An 8192-Point Fast Fourier Transform 3D-IC Case Study

W. Rhett Davis, *Member, IEEE,* Ambarish M. Sule, and Paul D. Franzon, *Member, IEEE*

Abstract— **3D stacking and integfration can provide system advantages. This paper explores an application driver for 3D ICs. Interconnect-rich applications especially benefit, sometimes up to the equivalent of two technology nodes. Another promising application area is that of logic-on-memory. This paper presents a case studies of an 8192-point Fast Fourier Transform (FFT) processor in order to quantify the benefit of the through-silicon vias in an available 180nm 3D process. The FFT shows a 22% reduction in cycle-time, coupled with an 18% reduction in energy per transform.**

I. Introduction

The combination of the technologies of wafer bonding and through-silicon vias (TSV) promises to enable scaling of electronic system performance beyond that provided just by Moore's law. This paper explores the applications that might benefit from 3D IC design and some of the advances in computer-aided design (CAD) needed to deliver such designs. When is it advantageous to go vertical and when is it not? Stacking two wafers together and integrating them with vertical vias is not cheap. As a rough rule of thumb, the additional processing cost is about equivalent to that of adding two additional layers of metal interconnect. This cost is even higher if individual die are stacked. This cost must be justified through performance gains or cost savings elsewhere in the system.

This cost is much greater than simply that of even "high-end" sophisticated packaging. When might it possibly be justified? Fortunately, there is a growing consensus that there are several, main-stream, circumstances which justify 3D integration.

The most explored advantage of 3D is to use it to reduce the interconnect distance between chip functions. Many researchers justify 3D from an interconnect delay and interconnect power, perspective. From a theoretical viewpoint, the advantages can be substantial. Several studies have presented a Rent's rule style of analysis that presents significant advantages [1,2,3]. The basic argument relies on the fact that with each additional added layer of transistors, there is a similar increase in the number of circuit functions that can be interconnected within a fixed wire length. This leads to a 25% or more decrease in worst case wire length [2,4], a similar decrease in interconnect power [5], and a decrease in chip area. However, experience shows that many designs do not realize this in practice. Fortunately, with careful choice appropriate design applications can be found. For example, FPGAs are very interconnect bound and can achieve substantial performance and power improvements when recast in 3D [3]. Another study of an LDPC-decoder in three tiers [6] showed a power reduction of 60% due to the reduced interconnect lengths and number of repeaters. Results obtained using two practical examples explored at NCSU are summarized in [7], showing that 3D integration can provide about the same performance advantage of two generations of technology scaling – a very compelling case.

Given that many designs do not show improvement from 3D integration, it is important to understand why some do and others don't. This paper presents a case study of a 3D-integrated design that attempts to aggressively improve the speed of an 8K-point Fast Fourier Transform (FFT). This FFT shows a 22% increase in speed when implemented in 3 tiers, which is perhaps the best improvement in speed shown to date for an end-user application in 3D integration.

II. Radix 2/4/8 MDC FFT Architecture

Fast-Fourier-Transform (FFT) structures typically have long wires, to route results from one butterfly circuit to another. Previously in [1], we examined an 8-point FFT processor that showed a mere 2% speedup from 3D integration. In this study, we wanted to pursue a much more aggressive design to see if there would be a more impressive improvement. We chose a fixed-point, 8192-point FFT primarily because it is well examined in the literature [9,10,11] and many high-speed designs are well known. We chose to implement a slight variant of these designs, which we believed would give us the best speed in the 180nm MITLL technology [8]. An aggressive design would likely show the best improvement from 3D-integration, because gate delays would be relatively small compared to the wire delays.

We chose the Multi-path Delay Commutator (MDC) architecture as the basis for our FFT, because it is a well understood approach for pipelining the FFT algorithm and tends to achieve the highest number of transforms per second. We chose it over the single-delay-feedback (SDF) and single-path delay commutator (SDC) architectures, which use less memory, but tend to be slower, because they produce only one output on each cycle.

After settling on the basic MDC architecture, we had to choose the radix of the basic butterfly operations. There are many variants of the Cooley-Tukey radix-2 FFT algorithm described in the literature, each with its own signal flow graph

W. Rhett Davis and Paul D. Franzon are with the Department of Electrical and Computer Engineering (ECE), North Carolina State University (NCSU), Raleigh, NC 27695 USA.

W. Rhett Davis (phone: 919-515-5857; e-mail: rhett_davis@ncsu.edu).

Paul D. Franzon (phone: 919-515-7351; e-mail: paulf@ncsu.edu)

Ambarish M. Sule is now with Qualcomm, Inc. in Cary, NC. (e-mail:amsule@qualcomm.com)

978-1-4244-2167-1/08/$25.00 ©2008 IEEE

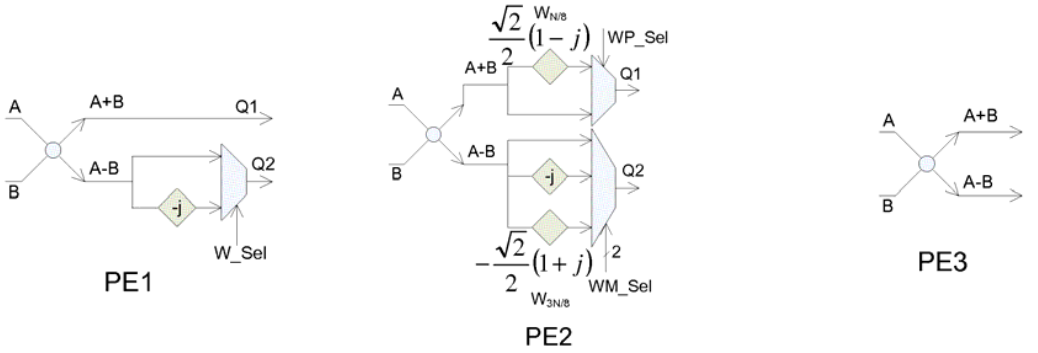

Figure 1: Processing Elements for the Radix-2/4/8 FFT Butterfly

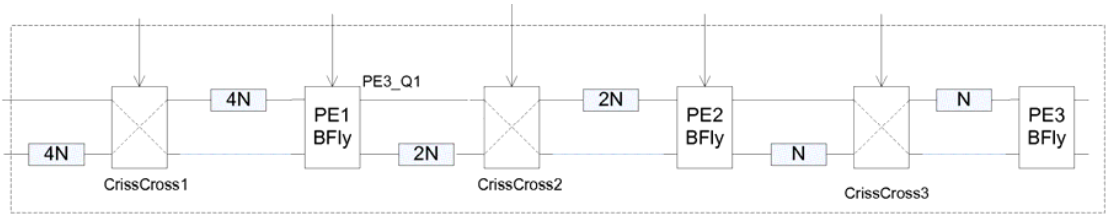

Figure 2: Radix-2/4/8 MDC FFT Butterfly

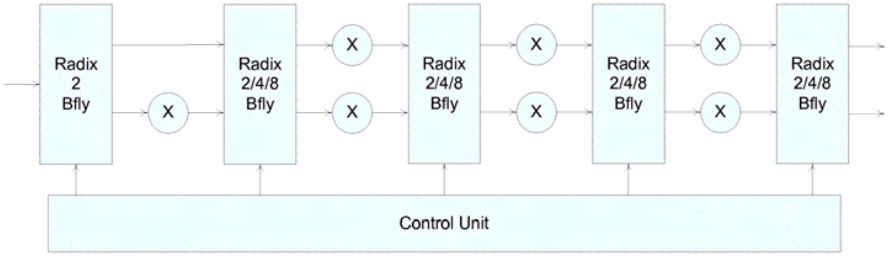

Figure 3: 8192-pt Radix-2/4/8 MDC FFT Design.

that can be mapped into hardware. The radix-2/4/8 algorithm [9], combines the butterflies of many radices into one signal flow-graph, and has been used in several high-performance VLSI implementations [9,11]. We chose this structure, because we believed that it would give us the highest throughput.

The processing elements for the Radix-2/4/8 Butterfly are as shown in Fig. 1. They are modified from the single-delay feedback versions shown in Jia [9] and Lin [12]. Complex multiplication by $\sqrt{2}/2$ can be done using 4 real multiplies and 2 additions. But when using fixed point numbers, real multiplication of a number by 2^N can be calculated by appropriately right-shifting or left-shifting the number N times. This approach can be used to perform complex multiplication by using only 12 additions as shown in [9]. For this constant multiplication, usage of 12 additions does not increase the error of the result. Thus, the Radix-2/4/8 Butterfly can be built without using any multipliers. The complete Radix-2/4/8 Butterfly is as shown in Fig. 2.

Any FFT with number of points divisible by 8 will require a cascade of Radix-2/4/8 Butterflies. For other FFTs, Radix-2 or Radix-2/4 Butterflies can be padded with the cascade of Radix-2/4/8 Butterflies. For this 8192-point design, we used the structure shown in Fig. 3.

The FIFO memories lie inside the radix-2/4/8 butterflies as shown in Fig. 2. The FIFO capacity required is divided by two at each stage of the pipeline, and so the first stage will have two FIFOs of 2K words, followed by 1K words, 512 words and so on until the last stage contains a simple 1-word register of 24 bits. The complex multipliers are the biggest combinational blocks in the design and lie in the critical path. As the entire structure of the design is pipelined, it is functionally very easy to pipeline the complex multipliers and reduce the critical path delay. The complex multiplier we used follows the structure described in [13], which reduces the complexity from 4 real multipliers and two adders to only 3 real multipliers and 5 real adders. The fixed-point VHDL package available from Doulos [14] is used to develop the RTL code for the complex multiplier. As the entire design has a flexible pipeline structure, it is easy to modify the number of pipeline stages if needed. This can be done by adding the appropriate number of registers in the datapath. We add registers at the input and output of every complex multiplier in the design to achieve the highest clock frequency.

III. Physical Design Flow

The design-flow begins by swapping the FIFO memories in the RTL code with code including three banks of memory, using the same approach described in section III. A single-tier

version was created for comparison, using two banks. The 3-tier design is then synthesized and put in to the RTL-to-GDS flow illustrated in Fig. 4. The design is first partitioned into the number of tiers available (3 in our case) using K-Metis. Because K-Metis does not handle the size difference between the standard-cells and memories, the memories are removed from the netlist before partitioning, and added back in afterwards. After partitioning, the top module contains only three sub-modules, corresponding to each tier. No cells are moved between tiers after this point.

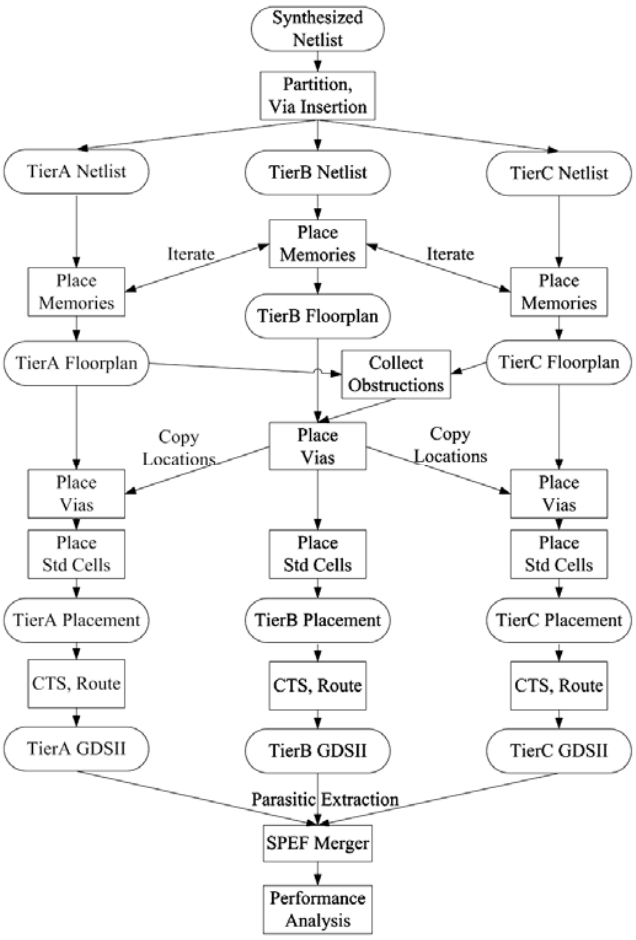

Figure 4. Physical 3D-IC Design Flow.

. These three pseudo-independent designs are then taken through the rest of the 3D physical design flow. An initial unconstrained placement is done individually on each of the tiers using Encounter. The 3 placed designs are then manually studied and modified to get the next iterated placement. Modifications involve manually changing the locations of memories to get a better floorplan. Each bank in the FIFO memories is placed in the same X-Y position. This is accomplished by first placing the memories on tier B and copying these locations to Tiers A and C. After the memories are pre-placed, the TSVs are placed on TierB and again the corresponding locations are copied onto Tiers A and C. The memories already placed in the tiers act as blockages during this placement. Next, the rest of the standard cells are placed individually on each tier. The clock trees are then individually synthesized and the 3 tiers are independently routed.

Next, 3 sets of parasitics are extracted into 3 SPEF files. Also, a netlist is produced for each tier that contains the clock and reset tree buffers. The three netlists along with the three SPEF files are then read into Synopsys Primetime and timing is reported. Similarly, they are read into Synopsys Primepower and power is reported.

Fig. 5 shows the final placed version of Tiers A, B and C. The overall density for TierA is 63%, for TierB is 67% and that for TierC is 66%. Fig. 5 shows the final routed versions of Tiers A, B and C respectively.

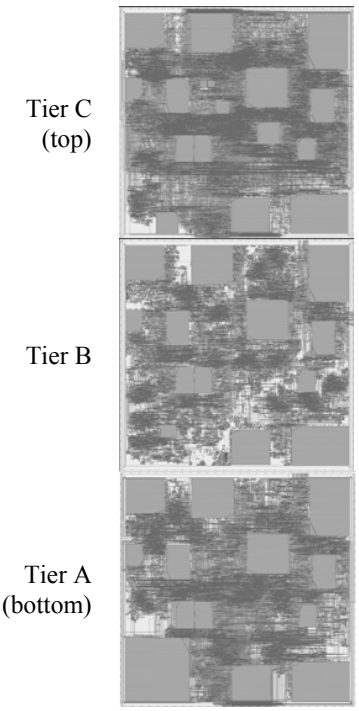

Figure 5. Routed Layout for the 3D FFT Design.

For the single-tier design, the floorplan was made as close as possible to the multi-tier design. The most significant difference was that for every FIFO memory on a particular tier in the multi-tier design, there were 2 sub-banks of memories to be placed in the single-tier design. For each of the FIFO memories, the 2 sub-banks were placed close to each other, in order to reduce the lengths of the wires connecting these two subbanks and the controllers. As in a usual EDA flow, the single-tier placement is followed by the routing and clock tree synthesis steps in that order and the performance of this design is compared with the three-tier approach.

IV. Physical Verification Results

The performance characteristics of the single-tier and 3-tier designs are compared in Table 1. First, we compare the performance of the single-tier design with previously published work to ensure that we are using a sufficiently aggressive basis for comparing single-tier and 3-tier performance. Designs with larger feature sizes are normalized

using the method of constant electric-field scaling and the scaling factors shown. As can be seen from this table, the FFT designed here is 8X-10X faster. Also of note in the table is that the FFT designed here shows an energy-per-transform that is on par with the lowest energy reported in the previous work. Therefore, the energy-delay-product (EDP) is considerably lower for this work. These results satisfy us that the single-tier design is a good basis for comparison.

Comparing the 3-tier and single-tier designs, we see that the area grew considerably, due to the difficulty of partitioning a design with so many large memories. In spite of this area expansion, the cycle-time is an impressive 22% smaller in 3 tiers. This is still far from the ideal 42% ($\sqrt{3}$) improvement that we were aiming for, but to our knowledge, it is the best speed improvement claimed to date for a design study comparing single-tier and multi-tier performance. The improvement comes entirely from the reduced wire-lengths in the three-tier implementation. In addition to the speed improvement, the 3-tier design uses 18% less energy per transform and has a 36% lower EDP. Although these reductions are quite good, they are eclipsed by the 60% reductions in energy and EDP claimed for the LDPC decoder in [6], which was implemented in the same process.

Detailed thermal analysis was not performed on this design, because the total power density was around 0.1 W/cm^2, which tends to produce a temperature rise of less that 10 degrees C above the heat-sink. Thermal analysis becomes important for systems in which the power density exceeds 1 W/cm^2.

V. Conclusions

This paper has shown a case study of an 8K-point Fast Fourier Transform in a 3D process with through-silicon vias in order to help quantify their benefit. The design achieves a 23% increase in speed when implemented in 3 tiers, which is perhaps the best improvement in speed shown to date for an end-user application in 3D integration.

Although these improvements are impressive, they are still not enough to motivate the use of TSVs. Although the semiconductor industry is moving toward 3D integration, the fabrication of TSVs is costly. If speed and power improvements of 20%-25% are the best that can be achieved with the use of TSVs, then we may find that companies will simply stack die and route signals to the chip periphery. The performance improvement afforded by TSVs is useful only for the highest performance designs, in which memory latency is usually the bottleneck. It is for these reasons that we believe logic-on-memory applications have the best chance of being the first commercial designs to use TSVs.

Acknowledgments

This project was funded by DARPA under contract FA8650-04-C-7127, managed by AFRL. We would like to thank Cadence and Synopsys for generously providing the CAD tools for this work. Thanks to MIT Lincoln Labs for providing access to their FD-SOI technology and for their aid in developing our design kit. Lastly, thanks to James Stine at Oklahoma State University generously providing access to the OSU-SoC standard-cell characterization scripts, which provided the basis for our library.

References

[1] W. Davis, *et. al.*, "Demystifying 3D ICs: The Pros and Cons of Going Veritical," *IEEE D&TOC*, Nov.-Dec. 2005.

[2] K. Banerjee, S. Souri, P. Kapur, K. Saraswat, "3-D ICs: A Novel Chip Design For Improving Deep-Submicrometer Interconnect Performance and Systems-on-Chip Integration," *Proc. IEEE*, May 2001.

[3] A. Rahman, S. Das, A. Chandrakasan, R. Reif, "Wiring Requirement and Three-Dimensional Integration Technology for Field Programmable Gate Arrays," *IEEE Trans. VLSI Sys.*, Feb. 2003.

[4] A. Rahman, . Fan, R. Reif, "Comparison of Key Performance Metrics in Two and Three Dimensional Interrgated Circuits," *IITC,* 2000.

[5] S. Das, A. Chandrakasan, R. Reif, "Timing, Energy and Thermal Performance of Three Dimensional Integratied Circuits," *GLS-VLSI*, Apr. 2004.

[6] L. Zhou, *et. al.*, "Implementing a 2-Gbps 1024-bit ½-rate Low-Density Parity-Check Code Decoder in Three-Dimensional Integrated Circuits," *ICCD*, Oct. 2007.

[7] H. Hua, "Design and Verification Methodology for Complex Three-Dimensional Digital Integrated Circuits," *Ph.D. Dissertation*, NC State University, 2006.

[8] J. Burns, *et. al.*, "Three-dimensional integrated circuits for low-power, high-bandwidth systems on a chip," *ISSCC*, Feb. 2001.

[9] L. Jia, Y. Gao, J. Isoaho, and H. Tenhunen, "A new VLSI oriented FFT algorithm and implementation," *Intl. ASIC Conference*, Sep. 1998.

[10] E. Bidet, D. Castelain, C. Joanblanq, and P. Senn, "A fast single-chip implementation of 8192 complex point FFT," *JSSC*, Mar. 1995.

[11] Y.W. Lin, H.Y. Liu, and C.Y. Lee, "A dynamic scaling FFT processor for DVB-t applications," *JSSC*, Nov. 2004.

[12] Y. T. Lin, P. Y. Tsai, and T. D. Chiueh, "Low-power variable-length fast Fourier transform processor," *IEE Proc. Comput. Digit. Tech.*, Jul. 2005.

[13] Y.H. Hu, "The quantization effects of the CORDIC algorithm," *IEEE Trans. Signal Processing*, Apr. 1992.

[14] VHDL Fixed Point Arithmatic Package, available online at http://www.doulos.com/knowhow/vhdl_designers_guide/models/fp_arith.

Table 1: Comparison of Single-Tier and 3-Tier FFT Processors with Other Published 8192-point Implementations.

	Jia [45]	Bidet [56]	Lin [56]	Single-Tier	3-Tier	change
Word-length	12	12	11	12	12	
Radix	2/4/8	4	2/4/8	2/4/8	2/4/8	
Process (μm)	0.6	0.5	0.18	0.18	0.18	
Scaling Factor	3.33	2.77	1	1	1	
Voltage	3.3	3.3	1.8	1.5	1.5	
Area (Normalized) (mm^2)	107 (9.63)	100 (13.0)	4.84	8.17	10.6	+30%
Clock Frequency (Normalized) (MHz)	20 (66.6)	20 (55.5)	20	167	214	+28%
Power (Normalized) (mW)	650 (58.5)	600 (77.7)	25.2	385	404	+5%
Exec. Time (Normalized) (μs)	400 (120)	400 (144)	717	24.4	19.0	-22%
Energy/Transform (Normalized) (μJ)	260 (7)	240 (11.2)	18.1	9.39	7.68	-18%
EDP (Normalized) (pJ-s)	104 (0.84)	96 (1.61)	13.0	0.23	0.15	-36%