

# Investigation and Comparison of Thermal Distribution in Synchronous and Asynchronous 3D ICs

Brent Hollosi<sup>1</sup>, Tao Zhang<sup>2</sup>, Ravi S. P. Nair<sup>3</sup>, Yuan Xie<sup>2</sup>, Jia Di<sup>1</sup>, and Scott Smith<sup>3</sup>

<sup>1</sup>Computer Science & Computer Engineering Department, University of Arkansas

<sup>2</sup>Department of Computer Science & Engineering, Pennsylvania State University

<sup>3</sup>Electrical Engineering Department, University of Arkansas

**Abstract**-This paper presents an analysis and comparison between synchronous and delay-insensitive asynchronous logic circuits on thermal distributions for investigating novel solutions to the heat dissipation problem in three-dimensional ICs. Due to the spatial and temporal distribution of switching activities in delay-insensitive asynchronous circuits, the thermal density as well as the temperature is largely reduced. Results show that the sample delay-insensitive asynchronous circuit exhibits lower average temperature and more uniform thermal distribution compared to its synchronous counterpart.

## I. INTRODUCTION

Interconnect is one of the major concerns in current and future IC designs from both performance and power consumption perspective. The emergence of three-dimensional (3D) ICs, with their intrinsic capability to reduce wire length, is one of the promising solutions to mitigate interconnect-related issues. In 3D technology, device layers are processed separately and stacked vertically with 3D vias providing the vertical connection between dies. One key advantage of 3D chips over traditional two-dimensional chips is the direct wire length reduction. This in-turn reduces parasitics associated with long interconnects, and leads to latency improvement due to reduced wire lengths.

In 3D ICs, however, thermal issues are considered to be a potential problem, because of the increasing power density in die stacking. Without special design, this thermal problem could be worse in synchronous, clocked 3D circuits. In such circuits, the simultaneous switching activities of all clocked elements, e.g., clock tree and flip-flops, result in certain “hot spots” that form around these elements, which in turn causes significant temperature differences across the IC. If the heat generated by these hot spots cannot be removed in a timely manner, the aforementioned thermal problem arises.

Delay-insensitive asynchronous circuits, on the other hand, do not have a clock. Instead, such circuits use handshaking signals to control circuit behavior. Due to the fact that the individual gate delay does not affect circuit functionality, a number of advantages, including low power, robust operation, and low noise/emission, can be achieved. One interesting feature of delay-insensitive asynchronous circuits, which is explored in this paper, is its distributed switching activity. In contrast to the centralized switching activity of synchronous circuits, there is no clock in delay-insensitive asynchronous

circuits, such that at any given time only those circuit elements that are currently processing data switch. Therefore, if the 3D IC consists of delay-insensitive asynchronous circuits, the thermal distribution problem can be largely alleviated.

In this paper the thermal distributions of synchronous and delay-insensitive asynchronous 32-bit floating-point co-processors are investigated and compared. The preliminary results show that the delay-insensitive asynchronous circuit exhibits more uniform thermal distribution compared to its synchronous counterpart.

## II. BACKGROUND

### A. Thermal Distribution Problem in 3D ICs

As process technologies scaled, gate performance improved, while the performance and power of interconnects began to dominate the system in recent years. To alleviate the power stress from interconnects, 3D IC design has been proposed as a promising solution. Although 3D ICs can dramatically shorten intermediate and/or global interconnects, and thus reduce power consumption, a major challenge in the adoption of 3D architecture is the increasing power density, which results from multi-layered 3D chip stacking [1]. Increased chip temperature can cause various problems, such as elevated leakage power, accelerating functional and timing failures. Furthermore, higher leakage power can in turn lead to further temperature increases, which makes the situation even worse.

To resolve this important 3D IC thermal problem, a number of solutions have been proposed recently. One effective way is to do thermal-aware floorplanning for 3D ICs [2][3][4]. Based on the thermal simulation results, the floorplanner can easily place the blocks to achieve a more uniform thermal distribution so that the average temperature of the chip can be reduced. On the other hand, thermal-aware task scheduling techniques are also proposed to control the chip temperature [5][6]. In addition, some other techniques, such as DVS (Dynamic Voltage Scaling) and DFS (Dynamic Frequency Scaling), can also be used as potential solutions.

Distinguished from the ideas mentioned above, the delay-insensitive asynchronous circuit design technique, on which we focus in this paper, presents a novel direction to lower chip temperature and make the thermal distribution more even and uniform. Without a clock tree, which is necessary for synchronous circuit, delay-insensitive asynchronous circuits

can save power, which directly results in lower temperature. Furthermore, delay-insensitive asynchronous circuits better distribute switching activity over time and space, which makes delay-insensitive asynchronous circuits even more competitive and attractive.

### B. Delay-Insensitive Asynchronous Logic and NULL Convention Logic

Delay-insensitive asynchronous (clockless) design styles, like NULL Convention Logic (NCL) [7], require very little, if any, timing analysis to ensure correct operation; they are said to be correct-by-construction. NCL circuits utilize multi-rail signals, such as dual-rail logic, to achieve delay-insensitivity. A dual-rail signal,  $D$ , consists of two wires,  $D^0$  and  $D^1$ . The DATA0 state ( $D^0 = 1, D^1 = 0$ ) corresponds to a Boolean logic 0, the DATA1 state ( $D^0 = 0, D^1 = 1$ ) corresponds to a Boolean logic 1, and the NULL state ( $D^0 = 0, D^1 = 0$ ) corresponds to the empty set meaning that the value of  $D$  is not yet available [8]. The two rails are mutually exclusive, such that both rails can never be asserted simultaneously; this state ( $D^0 = 1, D^1 = 1$ ) is defined as an illegal state. NCL circuits are comprised of 27 fundamental gates, called threshold gates [9]. The primary type of threshold gate is the TH $m$ n gate, depicted in Fig. 1, where  $1 \leq m \leq n$ . TH $m$ n gates have  $n$  inputs; at least  $m$  of the  $n$  inputs must be asserted before the output will become asserted; and NCL threshold gates are designed with *hysteresis* state-holding capability, such that after the output is asserted, all inputs must be deasserted before the output will be deasserted. Hysteresis ensures a complete transition of inputs back to NULL before asserting the output associated with the next wavefront of input data. Therefore, a TH $n$ n gate is equivalent to an  $n$ -input C-element [10]; and a TH1 $n$  gate is equivalent to an  $n$ -input OR gate.

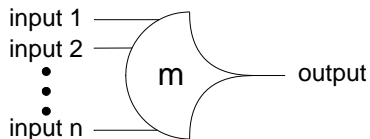


Fig. 1. TH $m$ n gate

NCL circuits communicate using request and acknowledge signals,  $K_i$  and  $K_o$ , respectively, to prevent the current DATA wavefront from overwriting the previous DATA wavefront, by ensuring that the two DATA wavefronts are always separated by a NULL wavefront [11]. The acknowledge signal from the receiving circuit is the request signal to the sending circuit. When the receiving circuit latches the input DATA, the corresponding  $K_o$  signal will be logic 0, indicating a *request-for-NULL (rfn)*; and when it latches the input NULL, the corresponding  $K_o$  signal will be logic 1, indicating a *request-for-DATA (rfd)*. When the sending circuit receives a *rfd/rfn* on its  $K_i$  input, it will allow a DATA/NULL wavefront to be output, respectively. This delay-insensitive handshaking protocol coordinates NCL circuit behavior, analogous to coordination of synchronous circuits by a clock signal. Additionally, delay-insensitivity requires a circuit to be *input-complete*, which means that all outputs may not transition from NULL to DATA until all inputs have transitioned from

NULL to DATA, and that all outputs may not transition from DATA to NULL until all inputs have transitioned from DATA to NULL [12].

## III. TECHNICAL RATIONALE

### A. Distributed Switching Activities in NCL Circuits

In NCL circuits, the distribution of switching activities can be viewed from two angles: spatial and temporal. The spatial distribution accounts for the locations where switching activities occur. In synchronous circuits, the circuit elements along the clock tree, e.g., buffers, switch every clock cycle. Heat generated at the locations of these elements will cause the temperature to rise, such that these locations are much hotter than other parts of the die. NCL circuits, on the other hand, do not have clock. The handshaking signals, which control circuit behavior, are incorporated within each pipeline stage and are spread across the chip. Therefore, there is no centralized high-temperature location.

Temporal distribution refers to the times when switching activities occur. In synchronous circuits, due to the coordination of clocks, most switching activities occur at the active edge of each clock cycle. In other words, all clock-controlled circuit elements switch at the same time. This results in a large peak current being drawn from the power supply. In contrast, the switching activities in NCL circuits are driven by data instead of clock. At any given time only those circuit elements that are currently processing data switch, while others remain idle. This feature can be viewed as automatic, fine-grained clock gating.

### B. Design of Sample Circuits for Thermal Distribution Comparison

To investigate the thermal distribution effect, a pipelined IEEE 32-bit single-precision floating-point co-processor was implemented using NCL and Boolean methods. The co-processors can perform addition, subtraction, and multiplication of two operands [13]. The design is divided into two sections: Adder/Subtractor and Multiplier. Fig. 2 shows the Adder/Subtractor logic, while Fig. 3 shows the Multiplier logic. The bold arrows in both figures show the longest path through the logic. Both circuits were pipelined into 16 stages such that the longest path delay was evenly distributed among all stages.

The Add/Sub bit is fed to the Multiplier only in the NCL version and this is to make the Multiplier circuit input-complete with respect to all inputs [12]. The NCL versions of Adder/Subtractor logic and the Multiplier logic were combined into the NCL co-processor as shown in Fig. 4. At first the NCL co-processor is reset to NULL and it requests DATA through  $K_o$ .  $X$  and  $Y$  are 32-bit IEEE single-precision floating-point inputs and are initially NULL, becoming DATA when  $K_o$  changes to *rfd*. The *Mul* bit determines whether the input DATA is passed to the Multiplier or Adder/Subtractor, by steering the DATA wavefront only to the appropriate circuit via the de-multiplexer register [14]. That means if the Multiplier logic receives DATA, the Adder/Subtractor remains NULL, and vice versa. At the output multiplexer register, the

same *Mul* bit is used to pass the DATA, from the circuit it chose earlier, through to output Z.

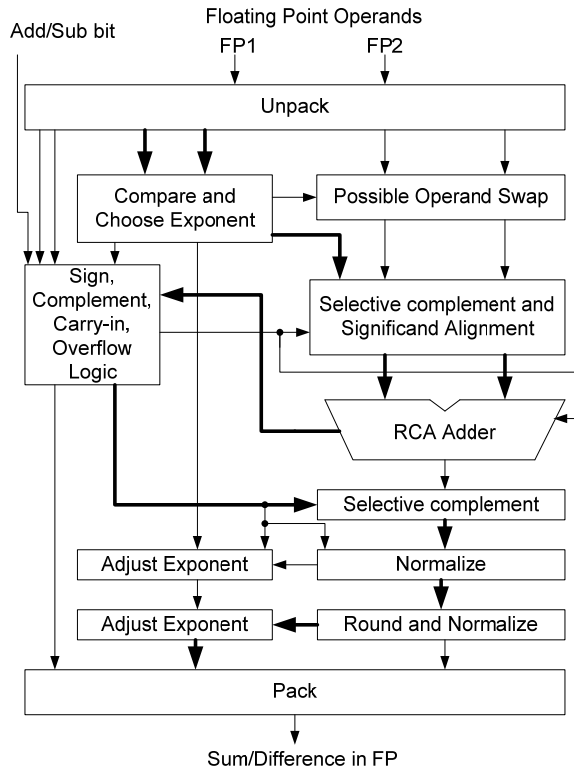


Fig. 2. Co-processor Adder/Subtractor Logic

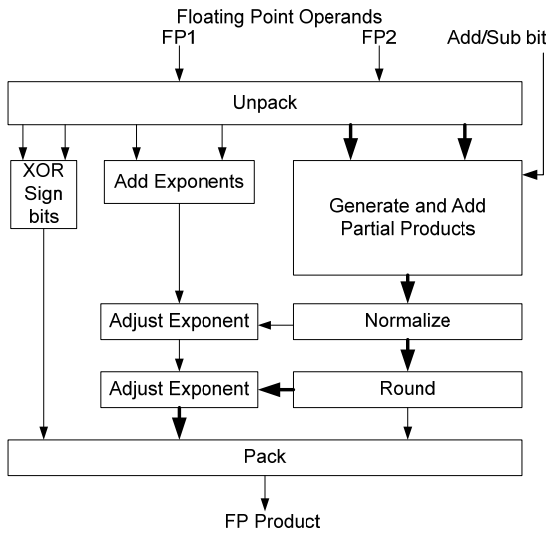


Fig. 3. Coprocessor Multiplier Logic

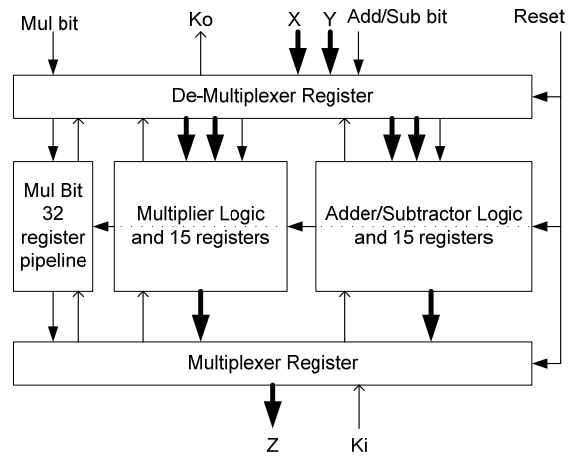


Fig. 4. NCL co-processor

When there is a request for DATA at  $K_i$ , the output Z becomes DATA. Note that at a time there can be 16 different DATA/NULL wavefronts in the NCL co-processor: 8 for the Adder/Subtractor and 8 for the Multiplier. So, a 32-bit pipeline stage is required for the *Mul* bit, to account for the 16 different operations (i.e., 2 stages for each DATA/NULL wavefront).

The Boolean versions of Adder/Subtractor logic and Multiplier logic were combined to form the Boolean co-processor shown in Fig. 5. The Boolean co-processor is a straightforward pipelined design as shown: input data is fed to both the Multiplier and the Adder/Subtractor sections. At the output, a multiplexer passes the output of the appropriate sections based on the *Mul* bit.

The co-processor designs for NCL and Boolean versions were written in VHDL and were verified using Mentor Graphics Modelsim.

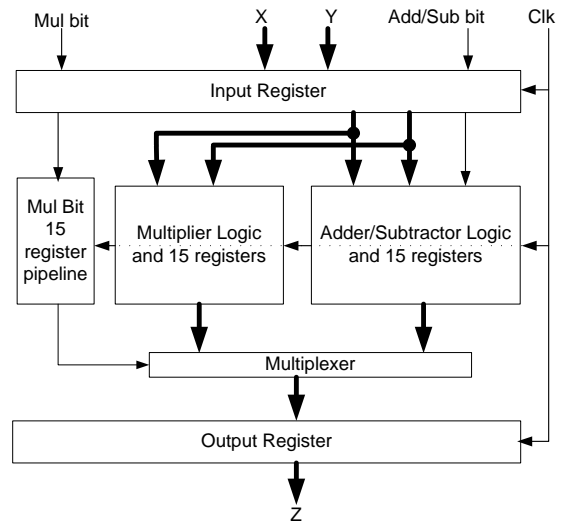


Fig. 5. Boolean co-processor

### C. Investigation and Comparison Strategy

The process of generating power data for the Boolean and NCL co-processors began with partitioning the two designs. Each design was partitioned into logically equivalent blocks to facilitate a more precise and directly comparable examination of power distribution. The partition scheme for each design is shown in Fig. 6 below.

Logic cells bounded by the initial and final register are partitioned and named according to the operation(s) performed, the pipeline stage number, and the cell type. Leading characters denote the addition/subtraction ('as') or multiplication ('m') operation. The number designates the pipeline stage. The final character 'g' or 'r' designates logic gates or registers. Each partitioned design was imported into Cadence using a previously developed and tested IBM sig5am cell library with schematics and layouts for all NCL and Boolean cells.

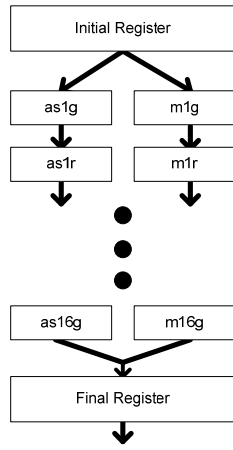


Fig. 6. Partition Scheme

For simulations, a VerilogA controller was coded to randomize input patterns and present them to each circuit at a rate of 5MHz. The supply current for each partition was measured and saved for each simulation. Each partition's current was sampled at 20 ns intervals making sure to include the current peaks during data transitions. The resulting current information was used to calculate the intervallic power information for each partition. Floorplans were generated for each design by examining the cell composition of each partition to ascertain its relative area and placing each partition according to its location in the data flow. This resulted in an analogous floorplan for each design, which facilitates a fair comparison of the thermal distribution amongst the partitions.

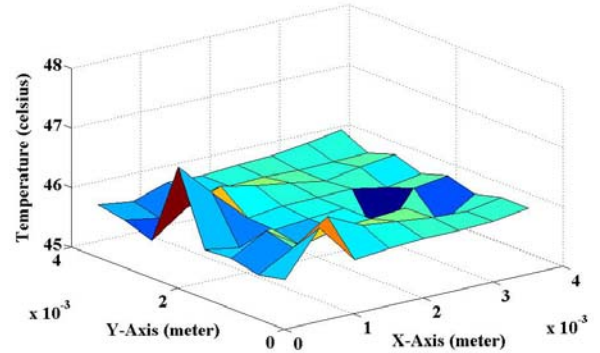
The thermal model we utilize to run the simulation is based on HotSpot, which includes HS3D. HotSpot reads the floorplan information and transient power values sampled from the design circuit as input data. Without loss of generality, we duplicate the asynchronous circuit and then stack these two asynchronous circuits together to form a two-layer 3D IC. The duplication is also applied to the baseline, synchronous circuit. After two-pass simulation (first run for the warm-up), HotSpot will output the entire thermal distribution. The detailed parameters of HotSpot are shown in Table I.

Version	4.1
Simulation Mode	grid-based
Initial Temperature	45°C
Ambient Temperature	45°C
Thermal Resistance	0.1 K/W
Number of Blocks	67
Number of Layers	2

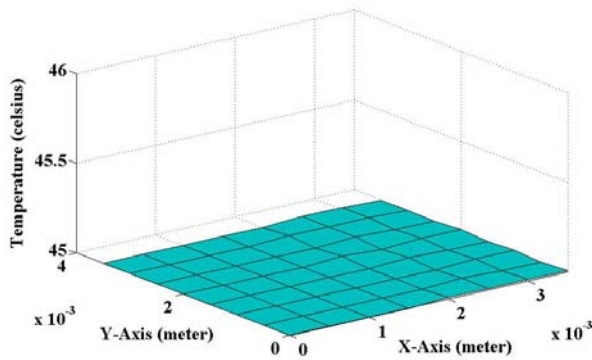
### IV. RESULTS AND ANALYSIS

Table II shows the experimental results, in which Layer1 is stacked over Layer0, and Fig. 7 gives a visual presentation for the thermal distribution in Layer0. As can be seen from the table and figure, while the average temperatures of the synchronous circuit are 46.01°C and 45.77°C in each layer, the average NCL circuit temperatures are only 45.01°C and 45.00°C. This is mainly from the high energy efficiency feature of NCL circuits, i.e., only the circuit elements that are currently processing data consume dynamic energy. Also, because Layer1 is closer to the heat sink, the average temperature of Layer1 is lower than that of Layer0. Furthermore, comparing the maximum (peak)/minimum temperatures between the synchronous and NCL circuits, which are 47.03°C/45.30°C and 45.01°C/45.00°C, respectively, it is obvious that the delay-insensitive asynchronous circuit has a much more uniform temperature distribution due to better spatial and temporal switching activity distribution. Both of these features, i.e., lower average temperature and more uniform thermal distribution, shown to be better in NCL circuits, are highly desirable for 3D ICs.

	Baseline Synchronous Circuit		Delay-insensitive NCL Circuit	
	Layer0	Layer1	Layer0	Layer1
Max Temperature	47.03°C	46.53°C	45.01°C	45.00°C
Min Temperature	45.30°C	45.40°C	45.00°C	45.00°C
Avg Temperature	46.01°C	45.77°C	45.01°C	45.00°C



(a) Synchronous co-processor thermal distribution



(b) NCL co-processor thermal distribution

Fig. 7. Comparison of Thermal Distribution in Layer0

## V. CONCLUSION

This paper presents a comparison of thermal distribution between traditional synchronous circuits and delay-insensitive asynchronous circuits. Our preliminary results from IEEE 32-bit floating-point co-processors show that the delay-insensitive asynchronous circuit exhibits lower average temperature and more uniform thermal distribution, both of which are highly desirable in solving the thermal problem in 3D ICs.

## REFERENCES

[1] G.M. Link, N. Vijaykrishnan, "Thermal Trends in Emerging Technologies," *International Symposium on Quality Electronic Design*, pp. 625-632, 2006.  
 [2] W.L. Hung, G.M. Link, Y. Xie, N. Vijaykrishnan and M.J. Irwin, "Interconnect and Thermal-aware Floorplanning for 3D

Microprocessors," *International Symposium on Quality Electronic Design*, pp. 98-104, 2006.  
 [3] B. Goplen and S. Sapatnekar, "Efficient Thermal Placement of Standard Cells in 3D ICs using Force Directed Approach," *IEEE/ACM International Conference on Computer-Aided Design*, pp. 86-89, 2003.  
 [4] J. Cong, J. Wei, and Y. Zhang. "A Thermal-Driven Floorplanning Algorithm for 3D ICs", *IEEE/ACM International Conference on Computer-Aided Design*, pp. 306-313, 2004.  
 [5] X. Zhou, Y. Xu, Y. Du, Y. Zhang, J. Yang. "Thermal Management for 3D Processors via Task Scheduling", *International Conference on Parallel Processing*, pp. 115-122, 2008.  
 [6] Y. Xie and W.L. Hung. "Temperature-Aware Task Allocation and Scheduling for Embedded Multiprocessor Systems-on-Chip (MPSoC) Design", *Journal of VLSI Signal Processing Systems*, pp. 177-189, 2006.  
 [7] K. M. Fant and S. A. Brandt, "NULL Convention Logic: A Complete and Consistent Logic for Asynchronous Digital Circuit Synthesis," *International Conference on Application Specific Systems, Architectures, and Processors*, pp. 261-273, 1996.  
 [8] S. C. Smith, R. F. DeMara, J. S. Yuan, D. Ferguson, and D. Lamb "Optimization of NULL Convention Self-Timed Circuits," *Integration, the VLSI Journal*, Vol. 37/3, pp. 135-165, 2004.  
 [9] Gerald E. Sobelman and Karl M. Fant, "CMOS Circuit Design of Threshold Gates with Hysteresis," *IEEE International Symposium on Circuits and Systems (II)*, pp. 61-65, 1998.  
 [10] D. E. Muller, "Asynchronous Logics and Application to Information Processing," in *Switching Theory in Space Technology*, Stanford University Press, pp. 289-297, 1963.  
 [11] S. C. Smith, R. F. DeMara, J. S. Yuan, M. Hagedorn, and D. Ferguson, "Delay-Insensitive Gate-Level Pipelining," *Integration, the VLSI Journal*, Vol. 30/2, pp. 103-131, 2001.  
 [12] S. C. Smith, "Completion-Completeness for NULL Convention Digital Circuits Utilizing the Bit-wise Completion Strategy," *The 2003 International Conference on VLSI*, pp. 143-149, June 2003.  
 [13] B. Parhami, "Computer Arithmetic: Algorithms and Hardware Designs," Oxford University Press, New York, 2000.  
 [14] S. K. Bandapati and S. C. Smith, "Design and Characterization of NULL Convention Arithmetic Logic Circuits," *Elsevier's Microelectronic Engineering Journal: Special Issue on VLSI Design and Test*, Vol. 84/2, pp. 280-287, February 2007.