy estimation tool called 3DCacti. Our tool was built on top of the Cacti 3.0 2D cache tool [9]. In addition to the 3D enhancements, we have also improved the models used for technology scaling and leakage power in the base 2D case. 3DCacti searches for the optimized configuration that provides the best delay, power, and area efficiency trade-off according to the cost function for a given number of different 3D partitions. The cost function for each configuration *i* is evaluated as:

$$\cos t_i = \frac{energy_i}{\max energy} * W_e + \frac{accesstime_i}{\max accesstime} * W_t + \frac{1}{areaefficiency_i} * W_{ae} + \frac{aspectratio_i}{\max aspectratio} * W_{ar}$$

where W_e, W_t, W_{ae} and W_{ar} are the weight of energy, delay, area efficiency, and array aspect ratio, respectively.

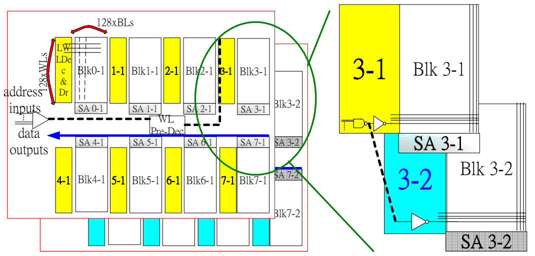

Figure 2: Cache with 3D divided wordline partitioning and mapped into 2 active device layers.

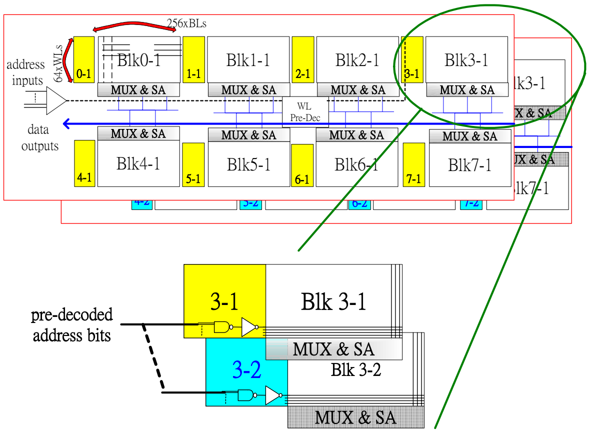

Figure 3: Cache with 3D divided bitline partitioning and mapped into 2 active device layers.

4.1 Implementation of 3DCacti

In this section, we emphasize only the portions of 3DCacti that are different from the original Cacti 3.0. We have grouped these changes into delay models, layout parameters and technology parameters.

Delay models: To model the resistance and capacitance of the 3D vias, we add the RC delay to implement the intra-array partitioning. The resistance of 3D via is estimated to be 10^{-8} ohm-cm^2 based on actual resistance measurement [10], and the capacitance is estimated as the capacitance of a 1 μ m-by-1 μ m contact using top metal layer and the height of the interlayer via is assumed to be 10um. With technology scaling, repeater insertion for long wires is necessary. We have also enhanced the wire delay model by implementing repeater insertion for long interconnects. We assume that repeaters are optimum sized and placed in the global routing wires where repeater insertion is possible, and thus the interconnect delay is modeled as in [16]:

$$D_{wire} = 2 * \sqrt{t_{unbuffer} * t_p}$$

where $t_{unbuferf}$ and t_p are the original interconnect delay without repeaters inserted and the delay of the optimized sized repeaters, respectively.

Layout parameters: Several configuration parameters are used in Cacti to divide a cache into sub-arrays for achieving delay, energy, and area efficiency trade-offs. In our implementation, two additional parameters, Nx and Ny, are added to model the intra-subarray 3D partitions. The cell level partitioning approach is implicitly simulated using a different cell width and height within Cacti.

Table 1 lists the definitions and effects of the different configuration parameters. The additional effects of varying each parameter other than the impact on length of global routing signals are listed in Table 1. Note that the tag array is optimized independently of the data array and the configuration parameters for tag array: Ntwl, Ntbl, and Ntspd are not detailed here.

Table 1: List of configuration parameters in 3DCacti.

	Definition	Effect
Ndbl	# of cuts on a cache to divide bitline	1. bitline length in each sub-array 2. number of sense amplifier 3. size of wordline driver 4. decoder complexity 5. multiplexers complexity in data output path
Ndwl	# of cuts on a cache to divide wordline	1. wordline length in each sub-array 2. number of wordline driver 3. decoder complexity
Nspd	# of sets connected to a wordline	1. wordline length in each sub-array 2. size of wordline driver 3. multiplexers complexity in data output path
Nx	# of intra-sub-array partitions by dividing wordline	1. wordline length in each sub-array 2. size of wordline driver
Ny	# of intra-sub-array partitions by dividing bitline	1. bitline length in a sub-array 2. complexity in multiplexers in data output path

Technology parameters: The scaling of the delay in transistors is different from that in wires. The original Cacti is built based on the technology parameters of 0.8µm technology. To estimate the delay and energy for smaller technologies, instead of applying the linear scaling on the final delay and energy numbers as in Cacti, we apply more accurate scaling rules derived from [11] on each individual technology parameter. We also assume the use of copper interconnect for technologies smaller than 0.18µm and account for the fact that aspect ratio of 6T SRAM is getting smaller as technology scales. We also adopt the "wide-bit" cell design for technologies smaller than 70nm according to the SRAM design fabricated in 65nm [12]. We have also augmented leakage models in the cache to account for leakage energy [17]. Different from [17], we use the transistor sizes scaled from original Cacti and insert repeaters where there are long wires or large loadings. The sizes of all transistors and repeaters are coupled with the configuration information to account for the transistor counts for estimating total leakage power. The increasing temperature due to the stacking [5] was factored into the leakage estimates by temperature estimates performed using an internal thermal analysis tool built on top of HotSpot [15]. The thermal analysis tool is first fed with the power and area information assuming operating at room temperature and the estimated temperature is then feedback to 3DCacti to account for the thermal difference due to stacking. Our results of a 1M bytes cache when being accessed every cycle in 25nm show that chip temperature rises from 45^{o}C in 2D chip to 50^{o}C ,65^{o}C, 92^{o}C, and 148^{o}C when stacking 2, 4, 8,and 16 layers of devices layers, respectively. Therefore, the increase in leakage power for stacking 2 and 4 device layers is negligible, but the increase in leakage power for stacking 8 and 16 active device layers are 3 times and 6 times, respectively. Our results show that using 2 or 4 active device layers, which are more practical for wafer stacking, the leakage energy caused by thermal issue is not significant. However, using more than 8 device layers will not provide any reduction in total energy when temperature-dependent leakage is significant. This temperature-leakage inter-dependence is accounted for in 3DCacti.

4.2 Validation of 3DCacti

In order to validate 3DCacti, we implemented the layouts of a 32KB 2-way set associative cache with 16 byte blocks in TSMC 180nm technology with a publicly available design and layout extraction tool called Magic, for three different cases. The three cases include a 2D layout and 3D layouts employing 3DWL and 3DBL with two device layers. For the 3D layouts, the layout for each layer was designed separately and the 3D vias were modeled as special contacts. The RC characteristics of the vias were modeled based on parameters provided in Section 4.1.

The 3DCacti estimates of delay and power are compared against Hspice simulation results from the layouts. The savings in delay and power of 3DWL and 3DBL as compared to the 2D design estimated by 3DCacti and HSPICE are shown in Table 2. We observe that the relative delay and power trends for these designs predicted by our model and the actual designs are similar. We validate only the relative trends instead of absolute numbers as the underlying technology parameters used by TSMC 180nm process by HSPICE and the scaled technology numbers used in 3DCacti for 180nm are different. Also, the use of our tool is envisioned in architectural exploration, where accurate relative trends are sufficient for making design decisions. The relative delay of each individual component of the cache for 3DWL and 3DBL demonstrates that each individual component delay modeled by 3DCacti also matches with HSPICE simulation results.

Table 2. Reductions in delay and energy of estimates from 3DCacti and Hspice simulation results of layout as compared to a 2D cache.

Method	3DCacti 2x1	HSPICE 2x1	3DCacti 1x2 3D	HSPICE 1x2
Access Time Savings	17.75%	17.33%	6.15%	6.91%
Power Savings	14.95%	9.23%	15.71%	10.07%

5. Design Exploration using 3DCacti

In this section, we explore various 3D partitioning options of caches using 3DCacti to understand their impact on delay and power. Further, we investigate the influence of system requirements on the 3D cache performance. Note that the data presented in this section is in 70nm technology, assuming 1 read/write port and 1 bank in each cache; unless otherwise stated.

5.1 3D Partitioning for Performance

First, we explored the best configurations for various degrees of 3DWL and 3DBL in terms of delay. Figure 4 shows the access delay and energy consumption per access for 4-way set associative caches of various sizes and different 3D partitioning settings. Recollect that Nx (Ny) in the configuration refers to the degree of 3DWL (3DBL) partitioning. First, we observe that delay reduces as the number of layers increase. On a more detailed look at the reason for this delay reduction using Figure 5, we observe that the reduction in global wiring length of the decoder is the main reason for benefits from multiple layers. We also observe that for the 2-layer case, the partitioning of a single cell using MLBS provides delay reduction benefits similar to the best intra-subarray partitioning technique as compared to the 2D design.

Another general trend observed for all cache sizes indicates that partitioning more aggressively using 3DWL provides caches with smaller delay. For example, in the 4-layer case, the configuration 4*1 has a delay that is 16.3% less than that of the 1*4 configuration for a 1MB cache. We observed that the benefits from more aggressive 3DWL stem from the longer length of the global wires in the X direction as compared to the Y direction before 3D partitioning is performed. The preference for shorter bitlines for delay minimization in each of the subarrays and the resulting wider subarrays in optimal 2D configuration is the reason for the difference in wire lengths along the two directions. For a cache whose best subarray configuration before 3D partitioning results in the global wire length in the X direction being longer, when wordlines are divided along the third dimension, they result in more significant reduction in critical global wiring lengths. It must be observed that because 3DCacti is exploring partitioning across the dimensions simultaneously, some configurations can result in 2D configurations that have wirelengths greater in the Y directions as in the 1MB cache 1*2 configuration for 2 layers. The 3DBL helps in reducing the global wire length delays by reducing the Y direction length. However, it is still not as effective as the corresponding 2*1 configuration as both the bitline delays in the core and the routing delays are larger (See Figure 5). These trends are difficult to analyze without the help of a tool to partition across multiple dimensions simultaneously.

The energy behavior for the corresponding best delay configurations tracks the delay behavior in many cases. For example, the 1MB cache energy behavior increases when moving from a 8*1 configuration to a 1*8 configuration. In these cases, the capacitive loading effects affecting delay also determine the energy trends. However, in some cases, the energy reduces significantly when changing configurations and does not track performance behavior. For example, for the 512KB cache using 8-layers, the energy reduces when moving from 2*4 to 1*8 configuration. This stems from the difference in the number of sense amplifiers activated in these configurations due to the different number of bitlines in the each subarray in the different configurations and the presence of the column decoder after the sense amplifier. Specifically, the optimum (Ndwl,Ndbl,Nspd) for the 512KB case is (32,1,16) for the 2*4 case and (32,1,8) for the 1*8 configuration. Consequently, the number of sense amplifiers activated per access for 1*8 configuration is half as much as that of the 2*4 configuration resulting in a smaller energy.

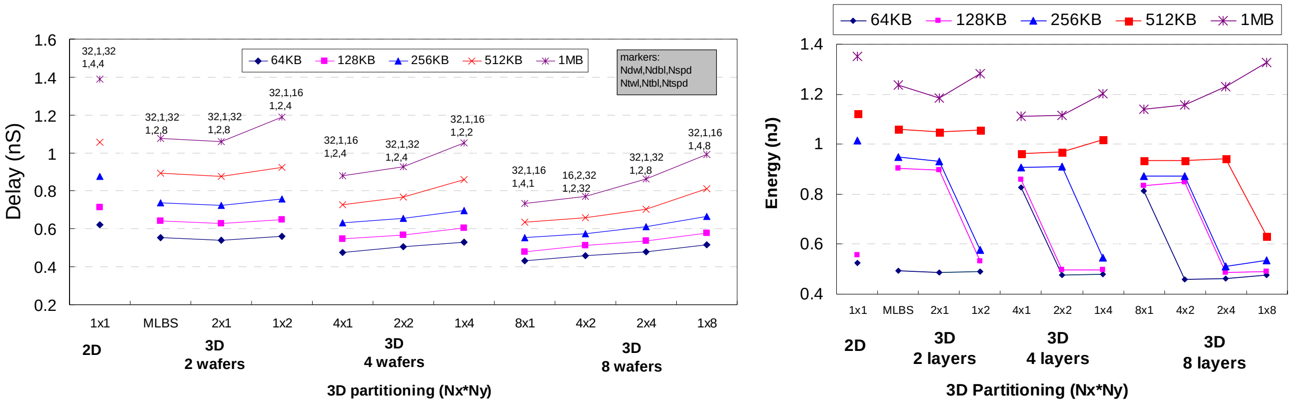

Figure 4: Access time and energy for different partitioning when setting the weightage of delay higher. Data of caches of associativity=4 are shown.

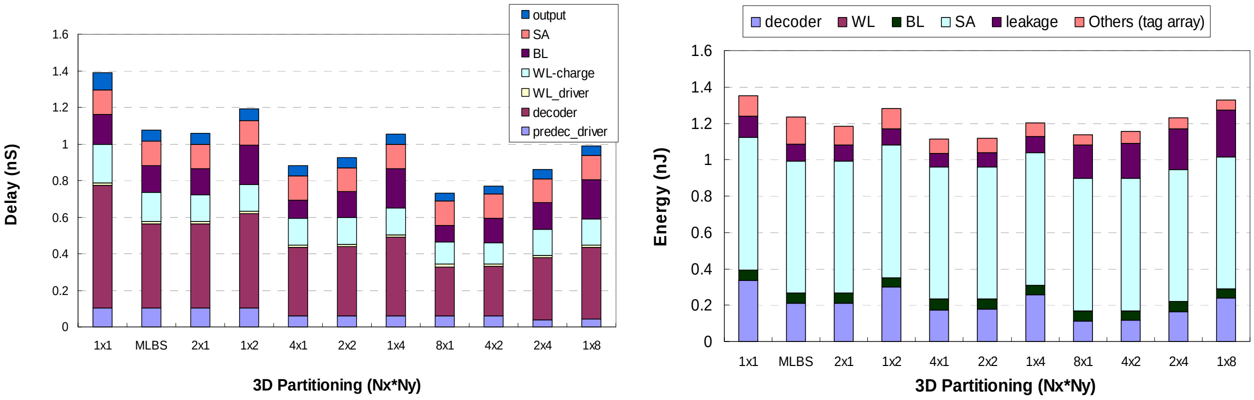

Figure 5: Access time and energy breakdown of a 1MB cache corresponding to the results shown in Figure 4.

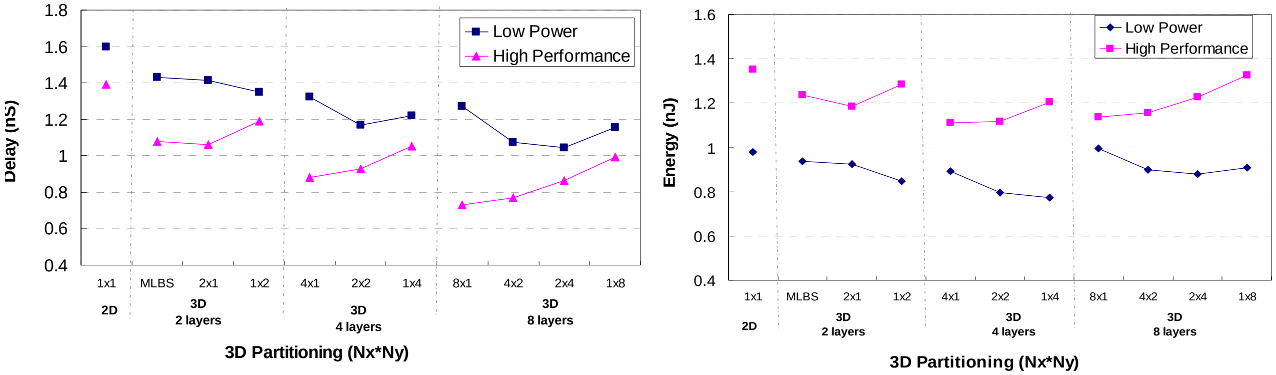

Figure 5: Comparison of access time and energy for different partitioning for high performance and low power systems. Data of associativity=4 are shown.

5.2 Varying cost function

The delay/energy savings achieved by 3D partitioning depends on the trade-off between delay, energy, and area efficiency. This trade-off varies according to system requirements. To explore the impact of system requirements on the delay/energy savings, we experiment on a 1MB cache (associativity=4) for different system requirements. We set the weights in the cost function to be (W_e:W_t:W_{ae}:W_{ar}) = (1:1000:1:1) and (W_e:W_t:W_{ae}:W_{ar}) = (1000:1:1:1) for high performance system and low power system, respectively. The results are shown in Figure 6. We can see that 3DWL is more efficient than 3DBL in terms of both delay and energy in high performance systems while 3DBL is more effective in low power systems. This is due to that when optimizing for low power, the optimum configuration partitions a cache in a fashion that the routing in y direction is longer than that in x direction which is different from that in high performance systems. The optimum configuration with each layer changes across different 3D partitioning strategies and system requirements. Further, the power behavior of the least delay time configurations for different 3D partitioning are quite different. Consequently, the simultaneous exploration of the different parameters along with desired weightage function to different cost function is important in deciding the 3D partitioning.

6. Conclusions

In this paper, we explore the architectural design of cache memories using 3D technologies. A cache delay and energy estimator called 3DCacti is proposed to explore different options for partitioning a cache across different active device layers. The estimator has been validated using actual designs. We observe that the savings in delay/energy through 3D partitioning depends on the cache size, system requirements, the number of device layers and technology nodes.

Acknowledgement: The authors would like to thank Kerry Bernstein and Albert Young from IBM for their help.

7. Reference

[1] Benini,L. and Micheli, G. D., "Networks on Chips: A New Soc Paradigm", Computer, Vol.35 No.01, 2002

[2] X initiative, http://www.virtual-silicon.com/

[3] Das, S., et al, "Technology, Performance, and computer-aided Design of Three-Dimensional integrated Circuits", ISPD, 2004

[4] Deng, Y., et al, " 2.5D System Integration: A Design Driven System Implementation Schema", ASP-DAC, 2004

[5] Das, S., "Timing, Energy, and Thermal Performance of Three-Dimensional Integrated Circuits", GLSVLSI 2004

[6] Black, B., et al, "3D Processing technology and Its Impact on IA32 Microprocessors", ICCD, 2004

[7] Mayega, J., et al, "3D Direct Vertical Interconnect Microprocessors test Vehicle", GLVLSI, 2003

[8] Jung, S.M., et al, "The Revolutionary and Truly 3-Dimentional 25F2 SRAM Technology with the Smallest S3 Cell, 0.16um2 and SSTFF for Ultra High Density SRAM", VLSI Technology Digest of Technical Papers, 2004

[9] Shivakumar, P., et al, "Cacti 3.0: An Integrated Cache Timing, Power, and Area Model", Western Research Lab. Research Report, 2001/2.

[10] Chen, K. N., et al, "Contact Resistance Measurement of Bonded Copper Interconnects for Three-Dimensional Integration Technology", IEEE Electron Devices Letters, 25 (1), 2004

[11] Farland, G.," CMOS Technology Scaling and Its Impact On Cache Delay", Ph. D. Thesis, Stanford University, 1997

[12] Zhang, K., et al, "A SRAM Design on 65nm CMOS technology with Integrated Leakage Reduction Scheme", IEEE Symp. On VLSI Circuits Digest of Technical Papers, 2004

[13] Ieong, M, et al, "Three Dimensional CMOS Devices and Integrated Circuits", CICC, 2003

[14] Lee,K.W., et al, "Three-Dimensional Shared Memory Fabricated Using Wafer Stacking Technology", IEDM, 2000

[15] "HotSpot Thermal Modeling Simulator", http://lava.cs.virginia.edu/HotSpot/

[16] Rabaey, J. ,*Digital Integrated Circuit Design 2nd Edition*, Princeton-Hall publication, 2002

[17] Mamidipaka, M., "Analytical Models for Leakage Power Estimation of Memory Array Structures", CODES+ISSS, 2004

[18] Kang, Y. H., et al, "Fabrication and Characteristics of Novel Load PMOS SSTFT (Stacked Single-crystal Thin Film Transistor) for 3-dimentional SRAM Memory Cell ", SOI Conference, 2004