A 3-D Split Manufacturing Approach to Trustworthy System Development

Jonathan Valamehr, *Student Member, IEEE,*
Timothy Sherwood, *Member, IEEE,*
Ryan Kastner, *Member, IEEE,*
David Marangoni-Simonsen, *Student Member, IEEE,*
Ted Huffmire, *Member, IEEE,*
Cynthia Irvine, *Member, IEEE,* and
Timothy Levin, *Member, IEEE*

***Abstract*—Securing the supply chain of integrated circuits is of utmost importance to computer security. In addition to counterfeit microelectronics, the theft or malicious modification of designs in the foundry can result in catastrophic damage to critical systems and large projects. In this letter, we describe a 3-D architecture that splits a design into two separate tiers: one tier that contains critical security functions is manufactured in a trusted foundry; another tier is manufactured in an unsecured foundry. We argue that a split manufacturing approach to hardware trust based on 3-D integration is viable and provides several advantages over other approaches.**

***Index Terms*—Advanced technologies, cryptographic controls, hardware, integrated circuits, physical security, security and privacy protection.**

I. Introduction

Security is an essential design goal in computer architecture, which must be addressed throughout the lifecycle of a system. The process of designing hardware requires trusting intellectual property (IP) cores and computer-aided design tools developed by third parties, as well as the fabrication and packaging of the final system. Sensitive IP is vulnerable to theft and modification during tape-out, even if a perfect design free of security flaws is sent to the foundry (a trusted foundry does not solve the problem of flawed designs).

Supply chain trust is a global issue. Many countries are concerned about the theft and malicious modification of sensitive designs of chips used in mission-critical systems. In response to this trend, governments have established trusted foundry programs. A trusted foundry is certified as a trusted environment able to produce leading-edge integrated circuits from trusted sources. While the availability of trusted foundries is beneficial, lower-priority projects may not have access to the trusted foundry, and the cost per unit may be higher.

We propose the use of 3-D integration to allow designers of trustworthy systems to leverage the capabilities of commodity foundries. In our approach, a design is split into two tiers: the computation plane and the control plane. The computation plane houses a high-performance processor and is manufactured in an unsecured foundry. The control plane contains critical security functions and is manufactured in a trusted foundry. The control plane is optional, and including it is a foundry-level configuration choice. This tier is a separate plane of circuitry stacked on the top of the computation plane, and the two tiers are joined with vertical interconnect. The decision to include or omit the control plane does not affect the function, performance, or cost of the computation plane. In addition to the benefits of split manufacturing, our technique provides a financial solution for system builders who wish to add security features to cutting-edge processors.

Contributions: In this letter, we show that a control plane, a custom die dedicated to security, has the potential to implement a variety of security functions in a cost-effective and computationally efficient way when joined to a computation plane using 3-D integration. Our approach uses circuit-level primitives for accessing signals on the computation plane so that they can be tapped, disabled, rerouted, or overridden to integrate with the control plane in a purely optional and minimally intrusive manner. We also extend our preliminary work [1] to incorporate: 1) refinements to the circuit-level primitives that support our approach; 2) new 3-D systems that we have designed to evaluate our approach; and 3) an argument comparing split manufacturing based on 3-D integration with other approaches to split manufacturing.

II. Motivation for 3-D Security

With 3-D integration, two integrated circuits are fused together to form a single chip, as shown in Fig. 1. The two dies are connected with through-silicon vias (TSVs) or face-to-face vias, depending on whether a face-to-back or face-to-face bonding process is used. The ability to connect multiple dies allows an optional die dedicated to security functions (the control plane) to be joined with a commodity processor die (the computation plane). The control plane has direct access to the internal signals of the computation plane, benefiting customers requiring application-specific security policy enforcement, information flow control, or other security-specific support.

While 3-D integration is an emerging technology, many 3-D systems have successfully overcome the initial challenges of 3-D technology, including thermal, testing, and yield. For

Manuscript received March 18, 2012; revised August 1, 2012; accepted October 22, 2012. Date of current version March 15, 2013. This work was supported in part by the National Science Foundation under Grant CNS-0910734, Grant CNS-0910389, and Grant CNS-0910581. A longer version of this transactions brief will be published as Naval Postgraduate School Technical Report NPS-CS-12-004. This paper was recommended by Associate Editor G. Loh.

J. Valamehr is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: valamehr@ece.ucsb.edu).

T. Sherwood is with the Department of Computer Science, University of California, Santa Barbara, CA 93106 USA (e-mail: sherwood@cs.ucsb.edu).

R. Kastner is with the Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: kastner@cs.ucsd.edu).

D. Marangoni-Simonsen is with the Department of Electrical Engineering, Harvey Mudd College, Claremont, CA 91711 USA (e-mail: dmarangonisimonsen@gmail.com).

T. Huffmire, C. Irvine, and T. Levin are with the Department of Computer Science, Naval Postgraduate School, Monterey, CA 93943 USA (e-mail: tdhuffmi@nps.edu; irvine@nps.edu; levin@nps.edu).

The views expressed in this paper do not represent the official policy of the United States Government, the Department of Defense, or the National Science Foundation.

Digital Object Identifier 10.1109/TCAD.2012.2227257

0278-0070/$31.00 © 2013 IEEE

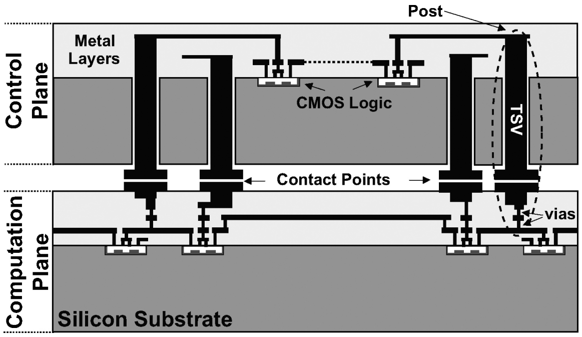

Fig. 1. Two-tier 3-D integrated circuit in which the control plane (above) and computation plane (below) are joined using a face-to-back bonding process. Face-to-face bonding is also possible.

example, Toshiba has applied 3-D integration to a CMOS image sensor camera module for mobile phones, achieving a 55% reduction in volume and a 36% reduction in footprint while satisfying high-speed I/O requirements for videos, with the required data rate per pin as high as 130 MB/s for VGA at 30 f/s and 650 MB/s for 3.2 Mpixel at 15 f/s [2]. Kim *et al.* joined a tier containing 64 CPU cores running at 277 MHz with a tier containing 256K of SRAM; their system achieves a 63.8 GB/s memory bandwidth [3]. Loh *et al.* provided detailed analysis of the advantages of 3-D integration for implementing a cache with their analysis showing significant reductions in access latency and energy per access. They also devised a 3-D floor plan for the Intel Pentium 4, removing critical paths and improving performance and power by 15% [4]. Loh *et al.* also showed that 3-D integration can improve clock frequency by 10.3% for the Alpha 21364, and that 3-D integration can expose instruction-level parallelism for the Alpha 21264, improving performance by 9.7% [4]. Loh *et al.* also showed that a 3-D version of a dynamic non-uniform cache architecture reduces average L2 access time by 50%; they also show that 3-D stacking can allow the cache size to increase, reducing average memory access latency by 13% and reducing off-chip bandwidth by 3x [4]. Using the Intel Core 2 Duo as a baseline, Black *et al.* showed that a 3-D stacked DRAM cache can reduce the cycles per memory access by 13% on average and as much as 55% while reducing off-chip bandwidth and power by 66% [5]. Loh proposes optimizations to 3-D DRAM that result in 1.75x speedup over prior 3D-DRAM approaches, and Loh also proposes a L2 miss handling architecture that achieves an extra 17.8% performance improvement [6]. Puttaswamy and Loh showed that a 3-D-partitioned cache can reduce latency by 21.5%, reduce energy consumption by 30.9%, and increase IPC by 12% [7].

Several approaches to split manufacturing are possible. First (option 0), a CPU can be fabricated in an unsecured foundry, and software implementing security functions can be loaded for execution onto the CPU in a secure facility. In general, implementing security functions in software is less costly than in hardware, but software implementations have worse performance and greater power consumption, and are more susceptible to tampering. Next (option 1), both the processor and hardware security functions reside on the same 2-D chip, which is manufactured in a trusted foundry (this is a base case that does not allow for split manufacturing). Next (option 2), a coprocessor implementing security functions can be manufactured in a trusted foundry, and it resides on the same circuit board as a main processor that is manufactured in an unsecured foundry. Finally (option 3), one tier is made in a trusted foundry, the other tier is made in an untrusted foundry, and the two are joined in a trusted facility using 3-D integration.

Based on the area, power, and performance figures from the literature presented above, we extrapolate that option 2 (using a separate processor and coprocessor connected at the circuit board level) will consume more power than option 1 (implementing everything on the same processor), and option 3 (using 3-D) will consume the least power, in general. Of course, individual designs may vary (some may not benefit from 3-D integration), but in general, option 3 will also have the least delay and greatest bandwidth, followed by option 1 and then option 2, which has the greatest delay and the least bandwidth due to the use of slow, power-hungry off-chip buses.

A variant of option 1 was recently proposed by the U.S. Intelligence Advanced Research Projects Agency (IARPA), in which an unsecured foundry, called the front-end-of-line (FEOL), manufactures a layer of transistor devices and then sends the unfinished wafer to a trusted foundry, called the back-end-of-line (BEOL), which adds metal layers that connect the devices to form useful circuits [8]. The interface between the FEOL and BEOL circuits in the IARPA program is different from the interface between the tiers in option 3, which allows one tier to monitor, disable, reroute, and override signals in another tier. Also, while the transition between the FEOL and BEOL foundries is an open research challenge for the IARPA program, joining separately made dies is already proven technology with 3-D integration.

A variant of option 0 uses reconfigurable hardware; an FPGA is made in an unsecured foundry, and a design is loaded onto the FPGA in a trusted facility. While an FPGA may achieve higher throughput than a CPU, the design is protected by bit-stream encryption, which can be thwarted by side-channel attacks on the bit-stream decryption mechanism. Anti-fuse FPGAs can help mitigate this problem, but it is a write-once technology, unlike SRAM-based FPGAs.

We consider the trust issues of split manufacturing in [9] and establish that our threat model includes unintentional hardware design flaws and malicious software in the computation plane; the threats of hardware Trojans (i.e., malicious inclusions), physical tampering/probing, simple/differential power analysis, and compromising RF/acoustic/photonic emanations are outside the scope of our work. A first-order concern is whether the output of the unsecured foundry needs to be independently trustworthy for the joined system to provide certain trustworthy functions. If so, it would seem to obviate the purpose of the effort. For option 3, we found that the independence of a control plane from interference by the computation plane through active corruption of the processing or passively, via withholding of services is a primary requirement for trustworthy behavior of the control plane [9]. However, the control plane can often choose whether to establish dependencies on the computation plane. The detection of malicious inclusions on the computation plane is another security feature that can be hosted on the control plane, although this technology is very immature. We believe it is not yet possible to add a layer of hardware to a computation plane that is riddled with malicious inclusions—effectively bearing an unknown degree of resemblance to its design—in the hopes that the composition of the two layers will be highly trustworthy. Nevertheless, provided that the requirements of self-protection and dependency layering are met for the control plane, it is possible to offer an alternate service to the computation plane, to actively override the computation plane for enforcement of policies, and to passively monitor the computation plane with high integrity [9]. Note that while option 3 allows a control plane to access the internal signals of a computation plane, this is not possible via the circuit board level coprocessor interface associated with option 2. Therefore, while a coprocessor can provide an alternate service such as encryption to a main processor, actively overriding or passively monitoring internal signals is impossible with option 2.

While the threats of malicious hardware, physical tampering, power analysis, and compromising emissions are outside the scope of this letter (we are assuming that the 3-D integrated circuit (IC) lacks countermeasures against these threats), we do not believe that 3-D integration necessarily increases the risk of a physical probing attack on sensitive signals carried by the inter-die vias. Probing for the purposes of testing is much harder for 3-D than for 2-D, due to the difficulty of probing an individual TSV and the risk of breaking a TSV. Furthermore, the difficulty of chemically removing the package of a 3DIC and separating the bonded layers is significantly greater than the challenge of probing a circuit board connecting a processor and coprocessor (option 2). For example, a 3-D cryptographic coprocessor's tiers are tightly bonded and have no exposed shared buses or I/O pins, and the inter-die vias are enclosed in a package and only accessible by removing the package and separating the tight bond between the tiers [10]. However, chemical–mechanical planarization (i.e., sanding) of a tier is available to professional attackers, and future work is needed to develop secure protocols for the inter-die interface of option 3.

Finally, comparing option 1 and option 3, we note that 3-D integration offers the potential to offload logic to the control plane. We argue that many security applications can benefit from this capability. For example, Tiwari *et al.* [11] developed an information flow tracking method that increases area by 70% over the base processor's area. Additional area allows for the implementation of additional cipher implementations [10], as well as real-time monitoring and processing of programs in execution [12].

III. 3-D Security Architecture

The control plane can include several security functions on one die, implemented as either passive or active monitors. A passive monitor accesses and analyzes data from the computation plane, e.g., memory accesses or instructions. Monitoring these events requires tapping some of the wires in the processor. Whereas passive monitoring allows for auditing, anomaly detection, and the identification of suspicious activities, systems enforcing security policies often require strong guarantees about restrictions to these types of behavior. A novel contribution of our work is the employment of active monitors, e.g., to control information flow between cores, to arbitrate communication, and to partition resources.

The key ability needed to support such functionality is to reroute signals to the control plane and then override them with potentially modified signals. With this technology, we can force all communication, memory accesses, and shared signals to travel to the control plane, where they are subject to both examination and control. For instance, we can ensure that confidential data being sent between two cores, which are traditionally forced to traverse a shared on-chip bus, is not leaked to a third party with access to that bus.

We have developed a method to modify signals on the computation plane that is accomplished in two parts when the control plane is connected. The first part is to ensure that the monitor has unfettered access to the signal (tapping), which is the same as the passive monitoring scenario described above. The second part is to disable the signal, preventing it from propagating (e.g., via a bus). The difficulty is that we must remove a capability, the connection between two components on the computation plane, only by adding a control plane. The computation plane must be fully functional without an attached control plane, yet it needs to be constructed so that by connecting circuitry, the targeted capability can be achieved. To accomplish this, components in the computation plane must be modified to support active monitoring.

Our preliminary work [1] introduces the circuit-level modifications needed for the control plane to perform its intended function and for the computation plane to function in its absence. The primitives each provide an environment for receiving one or two inter-die vias. We refer to this computation plane environment as a TSV receptacle or socket.

Tapping: This can be used to pull specific signals to the control plane without interrupting their original path. This is particularly useful when performing analysis (e.g., dynamic information flow tracking) of the flow of information on the computation plane without affecting its original functionality.

Disabling: This allows us to completely stop the flow of data on a specific signal line. Uses of disabling include the ability to isolate a specific resource from unintended accesses, or enforcement of policies that require tight guarantees on the integrity of data on a shared bus.

Overriding: This allows us to block the intended value of a signal and modify it to a value determined by the security layer. For some security applications, critical control signals need to be changed in order to adhere to a security policy that is being enforced by the control plane.

Rerouting: This combines tapping and disabling to send signals to the control plane and block their transmission to the original path. Rerouting can be used in situations where we want to create new controlled buses between resources on the computation plane. Rerouting also allows the use of a signal for a different purpose than originally intended. Once on the control plane, the signal can be analyzed and combined with other data from the control or computation planes, or simply

TABLE I
AREA OF GENERIC TSV RECEPTACLE IN 90-NM TECHNOLOGY NODE

	1 Generic TSV Receptacle	**128 TSV Receptacles**	**5-Stage MIPS Processor**
Area (library area units)	84.1	10764.8	240 000

stored for later use. This can then be coupled with overriding to change control or data outputs on the computation plane based on new logic in the control plane.

Diode: Diodes allow information to flow in only one direction; note that one-way communication can be enforced using various electrical techniques besides a diode, such as a buffer [13]. Such an arrangement can enforce a policy requiring that information flow from a low confidentiality component to a high confidentiality component but not vice versa.

Generic TSV receptacle: This can be used to support multiple control plane applications with the same computation plane, e.g., when different control planes are used or when the applications on a given control plane are reconfigured [13]. A given TSV receptacle can be used for different purposes from application to application. A generic TSV receptacle provides design flexibility at the cost of additional circuitry and posts. In order to explore the area ramifications of incorporating these Generic TSV Receptacles on the computation plane, we synthesized one Generic TSV Receptacle and compared it to a simple five-stage pipelined MIPS processor. While it is infeasible to build sockets for every signal, good candidates include registers, control signals, and shared buses; engineers must strike a balance between the generality of the interface and its performance and cost.

The Generic TSV Receptacle and MIPS processor were written in Verilog and synthesized using Synopsys Design Compiler in 90-nm technology. The area of the Generic TSV Receptacle, as shown in Table I, is 84.1 AU. This is a very small percentage of the area of the full processor. Even if we needed 128 Generic TSV Receptacles, the additional area added to the MIPS processor is about 4.5%, which is relatively small. This percentage would be even less when adding Generic TSV Receptacles to a large, modern commodity processor that includes structures such as caches, advanced branch predictors, and Floating Point Units.

Assured generic TSV receptacle: The Generic TSV Receptacle could include diodes to assure that the information flows of each post are precisely controlled, e.g., to prevent back flow from the computation plane.

IV. 3-D APPLICATIONS

A. Categories of 3-D Applications

1) *Isolation and Protection:* Isolation of active resources is one potential application of our circuit-level primitives. For example, in multi-core processors there are shared data and address buses that rely on a mutually trusting shared bus protocol, where each core is responsible for its own arbitration. This is problematical for the security of bus traffic on a system running code of varying trust levels on each core. Instead, we could use disabling to disconnect a core from the bus for any given amount of time, creating a time division multiple access protocol between the cores and the shared resources of interest.

2) *System Analysis and Monitoring:* It is often useful to monitor the activity of the computation plane for auditing, intrusion detection, or post-mortem analysis. Information flow tracking in the control plane, for example, attempts to identify, track, mitigate, and deter the execution of malicious code. We can use tapping to read signals of interest on the computation plane and overriding to optionally modify an exception signal without tampering with normal use.

3) *Secure Alternate Service:* Another potential application is augmenting the computation plane with additional security functions. For systems requiring high-bandwidth cryptography, a cryptographic engine on the control plane [10] can accept cryptographic instructions being executed on the computation plane, performing the operation immediately before sending the result back to the execution pipeline. This is achieved by using rerouting to extract the cryptographic instructions from the standard execution pipeline and execute them, and using overriding to inject the result into the pipeline as if it were part of the normal execution flow. 3-D integration allows the addition of any cipher implementation to be included in the system as a foundry-level option.

B. A 3-D Cache Monitor

In our preliminary work [1], we developed a custom architecture, implemented in the control plane, for eliminating access-driven cache side channel attacks. Concurrent processing platforms present several security issues; although these architectures provide increased performance through instruction-level parallelism, their methods of resource sharing leave them vulnerable to side channel attacks. In our architecture, the control plane maintains a cache protection structure that indicates, for each cache line, whether it is protected, and if so, for which process. When a different process loads or stores data related to a protected cache line, no eviction will occur, and the data is not cached unless an alternate line is available in the cache protocol being used. The cache protection structure on the control plane stores security bits, representing locks on shared cache entries. With this in place, when instructions proceed to load or store data, these security bits are first checked to determine whether to grant a cache eviction. When the control plane is not attached, the cache functions as normal. However, when the control plane is added, we can avoid undesirable cache evictions.

As a proof of concept, we have developed a synthesizeable version of our security mechanism in Verilog. We designed the security mechanism as a separate module that is interfaced with a simple four-way set associative cache. We synthesized both modules using Altera Quartus, targeting the Stratix II FPGA with the compiler set to optimize for performance, and have verified that the design is functional, easily scaled, and can be implemented with low overhead. We found that the 3-D cache eviction monitor does not increase the critical path of the circuit, and we observe that this type of cache-line locking produces very little performance degradation for many programs. Our experiments show that the critical path resides in the computation plane, and adding the control plane does not affect the cycle time of the joined circuit. However, our experiments do not take into account the delay of the vertical posts between the computation plane and the control plane. Loi *et al.* [14] characterized the worst-case delay of a 3-D bus that travels from one corner of a chip to the opposite corner of a 3-D layer above, and they found this delay to be about 0.29 ns. Even with the addition of this worst-case bus delay to the 3-D cache eviction monitor's critical path, the new critical path is still less than that of the cache/cache controller, further confirming that the addition of the 3-D cache eviction monitor will have minimal effect on the performance of the cache.

C. Security Levels

We use a MIPS CPU designed in Verilog as the computation plane circuit to construct a system that assigns a two-bit tag, representing one of four possible security levels (e.g., TS, S, C, and U), to every address in memory. The tags are stored in a dedicated region of memory. We attach a control plane that acts as a regulator, operating in parallel with the MIPS CPU. DIP switches on the Xilinx Virtex-V FPGA development board are used to set the security level of a process that is executing a small program consisting of approximately ten MIPS instructions. The regulator will prevent the process from executing instructions with a higher security level. For example, if a process at a lower level branches to an instruction with a higher label, the regulator allows the branch but skips over all instructions at the higher level until the program counter reaches an instruction with a label that is equal to or less than the level of the process. That is, the MIPS CPU must wait until the next available instruction that is at or below its level. The regulator accomplishes its policy enforcement duties by performing shadow computations on the tags in parallel with the actual computation being performed on the MIPS CPU.

D. Cryptographic Coprocessor

We use the MIPS CPU described in Section IV-C to construct a system that uses a cryptographic coprocessor in the control plane to provide memory encryption and decryption to the CPU in the computation plane [10]. This system encrypts incoming data writes and decrypts outgoing data reads. It intercepts writes to memory and overrides them with encrypted data. The cryptographic core monitors data writes and reads, checking the write address against a predetermined list to determine whether it is an instruction. If it is not an instruction, the system encrypts the data into memory. Also, the output data is only decrypted if the incoming read address corresponds to data. The cryptographic coprocessor is a stripped-down AES core from the Open Cores website. We evaluated the combined circuit, running a small program, on a Xilinx ML501 board with a Virtex-V LX50 FPGA.

V. Conclusion

3-D integration is a promising approach to split manufacturing. We described the circuit-level primitives to enable passive and active monitoring of the computation plane by adding a control plane. We also described several 3-D systems that apply our primitives. Future work will involve tape-out of a prototype 3-D IC and development of novel secure applications that leverage 3-D integration.

Acknowledgment

The authors wish to thank the anonymous reviewers for their insightful comments.

References

[1] J. Valamehr, M. Tiwari, T. Sherwood, R. Kastner, T. Huffmire, C. Irvine, and T. Levin, "Hardware assistance for trustworthy systems through 3-D integration," in *Proc. ACSAC*, Dec. 2010, pp. 199–210.

[2] H. Yoshikawa, A. Kawasaki, T. Iizuka, Y. Nishimura, K. Tanida, K. Akiyama, M. Sekiguchi, M. Matsuo, S. Fukuchi, and K. Takahashi, "Chip scale camera module (CSCM) using through-silicon-via (TSV)," in *Proc. ISSCC*, Feb. 2009, pp. 476–477, 477a.

[3] D. H. Kim, K. Athikulwongse, M. B. Healy, M. Hossain, M. Jung, I. Khorosh, G. Kumar, Y.-J. Lee, D. Lewis, T.-W. Lin, C. Liu, S. Panth, M. Pathak, M. Ren, G. Shen, T. Song, D. H. Woo, X. Zhao, J. Kim, H. Choi, G. Loh, H.-H. Lee, and S. K. Lim, "3D-MAPS: 3-D massively parallel processor with stacked memory," in *Proc. IEEE ISSCC*, Feb. 2012, pp. 188–190.

[4] G. H. Loh, Y. Xie, and B. Black, "Processor design in 3-D die-stacking technologies," *IEEE Micro*, vol. 27, no. 3, pp. 31–48, May–Jun. 2007.

[5] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCaule, P. Morrow, D. W. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen, and C. Webb, "Die stacking (3-D) microarchitecture," in *Proc. 39th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Dec. 2006, pp. 469–479.

[6] G. H. Loh, "3-D stacked memory architectures for multi-core processors," in *Proc. ISCA*, Jun. 2008, pp. 453–464.

[7] K. Puttaswamy and G. H. Loh, "Implementing caches in a 3-D technology for high performance processors," in *Proc. IEEE ICCD*, Oct. 2006, pp. 525–532.

[8] *Trusted Integrated Chips (TIC) Program Broad Agency Announcement 11-09*. (2011, Oct.) [Online]. Available: http://www.iarpa.gov//solicitations_tic.html

[9] T. Huffmire, T. Levin, M. Bilzor, C. E. Irvine, J. Valamehr, M. Tiwari, T. Sherwood, and R. Kastner, "Hardware trust implications of 3-D integration," in *Proc. 5th Workshop Embedded Syst. Security*, Oct. 2010, pp. 1–10.

[10] J. Valamehr, T. Huffmire, C. Irvine, R. Kastner, C. K. Koc, T. Levin, and T. Sherwood, "A qualitative security analysis of a new class of 3-D integrated crypto co-processors," in *Proc. Festschrift Jean-Jacques Quisquarter*, LNCS 6805. Mar. 2012, pp. 364–382.

[11] M. Tiwari, H. Wassel, B. Mazloom, S. Mysore, F. Chong, and T. Sherwood, "Complete information flow tracking from the gates up," in *Proc. Int. Conf. ASPLOS*, Oct. 2006, pp. 109–120.

[12] S. Mysore, B. Agrawal, S. Lin, N. Srivastava, K. Banerjee, and T. Sherwood, "Introspective 3-D chips," in *Proc. 12th Int. Conf. ASPLOS*, Oct. 2006, pp. 264–273.

[13] T. Huffmire, T. Levin, C. Irvine, R. Kastner, and T. Sherwood, "3-D extensions for trustworthy systems," in *Proc. Int. Conf. ERSA*, Jul. 2011, pp. 45–54.

[14] G. L. Loi, B. Agrawal, N. Srivastava, S.-C. Lin, T. Sherwood, and K. Banerjee, "A thermally-aware performance analysis of vertically integrated (3-D) processor-memory hierarchy," in *Proc. 43rd DAC*, Jul. 2006, pp. 991–996.