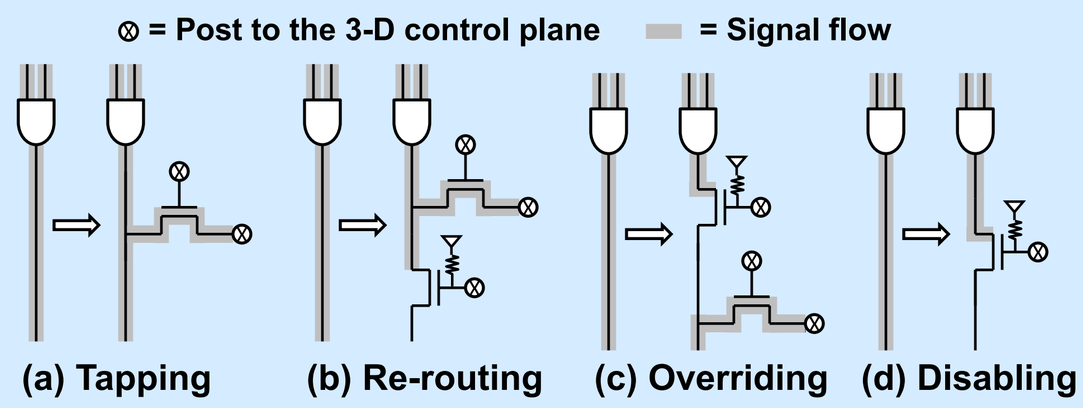
3D Security

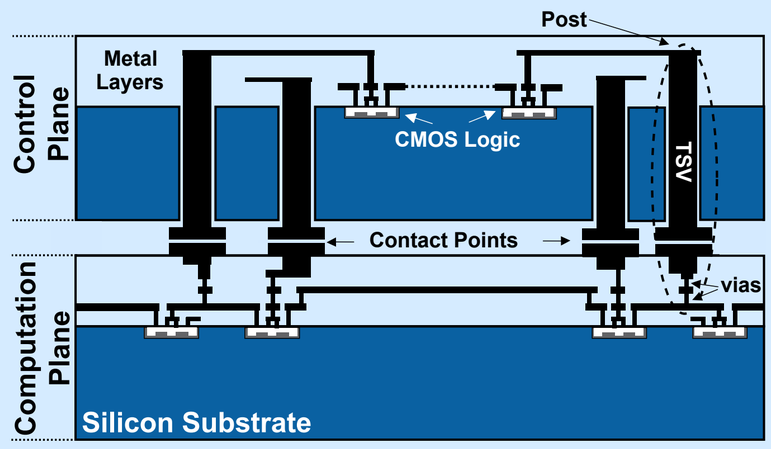
http://cisr.nps.edu/projects/3dsec.html

Ted Huffmire, Ryan Kastner, Tim Sherwood, Cynthia Irvine, Tim Levin, and Jonathan Valamehr

3Dsec: Trustworthy System Security through 3D Integrated Hardware

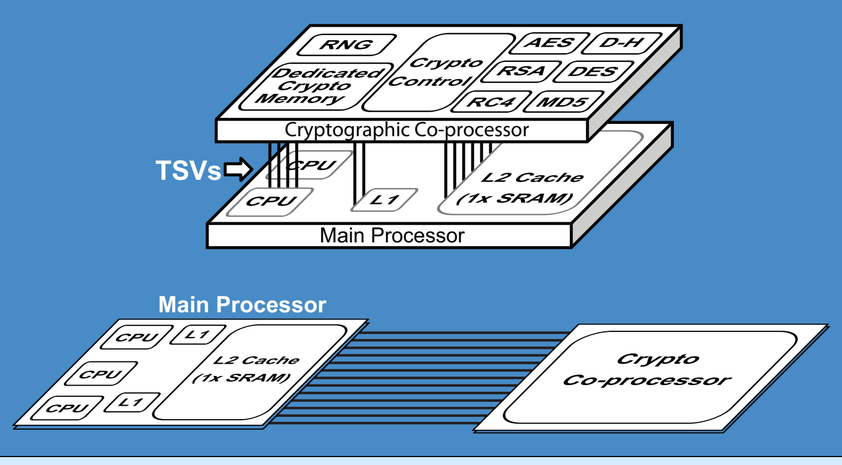
This project is investigating a novel approach to trustworthy system development based on 3D integration, an emerging chip fabrication technique in which two or more integrated circuit dies are combined into a single stack using vertical conductive posts. Since the dies may be manufactured separately, 3D circuit integration offers the option of enhancing a commodity processor with a variety of custom security functions, which are manufacturing options applicable only to those systems that require them. This research introduces a fundamentally new method to incorporate security mechanisms into hardware and has the potential to significantly shift the economics of trustworthy systems. We use the term *computation plane* to refer to a commodity processor die, and we use the term *control plane* to refer to an additional die, containing customized security functions, that is joined to the computation plane. Since the computation plane must be able to function correctly in the absence of the control plane, circuit-level primitives are used in conjunction with the posts for communication between the planes. The basic classes of 3D "applications" that can be constructed include: (1) secure alternate service; (2) isolation and protection; and (3) passive monitoring.

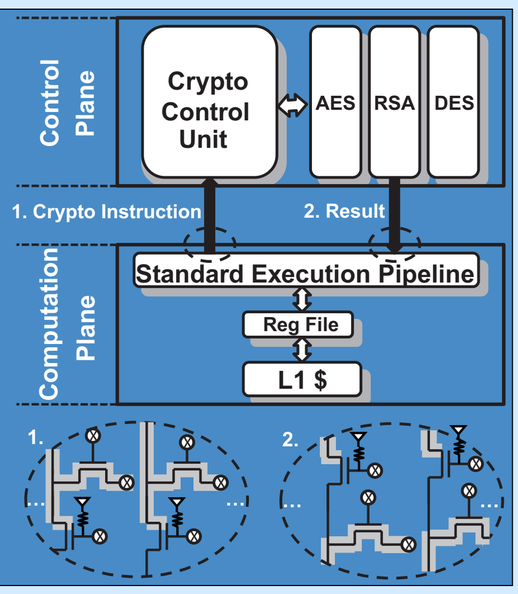




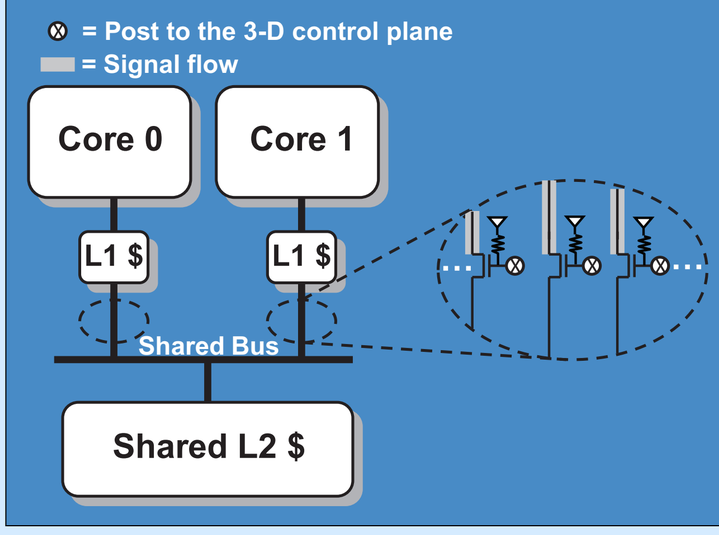
Secure Alternate Service

A *secure alternate service* provides a trustworthy enhancement or alternative to the service provided in the computation plane. An example of a secure alternate service is a cryptographic processor and secure storage implemented in the control plane, like a traditional coprocessor but with much higher bandwidth. Cryptographic keys and other valuable data can be stored and protected in the control plane. Other secure alternate services include computation logic (e.g., ALU, stack, etc.) and processor-core interconnect (e.g., bus and network-on-chip grids), which can be implemented in the control plane either as an addition or enhancement to existing mechanisms in the computation plane or as a duplicate but more highly assured service.



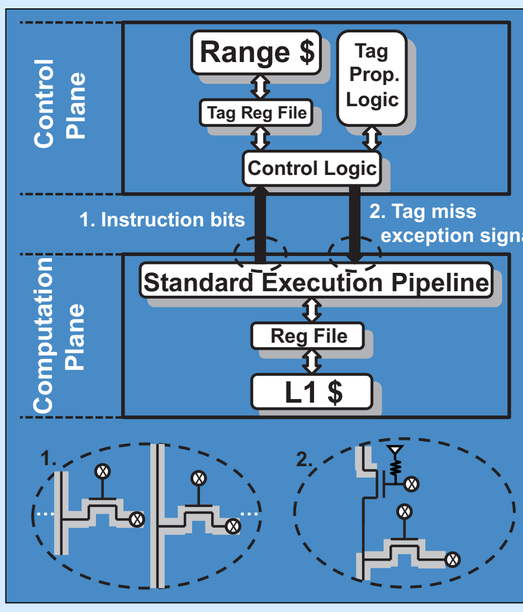


Isolation and Protection

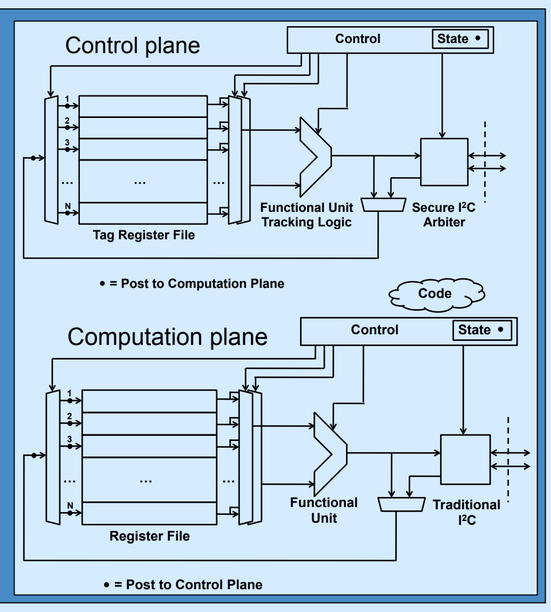


Another category of control plane application actively overrides the computation plane to enforce some security policy, such as for access control. In one form, this type of mechanism can eliminate *points of interference*, such as the cache and branch prediction unit. Disabling connections (e.g., "cutting" wires) to isolate or control the flow of information between cores in a multi-core computation plane (viz., per the reference monitor concept) also falls under this category. *Points of interference* in the computation plane include architectural elements such as the cache and branch prediction units which are hardware features from different protection domains. 3D control functions that actively override the connections to these shared resources can eliminate their misuse, for example, if one process can observe the cache use of another process, a covert channel can be established between them. A function in the control plane can eliminate this covert channel by overriding the computation plane such that each process is restricted to its designated region of the cache. Side channels have become a vexing problem as new points of interference are realized, e.g., in the cache or branch prediction unit. To solve this requires identification of the cut points required to isolate a processor, severing the ones that are not critical and saving or restoring the rest.

Passive Monitoring



Another broad category of control plane application is that which passively monitors the computation plane. These applications may record various properties and values from the computation plane, but do not alter any data or behavior. Examples include auditing and information flow tracking, or computing and reporting a checksum on the architectural state of the computation plane to ensure a secure boot sequence, for example. Passive monitoring can also be used to perform runtime checks on the computation plane for security and correctness, enhancing runtime self-tests of the processor, and performance profiling.



**NSF Secure and Trustworthy Cyberspace (SaTC)
Principal Investigators Meeting
November 27-29, 2012
National Harbor, MD**