3-D Extensions for Trustworthy Systems

(Invited Paper)

Ted Huffmire*, Timothy Levin*, Cynthia Irvine*, Ryan Kastner† and Timothy Sherwood‡
*Department of Computer Science
Naval Postgraduate School, Monterey, CA 93943
Email: {tdhuffmi,levin,irvine}@nps.edu
†Department of Computer Science and Engineering
University of California, San Diego, La Jolla, CA 92903
Email: kastner@cs.ucsd.edu
‡Department of Computer Science
University of California, Santa Barbara, Santa Barbara, CA 93106
Email: sherwood@cs.ucsb.edu

Abstract

Trustworthy system development entails a high non-recurring engineering (NRE) cost together with a low volume of units over which to amortize that cost. For example, the potential for developmental and operational attacks against hardware requires countermeasures that make it very expensive to design and manufacture custom hardware used to build high assurance systems. To address these problems, we propose an approach to trustworthy system development based on 3-D integration, an emerging chip fabrication technique in which two or more integrated circuit dies are fabricated individually and then combined into a single stack using vertical conductive posts. With 3-D integration, a general-purpose die, or computation plane, can be combined with a special-purpose die, or control plane. We discuss the security advantages of using 3-D integrated hardware in sensitive applications, where security is of the utmost importance, and we outline problems, challenges, attacks, solutions, and topics for future research.

I. Introduction

Hardware-oriented security is growing in importance as attackers increasingly target the lowest level of system abstraction. 3-D integration is an emerging technology for designing efficient chips by stacking two or more integrated circuit (IC) dies and connecting them with conductive posts. Unlike traditional coprocessors, a 3-D integration design approach offers the ability to monitor and even override internal structures of a processor. For example, on-chip bus traffic can be monitored, and bus connections can be disabled. With these capabilities, 3-D integration can be used to provide secure alternate services (e.g., cryptographic processing at much higher bandwidth than a coprocessor), isolation, and passive monitoring for mass-produced processors.

In our basic paradigm, a 3-D chip consists of one die that is a commodity microprocessor and another die that contains application-specific security functionality; we refer to the commodity die as the *computation plane* and the custom die as the *control plane* (or *resource*[1] plane in situations where resources are simply made available). Figure 1 shows an example system in which multiple CPU cores reside in the computation plane, and routers reside in the control plane. The control plane and computation plane can be fabricated at separate foundries and conjoined in a third facility. When the control plane contains mechanisms that enforce a policy on the computation plane, to achieve a requisite level of trust, the fabrication of the control plane and the conjoining operation can take place in a trusted foundry.

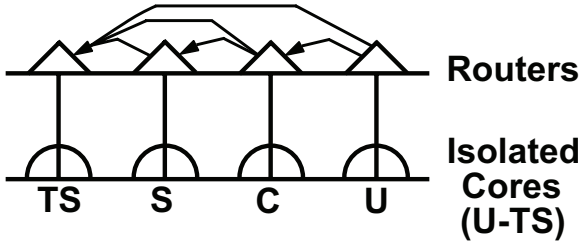

Fig. 1. Isolated cores, surrounded by 2-D moats, and network-on-chip routers. Here, using two IC planes, the routers help to enforce a Multilevel Security (MLS) policy on information flow in the lower plane. In this example, each core has been assigned a label of either TOP SECRET (TS), SECRET (S), CONFIDENTIAL (C), or UNCLASSIFIED (U).

Developing high assurance systems is costly. Our approach has the potential to reduce the cost of developing hardware for high assurance systems by joining a mass-produced computation plane with a custom control plane. Our approach provides several advantages, including (1) dual use of the computation plane, which can be optionally combined with a control plane housing application-specific security functions; (2) physical isolation and logical disentanglement of security functions in the control plane from the non-security circuitry in the computation plane; (3) controlled lineage (e.g., use of a trusted foundry to manufacture the control plane); (4)

[1]Resources are the totality of all active and passive entities on a chip, across a wide range of abstractions. For example, a storage buffer, an accumulator, and a bus are resources.

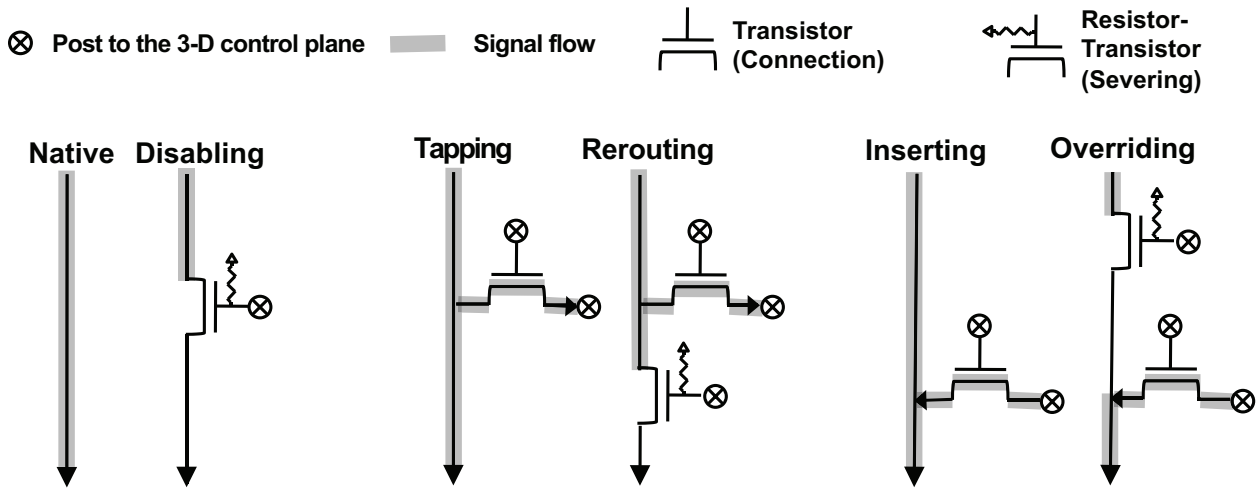

Fig. 2. Circuit-level primitives for trustworthy 3-D design [28]. The *disabling* circuit can stop a signal in the computation plane from flowing, based on the control plane's command, which is sent through a dedicated post. The *tapping* circuit copies a signal from the computation plane to the control plane. Two posts are needed: one to carry the signal to the control plane and another for the command to connect the signal. The *rerouting* circuit combines tapping and disabling so that the original signal only goes to the control plane. The *inserting* circuit carries a signal from the control plane to a circuit on the computation plane. The *overriding* circuit combines inserting and disabling, first disabling the original signal in the computation plane and then introducing a new signal from the control plane.

high bandwidth communication and low latency between the computation plane and components in the control plane such as coprocessors, memory, or other devices; and (5) direct, granular access by the control plane to internal structures in the computation plane.

The threat model we address is that of malicious hardware and software in the computation plane, although for this work we assume that the primitives we introduce on the computation plane remain intact, e.g., in the face of various malicious inclusions and probing of the computation plane.

In this paper, we present concepts and ideas for 3-D design based on a system security architecture that supports a variety of policies including those that specify legal communication between policy equivalence classes[2]. We explore minor modifications to the 3-D design flow to support these methods. We also describe new circuit-level primitives to support this technique and introduce the use of distinct layers available in a 3-D IC as a primitive for the physical isolation of hardware components. Specifically, the contributions of this paper are:

- The application to 3-D IC design of a proven design practice based on a system security architecture.
- A general-purpose circuit-level primitive to support this design approach by allowing different control planes to be conjoined with the same computation plane (or vice-versa).
- A diode circuit-level primitive to support this approach by enforcing the one-way flow of information in a 3-D IC.
- A design approach that uses distinct IC layers, with separate lineage and developmental assurance, to achieve physical separation of hardware components, providing secure application capabilities even in the presence of an untrusted processor and OS software.
- Requirements for automated 3-D IC design tools for the physical layout of components. Since fully automated Electronic Design Automation (EDA) for 3-D circuit design and layout are still evolving, this paper provides high-level requirements analysis aimed at influencing their development so that security is adopted as a principal constraint in the practice of 3-D IC design.
- Offloading of testing circuitry to removable test planes to reduce the cost of design for test.

[2]The system resources are partitioned into separate classes, where each member of a class is treated equivalently with respect to the security policy. Technically, an equivalence class is formed by a set and binary relation. Here, the set is all system resources, and the relation is "has the same policy as."

II. Standardization of Interfaces

The ability to conjoin different control planes with the same computation plane allows a variety of application-specific security enhancements (e.g., policy enforcement mechanisms) while reusing a mass-produced computation plane. To accomplish this, however, requires a standard interface to the computation plane. A standardized interface also enables a mass-produced control plane (e.g., a 3-D crypto coprocessor) to be conjoined to a variety of computation planes. Achieving standardization requires overcoming several challenges:

- Standard placement of posts
- Diverse manufacturing processes (e.g., face-to-face vs. face-to-back bonding)
- Diverse electrical and timing properties of dies
- Diverse sizes and form factors of dies
- Diverse packaging options for 3D-ICs
- Standardization places constraints on 3-D floor planning and layout of TSVs

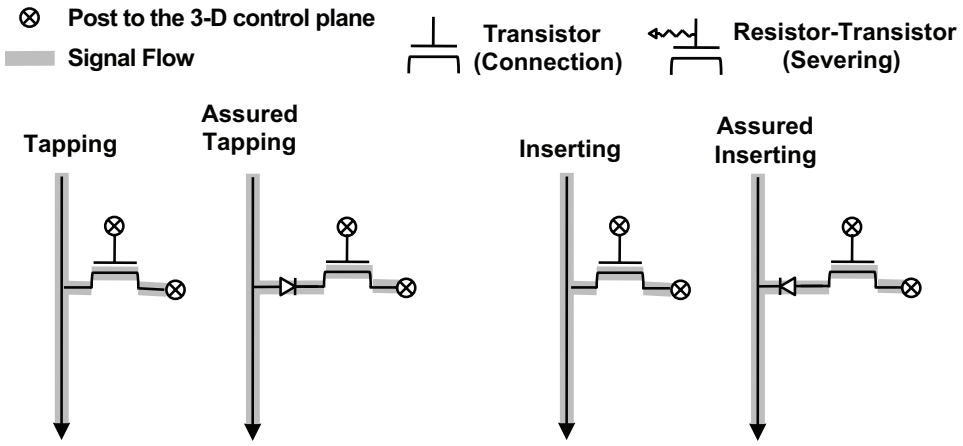

Fig. 3. Primitive TSV functions and corresponding selectively refined functions. The diode is placed between the native (computation plane) circuit and the TSV receptacle, such that the diode controls the flow regardless of how the TSV receptacle is used by the control plane.

The basic idea is the identification of a standard set of logical TSV receptacles on the computation plane, each related to a basic function (e.g., tapping the internal bus, tapping the instruction pipeline, etc.). As long as the basic set of TSV receptacles is present, then variance in their physical layout from processor type to processor type can be accommodated in simple realignment (e.g., with a hardware adaption layer) of the control plane. Furthermore, reusing the precise physical layout of either a control plane or a computation plane is both more challenging and less interesting than reusing the higher-level design of a custom plane.

A. Standard Primitives

In this section we introduce two novel circuit-level primitives, a *diode* and a *general-purpose TSV receptacle*, that support our design approach. Previous work introduced five primitive functions for controlling information flow between planes: disabling, tapping, rerouting, inserting, and overriding, as shown in Figure 2. First, we introduce the diode and two assured functions that result from its application.

1) Diode: A diode, as shown in Figure 3, allows information to flow in only one direction from one component to another[3]. Such an arrangement can enforce a policy requiring that information can flow from a low confidentiality component to a high confidentiality component[4] but not vice-versa. In other words, in a system whose resources have been partitioned into policy equivalence classes, a diode can be applied to a primitive to enforce the one-way flow of information between these classes. Diodes provide a granularity of enforcement related to the granularity of components that they connect. Diodes can be static or programmable, and they can ensure that vertical posts do not violate the inter-plane policy, e.g., by placing diodes between a post and the circuitry of a plane. We expect, however, that this hard-wired enforcement can have unforeseen effects, e.g., on the performance of bidirectional communication protocols, and that *programmable diodes* would provide flexibility in the designs supported.

The primitives each provide an environment for receiving one or two posts. We refer to this computation plane environment as a *TSV receptacle/socket*.

2) Generic TSV Receptacle: A general-purpose TSV receptacle 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. These generic TSV receptacles are used to anchor posts on the computation plane to minimize the number of TSV receptacle types that the processor manufacturer must produce and allows a given TSV receptacle to be used for different purposes from application to appplication. Supporting these features requires identifying standard locations for posts on the computation plane such that generality is balanced against the hardware resources (e.g., posts) required (see future work).

Figure 4 shows a general-purpose TSV receptacle that supports any of the five basic circuit-level primitives. To make use of a TSV receptacle, signals on the control plane determine which primitive is active at a given point in time. Thus, different 3-D applications can use the same TSV receptacle in a different way. Only one configuration of the Generic TSV Receptacle is required even though, for example, one application might read from a TSV receptacle, and another might override circuits with it.

A generic TSV receptacle provides design flexibility at the cost of additional circuitry and posts. The Generic TSV Receptacle accepts four posts (three control and one data) implementing four primitive features internally: one tapping or insertion feature and two disabling features. These are combined to implement disabling (i), tapping (ii and iii), rerouting (ii, iii, and iv), inserting (ii and iii), overriding (i, ii, and iii), and native (none of the posts), which has no effect.

[3]Note that one-way communication can be enforced using other electrical techniques besides a diode

[4]Given a lattice of confidentiality markings, "high" markings are those that are closer to the top (the universal upper bound), and "low" markings are those that are closer to the bottom (the universal lower bound).

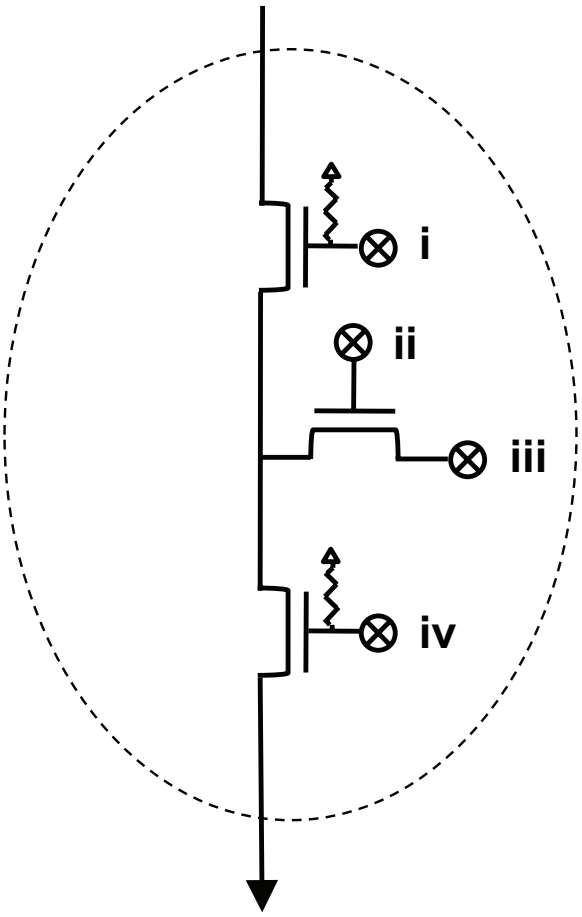

Fig. 4. Generic TSV Receptacle. A general-purpose TSV receptacle supports any of the basic circuit-level primitives (disabling, tapping, rerouting, inserting, and overriding). Signals from the control plane determine which primitive is active at a given point in time. Thus, the application on each different control plane could use a given TSV receptacle in a different way: one application might tap a TSV receptacle, and another might override it. The control posts are i and ii, and the data posts are iii and iv.

The Generic TSV Receptacle could include *diodes* to assure that the information flows of each post are precisely controlled, in which case the diode for post iii could be programmable to support reading or writing.

3) Application Classes: This section describes several categories of 3-D applications that can be built using our framework [28], [8]:

- Secure Alternate Service. This category provides a trustworthy enhancement to the service provided by the computation plane. Examples include ciphers, key storage, compression, and network-on-chip (NoC) routers.
- Isolation and Protection. This category actively overrides the computation plane to enforce access control, eliminate points of interference, or disable communication. Examples include the 3-D cache eviction monitor described in [28] and enforcing a policy on buses or NoC routers in the computation plane, as shown in Figure 6.
- Passive Monitoring. This category passively monitors the computation plane. Examples include audit, information flow tracking, and runtime checks.

III. Isolation of Hardware Equivalence Classes

Arranging system components to structurally support a security policy results in a *security architecture* [14]. Realization of a multi-level security (MLS) policy in a 3-D system requires establishing (1) policy equivalence classes; (2) isolation of components according to those classes; and (3) controlled interaction between classes according to an inter-class communication policy.

4) Policy Equivalence Classes: Grouping similar entities into a domain or *equivalence class* helps simplify the design of secure systems.

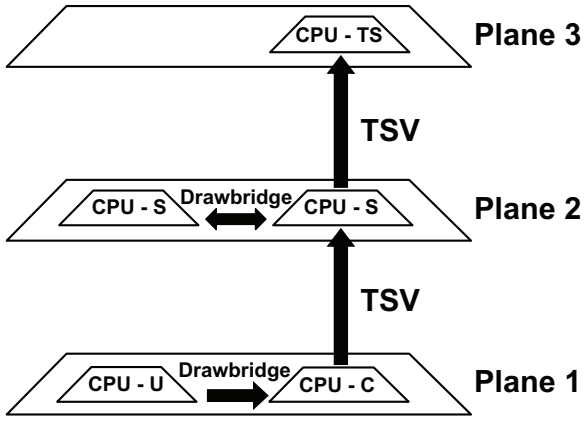

Fig. 5. A hypothetical layout of CPUs on several computation planes, where each CPU is a point in the lattice. This system consists of three layers: the lower layer contains a CPU with an UNCLASSIFIED (U) label and a CPU with a CONFIDENTIAL (C) label; the middle layer contains two CPUs with a SECRET (S) label; and the upper layer contains a CPU with a TOP SECRET (TS) label.

Consider a system security policy for a 3-D IC in which the computation plane contains several cores: a given application workload may require that two cores devoted to a sensitive application be isolated from the rest. To achieve isolation, computational components belonging to the same equivalence class can be placed on the same layer (die). On the other hand, multiple equivalence classes may reside on a layer (see the lower plane in Figure 5) if they are spatially or otherwise separated. Thus, the dies and cores can be partially ordered with strong separation, as shown in Figure 5. To achieve controlled sharing, only specified inter-die posts are permitted, and the 3-D design is statically checked to ensure that posts do not violate the policy. In other words, the physical separation between layers and between cores on a layer result in a conceptual moat, and connections that cross moat boundaries (viz., *drawbridges*), must conform to the policy, as was shown to be effective in [7]. For generality, the diodes and junctions in Figure 5 would be programmable, to support a wide range of policies.

3-D integration offers the unique capability to physically isolate hardware components by arranging circuitry into distinct layers. We extend the idea of a 2-D moat into 3-D,

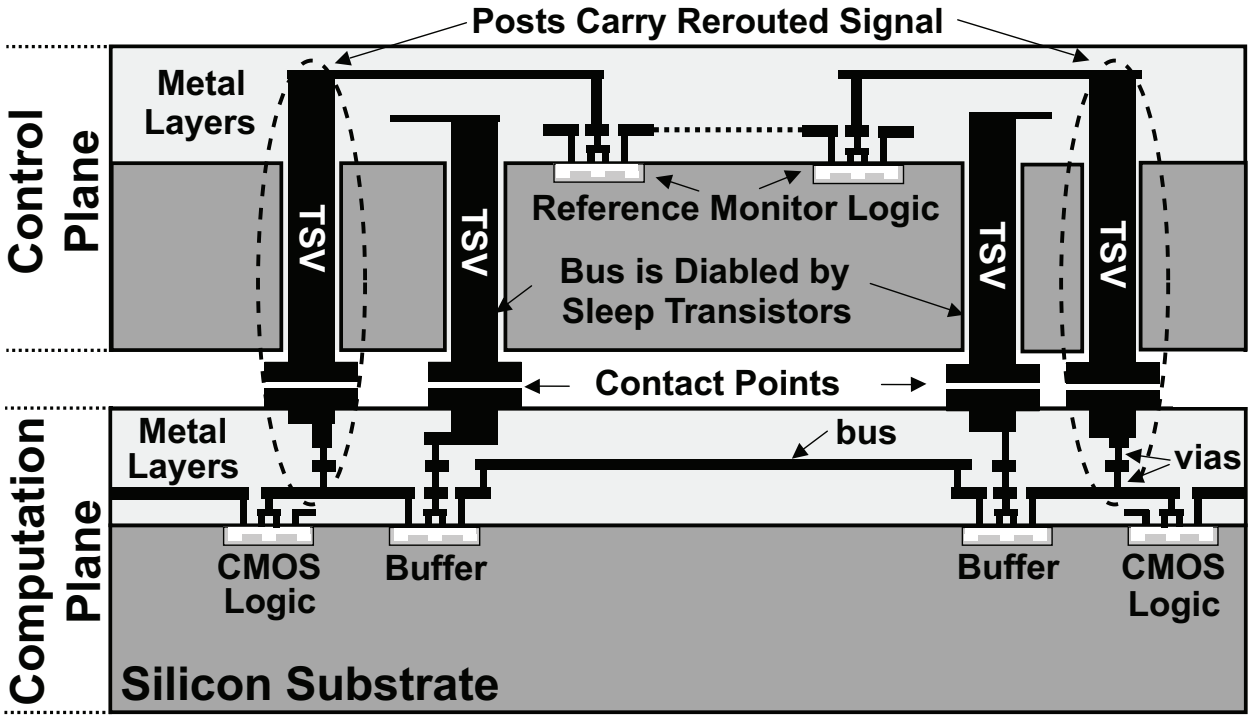

Fig. 6. Example of rerouting bus signals in the computation plane through the control plane [28].

in which physical isolation possible with separate layers is similar to a 2-D moat.

We can use a combination of moats and 3D planes to isolate the equivalence classes. *Moats* separate CPUs on a given plane, and each plane is physically isolated from other planes. *Drawbridges* connect different moats on the same plane.

The isolation need not be complete. The standard MLS policy for information flow allows a "downward" flow in the lattice. Diodes and other electrical mechanisms can enforce such a one-way flow and we can use automated analysis techniques such as information flow tracking [27], [25] to verify that the flows conform to the policy.

To facilitate the controlled interaction of isolated layers, it is necessary to ensure that only specified vertical connections exist between these layers, where the specification includes directionality of information allowed between layers consistent with the MLS sensitivity of the layers. We also extend the idea of a 2-D drawbridge into 3-D, in which the architectural integrity of vertical connections between layers must be statically checked, similar to a 2-D drawbridge.

Furthermore, it is possible to use 2-D moats and drawbridges within an individual layer, when design requirements (e.g., cost limitations) dictate that more than one equivalence class must reside on a layer. In order to separate the cores residing on a given chip, various structures must be partitioned and/or virtualized, including on-chip memory and computational components. Other *points of interference* between cores that must be constrained include interconnect (e.g., on-chip bus or on-chip network), I/O devices, and dependencies (e.g., power, I/O, privilege, etc.). Other structures (e.g., micro-connections not visible at a high level of system abstraction) may also need to be partitioned.

5) Core Isolation: Isolation is a foundational concept in computer security [24]. At the hardware level, it is possible to isolate circuitry spatially. Hardware functionality can be physically isolated by using air gaps or other techniques described as moats and drawbridges [7], [9]. To facilitate the controlled interaction between isolated equivalence classes, it is necessary to ensure that only specified connections exist between these domains. In the case where system components communicate over a shared bus (or on-chip network), the interconnect must be designed to prevent illegal information flow between equivalence classes, e.g., using diodes or similar arrangement to control the direction of flow.

6) Inter-Class Communication Policy: At the highest level of abstraction, an inter-class communication policy for a 3-D system must first specify what the equivalence classes are and what information flows are permissible between the classes. Then, the system resources should be partitioned with respect to the equivalence classes. Figure 5 shows an example of a lattice policy. Barring the use of a time sharing approach, the policy should also specify which equivalence class each core is assigned. A given equivalence class may span multiple dies and even off-chip I/O and memory traffic. This high-level policy must be mapped to enforcement mechanisms at a lower level of abstraction, either by a talented human designer or by automated tools that assist the designer in visualizing the 3-D security architecture and exploring the design space of

3-D mechanisms. Finally, the flows allowed between classes are chosen, providing a partial order relation required of the lattice. For example, in an MLS policy, we order the classes (TS $\geq$ S $\geq$ C $\geq$ U) and allow a flow from class A to class B only if B $\geq$ A [3].

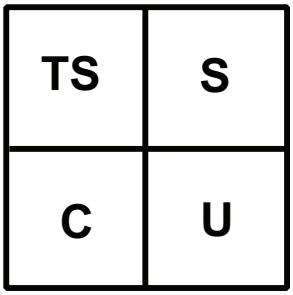

Fig. 7. Top view of a hypothetical multi-core integrated circuit. Four regions of the chip correspond to the hierarchical General Service (GENSER) equivalence classes of TOP SECRET (TS), SECRET (S), CONFIDENTIAL (C), and UNCLASSIFIED (U).

Figures 7 and 8 show examples of more complex 3-D security architectures. The system shown in Figure 8 obeys an MLS policy, as described above. Diodes enforce the one-way flow of information. Junctions, which are two-way connections, connect memory and cores with equal labels. Programmable junctions can be set to act as a diode for which the layer reads the TSV, a diode for which the TSV reads the layer, or a two-way connection.

To avoid the security issues of sharing a CPU between equivalence classes (e.g., side channels and other covert channels) we dedicate each CPU to an equivalence class in our lattice-based policy, and then ensure that unwanted interference between CPUs is impossible.

Included in the equivalence class of a given CPU are its dedicated memory (L1) and any dedicated board-resident devices. L2 memory may be shared with another CPU only if cache side channel interference has been eliminated, effectively virtualizing the L2 cache [29], as shown in Figures 9 and 10.

7) 3-D Design Flow: As the standards for 3-D Electronic Design Automation (EDA) tools are still emerging, a complete standard design flow is not yet available, as described in [1], [21], and [15]. Existing design flows are limited to a specific 3-D fabrication technology and a fixed number of planes, and there is no standard fabrication technology. Therefore, talented human designers must balance multiple constraints simultaneously to achieve the desired design properties. In a 3-D design flow, the positioning of components on different layers may perturb the properties achieved on the individual layers, e.g., vertical proximity may increase thermal factors.

3-D design is very challenging because the large number of interacting constraints result in a complex optimization problem. Specifically, the designer must balance several factors in achieving overall properties such as performance, bandwidth, yield, cost, and testability:

- thermal limits, power density, and electrical interference
- power distribution (current delivery) [11] and power limits
- clock delivery, timing analysis, and dissimilar clock rates
- packaging, handling, wafer alignment, and bonding
- mechanical stress [2] and metal resistivity/expansion
- via density, via size, via count, and via location
- the number of vertical connections required and the length of the posts
- variation across planes: dissimilar via resistance and pitch
- I/O, die-to-die signaling, bus fan-in, and bus fan-out

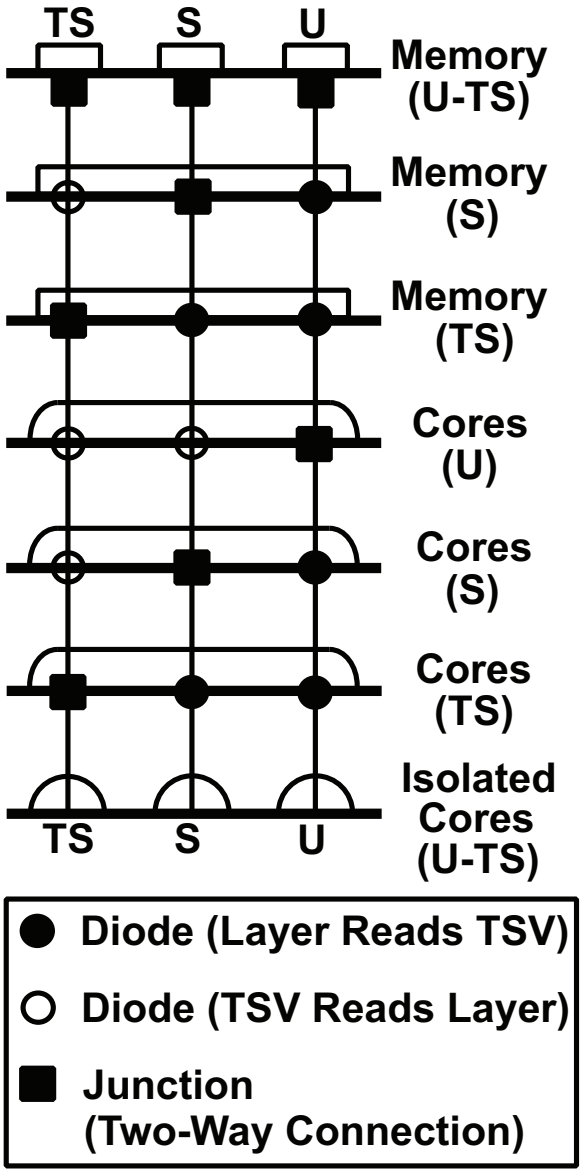

Fig. 8. Side view of a hypothetical 3-D integrated circuit with seven layers.

Despite these daunting challenges, numerous 3-D systems have been built (see Section VII: Related Work), including a complex processor with stacked memory fabricated at Georgia Tech [17], [16].

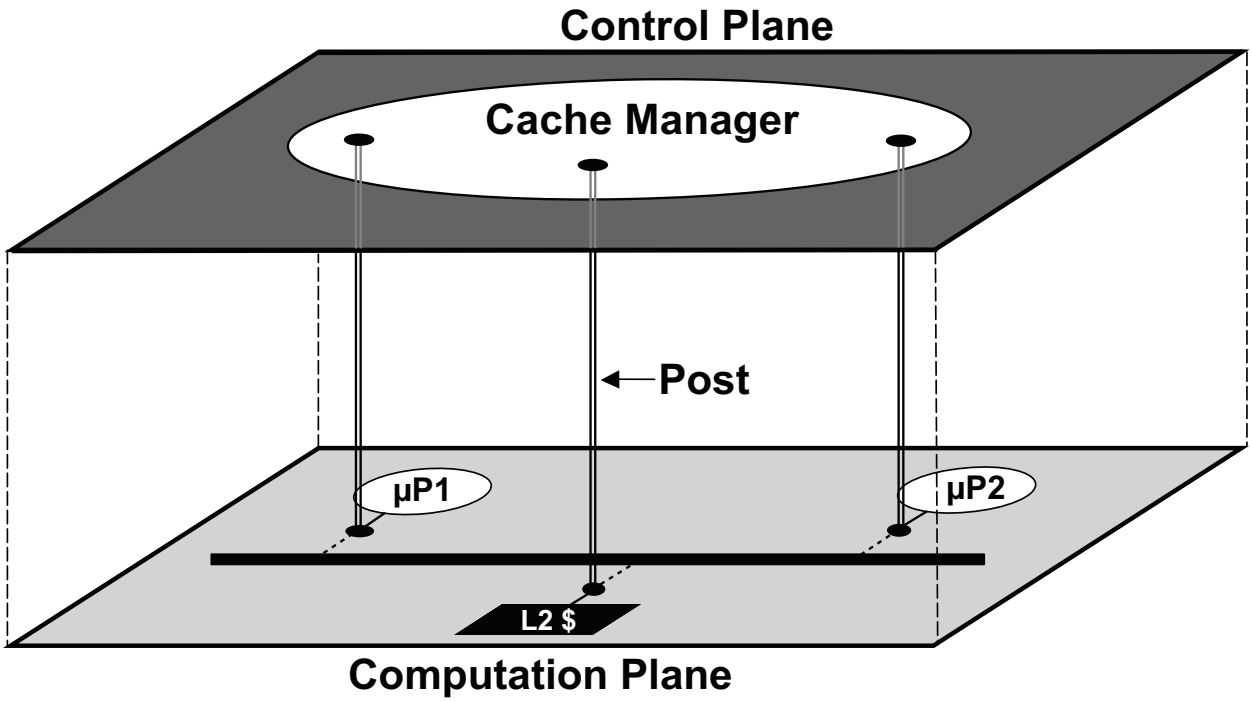

Fig. 9. System-level view of a two-tier 3-D IC. The computation plane contains a dual-core chip multi-processor and shared L2 cache, all connected to a shared bus. The control plane contains a cache manager that partitions the shared L2 cache.

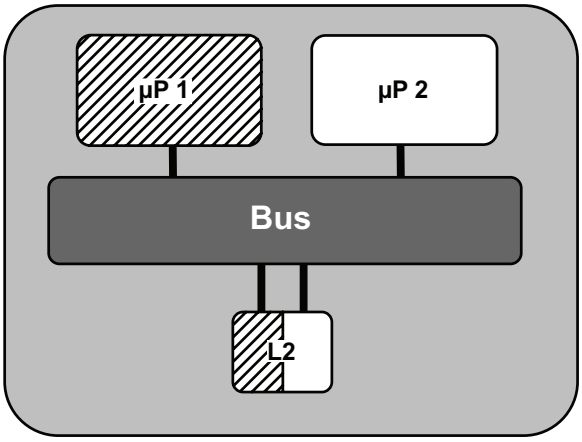

Fig. 10. System-level view of the computation plane of the two-tier 3-D IC shown in Figure 9.

8) Design Flow Modifications: Our technique involves making minor modifications to the 3-D IC design flow, specifically the stage referred to as *floor planning*, which is performed after logic synthesis and the validation of the functional correctness of the circuitry. In other words, the design is initially agnostic about floor planning, yet partitioning of certain resources among layers (to respect a policy-based partial ordering of resources) may be a first-order constraint.

To most effectively use 3-D technology, the designer must consider the ordering and connections of the 3-D security architecture as constraints when performing floor planning. Prior to floor planning, the designer must already have a coarse-grained understanding of (1) what computational resources will reside on which layers; (2) the vertical connections permitted between layers; (3) the transitive closure of flows induced by posts; (4) the horizontal connections permitted between components within a layer; and (5) constraints on 2-D and 3-D interconnect (e.g., bus or router), such as enforcing a Time Division Multiple Access (TDMA) scheme.

Arriving at this coarse-grained picture requires the designer to refine the security architecture. Automated tools can assist the designer in exploring the design space of 3-D security architectures and to validate that the security architecture correctly supports the policy.

A multistep floor planning approach is described in [21], which involves first assigning blocks to planes (i.e., *partitioning*) and then allowing blocks to be rearranged within a plane (i.e., *intra-plane moves*). Allowing simultaneous inter- and intra-plane moves results in an unmanageable optimization problem size. Working within this multistep paradigm, security constrains the first step when, for example, blocks are assigned to planes according to the functional relationship between planes and blocks. Security also constrains the second step when blocks are rearranged within a plane to achieve the specified security properties. For example, intra-plane rearrangement of blocks results in the positioning of a TS memory region in one plane directly above a TS core in another plane and may require the arrangement of air gaps between blocks within a plane. To decrease the complexity of the optimization problem, the design tools could be instructed to only allow the

partial ordering of components rather than the more stringent requirement of assigning components to specific planes.

Floor planning establishes the precise geometric boundaries of (1) the computational cores on the layers; (2) the vertical connections between layers; and (3) the horizontal connections within a layer. At this stage, it is necessary to statically check that horizontal connections that cross 2-D moat boundaries and vertical connections that cross layers (both extended with transitive relations) do not violate the system security policy (e.g., checking for extraneous connections between equivalence classes). If these geometric boundaries of elements determined during the layout process are specified in the GDS II database file format, a static analysis program can check the design for policy conformance by analyzing the GDS II file.

A. Other Uses besides MLS

1) Interconnect Built to Support MLS Policy: As discussed earlier, an MLS security policy is based on a *lattice* of sensitivity labels, e.g., TOP SECRET (TS), SECRET (S), CONFIDENTIAL (C), UNCLASSIFIED (U). Each resource controlled by the policy is assigned a label, such that the labels partition the resources. All of the resources with the same label are said to be in the same *equivalence class*.

Lattice-based policies can be generalized beyond those with national security labels, as long as there is a means of determining in which equivalence class a resource resides, and the equivalence classes form a lattice.

2) Self-Protection and Dependency Layering: A secure application must be protected from attack and must not depend on any components that are less trustworthy. In a 3-D system, self-protection requires that the computation plane cannot short-circuit or surge the power of the control plane, make requests to the control plane that cause buffers to overflow, or modify the control plane. Dependency layering requires that the control plane not request service from or wait on the computation plane.

This paper does not address package-level concerns related to I/O and power other than the isolation provided by separate layers within the enclosure of the package. A more thorough discussion of the axioms of self-protection and dependency layering is applied to 3-D IC design in [8].

B. Topology Considerations

The form factor of the dies is a security architecture design consideration. For example, a small memory tile could be stacked on top of one moated region belonging to an equivalence class. Provided that its form factor allows it to remain within the bounds of the moated region on which it is placed, it should be separate from another memory die stacked on top (but within the bounds of) another moated region belonging to a different equivalence class. A wide variety of 3-D structures are possible in this manner.

IV. Power and I/O Independence

A variety of methods for achieving I/O independence for the control plane are possible [8]. First, the control plane can be located closest to the I/O and power pins, such that it provides these services to the computation plane. Otherwise, wireless methods include capacitive/inductive coupling, short-range RF, short-range optical, and simply attaching EEPROM. Wired options include JTAG interface, serial cable, dedicated pins, TDMA over HyperTransport, and dedicated memory ranges. Providing an independent source of power to the control plane can be achieved using the same circuit-level primitives described above for computation signals. Wireless power transmission technology is another option.

V. Offloading of Testing Circuitry

Design for test circuitry consumes significant die area and therefore plays a major role in the cost of mass-produced processors[5], [26]. DFT must not impact performance; must have a small area cost; allow multiple uses when possible; and be integrated into the design from the beginning [5].

While 3D-ICs have their own set of daunting test challenges [13], [30], [19], we argue that significant parts of the DFT circuitry can be offloaded from the computation plane to the control plane to reduce area impact. Furthermore, a 3-D approach can mitigate some security concerns associated with DFT circuitry. For example, Yang et al. showed that crypto processors are vulnerable to attacks that use scan-based design for test to steal crypto secrets [31]. With a 3-D approach, the test interface does not have to ship with production systems; instead, only a select number of computation planes are joined with testing planes to support factory testing.

VI. Experimental Framework

We are in the process of preparing a test bench for experimentally validating and demonstrating the effectiveness of our circuit-level primitives to determine how well they work when fabricated. The 3-D chip will include a test harness which will manage the invocation of and vary the inputs to each circuit-level primitive and will read the outputs to verify that the expected behavior occurs. The experiment will build the primitives in different configurations. The frequency at which a circuit can operate without modification will be compared against: (1) the frequency at which it can operate with the circuit-level modifications; and (2) the frequency at which it an operate with both the circuit-level modifications and the addition of a control plane.

VII. Related Work

While 3-D integration is an emerging technology, a variety of 3-D applications have been realized, including imaging [32], medicine [10], particle physics [6], reconfigurable hardware [23], as well as high-performance microprocessors, as described in [4], [22], [17], [16], and [12],

A 3-D cache monitor designed to mitigate access-driven cache side channel attacks in a simultaneous multithreading processor's shared memory [29] has negligible computation plane overhead in terms of area, delay, and performance [28]. The control plane maintains a data structure that records whether cache lines are protected and for what process. Cache

evictions of a protected line are denied based on the security policy. Experiments used an FPGA synthesis tool to collect area and timing information, comparing the case involving just the computation plane, the case involving just the control plane, and the case involving the combined computation and control plane. Evaluation of the impact of the delay of the posts was based on data from Loi et al. [18], which found that the worst-case delay between opposite corners of a chip to be approximately .29ns, which is too small to affect the performance of the critical path of the 3-D cache eviction monitor. Analysis by Mysore et al. showed that the additional area required for vias is small for 3-D applications that perform introspection and profiling of the computation plane [20].

This paper builds on earlier work presented in [28] and [8] that applied 3-D integration to the problem of hardware-oriented security and trust as well as on the *moats and drawbridges* technique developed for Field Programmable Gate Arrays (FPGAs) presented in [7] and [9]. In addition to the 3-D circuit-level primitives presented in [28] and [8], in this paper we discuss an additional circuit-level primitive, the *diode*, and we build upon this diode primitive to support policy enforcement in a 3-D IC. We also introduce a *general-purpose TSV receptacle* that can implement any of the primitives described in earlier work, supported as necessary by the diode primitive introduced in this paper.

VIII. Future Work

The use of disabling posts to selectively disable (by removing power) wires in the computation plane that violate the isolation of components could break some designs, in which case the circuitry could remain unchanged but its use audited by other posts. Determining which connections to disable could utilize analytical tools from graph theory, e.g., determination of *cut points*. 3-D posts connected to these cut points could be used to selectively disable the connections without affecting other portions of the circuit. Ideally, the native computation plane designer eliminates all points of interference between the cores of a multi-core processor. However, some connections are needed by some applications but not others. It is possible that these connections could be selectively disabled according to the policy. We also leave the following to future work:

- The use of a reference monitor for providing fine-grained policy enforcement in 3-D chips.
- The use of disabling posts in the context of dynamic runtime policy changes, e.g., to allow moats that are configurable at runtime, i.e., movable moats.
- The use of configurable diodes in the context of reconfigurable policy changes.
- The use of disabling and inserting posts to modify the behavior of interconnect in the computation plane.
- The use of rerouting and overriding posts to force bus traffic in the computation plane to take a detour to an alternate bus in the control plane where various policies can be enforced.
- Using disabling posts to partition a computation plane consisting of entwined cores that are not already spatially isolated.
- The use of a module in the control plane to erase architectural state in the computation plane in order to address data remanence.
- Identification of standard locations for TSV receptacles on general-purpose processors.

IX. Conclusion

Security must become a first-order design constraint in the engineering of 3-D systems. We have described a design approach based on a 3-D system security architecture. To support this approach, we have described two novel circuit-level primitives: (1) a general-purpose TSV receptacle that allows the same posts to be reused in different ways by different applications and (2) a diode that restricts the vertical flow of information to one direction. Our design approach also takes advantage of the physical isolation provided by distinct layers. Finally, we describe modifications to the floor planning stage of the 3-D design flow that are necessary to support our design approach. We strongly recommend that the 3-D EDA community incorporate features in commercial design tools for the hardware-oriented security and trust community to constrain the floor planning of components in a 3-D IC. Design flows should provide researchers and practitioners the flexibility to, for example, achieve isolation of and programmable partial ordering of selected components.

X. Acknowledgments

This research was funded by National Science Foundation Grants CNS-0910734, CNS-0910389, and CNS-0910581.

References

[1] Global Semiconductor Alliance. Tour guide to 3D-IC design tools and services. In *Presentation at the 3D/TSV Technology Reception: Education, Benefits, and Solutions at the Design Automation Conference (DAC)*, Anaheim, CA, June 2010.

[2] Krit Athikulwongse, Ashutosh Chakraborty, Jae-Seok Yang, David Z. Pan, and Sung Kyu Lim. Stress-driven 3D-IC placement with TSV keep-out zone and regularity study. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, November 2010.

[3] D.E. Bell and L.J. La Padula. Secure computer system: Unified exposition and Multics interpretation. Technical Report ESD-TR-75-306, The MITRE Corporation, Hanscom Air Force Base, Bedford, MA, USA, March 1976.

[4] Bryan Black, Murali Annavaram, Ned Brekelbaum, John DeVale, Lei Jiang, Gabriel H. Loh, Don McCaule, Pat Morrow, Donald W. Nelson, Daniel Pantuso, Paul Reed, Jeff Rupley, Sadasivan Shankar, John Shen, and Clair Webb. Die stacking (3D) microarchitecture. In *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Orlando, FL, December 2006.

[5] Adrian Carbine and Derek Feltham. Pentium pro processor design for test and debug. *IEEE Design and Test of Computers*, 15(3), July–September 1998.

[6] Marcel Demarteau, Yasuo Arai, Hans-Gunther Moser, and Valero Re. Developments of novel vertically integrated pixel sensors in the high energy physics community. In *IEEE International Conference on 3D System Integration*, San Francisco, CA, September 2009.

[7] Ted Huffmire, Brett Brotherton, Gang Wang, Tim Sherwood, Ryan Kastner, Timothy Levin, Thuy Nguyen, and Cynthia Irvine. Moats and drawbridges: An isolation primitive for reconfigurable hardware based systems. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 2007.

[8] Ted Huffmire, Timothy Levin, Michael Bilzor, Cynthia E. Irvine, Jonathan Valamehr, Mohit Tiwari, Timothy Sherwood, and Ryan Kastner. Hardware trust implications of 3-D integration. In *Proceedings of the 5th Workshop on Embedded Systems Security (WESS)*, Scottsdale, AZ, October 2010.

[9] Ted Huffmire, Timothy Levin, Thuy Nguyen, Cynthia Irvine, Brett Brotherton, Gang Wang, Timothy Sherwood, and Ryan Kastner. Security primitives for reconfigurable hardware-based systems. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 3(2), May 2010.

[10] Y. Kaiho, Y. Ohara, H. Takeshita, K. Kiyoyama, K-W Lee, T. Tanaka, and M. Koyanagi. 3D integration technology for 3D stacked retinal chip. In *IEEE International Conference on 3D System Integration*, San Francisco, CA, September 2009.

[11] Nauman H. Khan, Syed M. Alam, and Soha Hassoun. System-level comparison of power delivery design for 2D and 3D ICs. In *Proceedings of the IEEE International Conference on 3D System Integration (3DIC)*, San Francisco, CA, September 2009.

[12] Jongman Kim, Chrysostomos Nicopoulos, Dongkook Park, Reetuparna Das, Yuan Xie, N. Vijaykrishnan, Mazin S. Yousif, and Chita R. Das. A novel dimensionally-decomposed router for on-chip communication in 3D architectures. In *Proceedings of the 34th International Symposium on Computer Architecture*, San Diego, CA, June 2007.

[13] Hsien-Hsin S. Lee and Krishnendu Chakrabarty. Test challenges for 3d integrated circuits. *IEEE Design and Test of Computers*, 26(5), September/October 2009.

[14] Timothy E. Levin, Cynthia E. Irvine, Clark Weissman, and Thuy D. Nguyen. Analysis of three multilevel security architectures. In *Proceedings of the ACM Computer Security Architecture Workshop (CSAW)*, Fairfax, VA, November 2007.

[15] Zhouyuan Li, Xianlong Hong, Qiang Zhou, Shan Zeng, Jinian Bian, Hannah Yang, Vijay Pitchumani, and Cheng-Kuan Cheng. Integrating dynamic thermal via planning with 3D floorplanning algorithm. In *Proceedings of the International Symposium on Physical Design (ISPD)*, San Jose, CA, April 2006.

[16] Gabriel H. Loh. 3-D stacked memory architectures for multi-core processors. In *International Symposium on Computer Architecture (ISCA)*, Beijing, China, June 2008.

[17] Gabriel H. Loh, Yuan Xie, and Bryan Black. Processor design in 3D die-stacking technologies. *IEEE Micro*, 27(3), May-June 2007.

[18] Gian Luca Loi, Banit Agrawal, Navin Srivastava, Sheng-Chih Lin, Timothy Sherwood, and Kaustav Banerjee. A thermally-aware performance analysis of vertically integrated (3-D) processor-memory hierarchy. In *Proceedings of the 43rd Design Automation Conference (DAC)*, San Francisco, CA, July 2006.

[19] Erik Jan Marinissen. Testing tsv-based three-dimensional stacked ics. In *Proceedings of the Conference on Design, Automation, and Test in Europe (DATE)*, Dresden, Germany, March 2010.

[20] S. Mysore, B. Agrawal, S.C. Lin, N. Srivastava, K. Banerjee, and T. Sherwood. Introspective 3-D chips. In *Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, San Jose, CA, October 2006.

[21] Vasilis F. Pavlidis and Eby G. Friedman. *Three-Dimensional Integrated Circuit Design*. Morgan Kaufmann, Boston, MA, 2009.

[22] K. Puttaswamy and G.H. Loh. Thermal analysis of a 3D die-stacked high-performance microprocessor. In *Proceedings of the 16th ACM Great Lakes Symposium on VLSI (GLSVLSI'06)*, Philadelphia, PA, May 2006.

[23] Seyyed Ahmad Razavi, Morteza Saheb Zamani, and Kia Bazargan. A tileable switch module architecture for homogeneous 3D FPGAs. In *Proceedings of the IEEE International Conference on 3D System Integration*, San Francisco, CA, September 2009.

[24] Jerome H. Saltzer and Michael D. Schroeder. The protection of information in computer systems. *Communications of the ACM*, 17(7), July 1974.

[25] G. Edward Suh, Jae W. Lee, David Zhang, and Srinivas Devadas. Secure program execution via dynamic information flow tracking. In *Proceedings of the 11th international conference on Architectural support for programming languages and operating systems (ASPLOS)*, Boston, MA, October 2004.

[26] Kenneth M. Thompson. Intel and the myths of test. *IEEE Design and Test of Computers*, 13(1), Spring 1996.

[27] Mohit Tiwari, Hassan Wassel, Bita Mazloom, Shashidhar Mysore, Frederic Chong, and Timothy Sherwood. Complete information flow tracking from the gates up. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Washington, DC, March 2009.

[28] Jonathan Valamehr, Mohit Tiwari, Timothy Sherwood, Arash Arfaee, Ryan Kastner, Ted Huffmire, Cynthia Irvine, and Timothy Levin. Hardware assistance for trustworthy systems through 3-D integration. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, Austin, TX, December 2010.

[29] Z. Wang and R. Lee. New cache designs for thwarting cache-based side channel attacks. In *Proceedings of the 34th International Symposium on Computer Architecture*, San Diego, CA, June 2007.

[30] Xiaoxia Wu, Paul Falkenstern, Krishnendu Chakrabarty, and Yuan Xie. Scan-chain design and optimization for three-dimensional integrated circuits. *ACM Journal on Emerging Technologies in Computing Systems*, 5(2), July 2009.

[31] Bo Yang, Kaijie Wu, and Ramesh Karri. Secure scan: A design-for-test architecture for crypto chips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(10), October 2006.

[32] Hiroshi Yoshikawa, Atsuko Kawasaki, Tomoaki Iiduka, Yasushi Nishimura, Kazumasa Tanida, Kazutaka Akiyama, Masahiro Sekiguchi, Mie Matsuo, Satoru Fukuchi, and Katsutomo Takahashi. Chip scale camera module (CSCM) using through-silicon via (TSV). In *IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, February 2009.