RCsec reconfigurable security

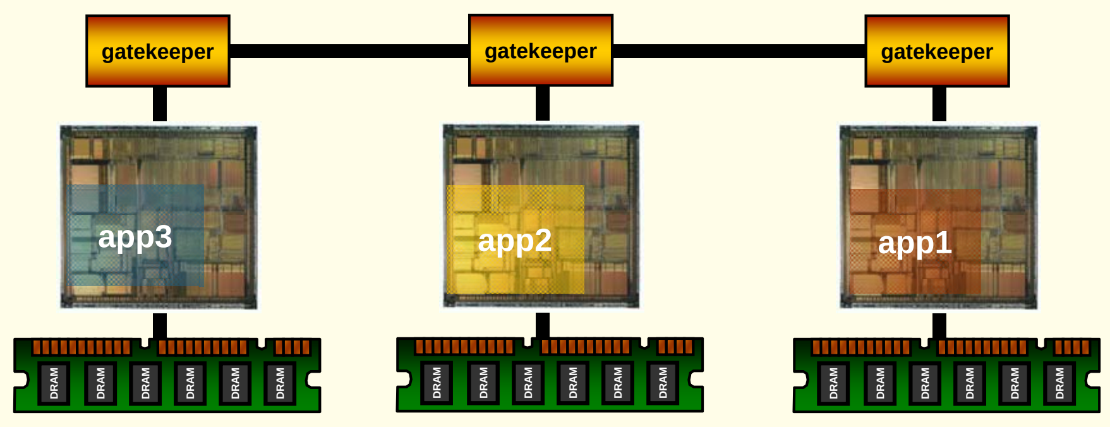
reconfigurable security

http://www.cs.ucsb.edu/~arch/RCsec/

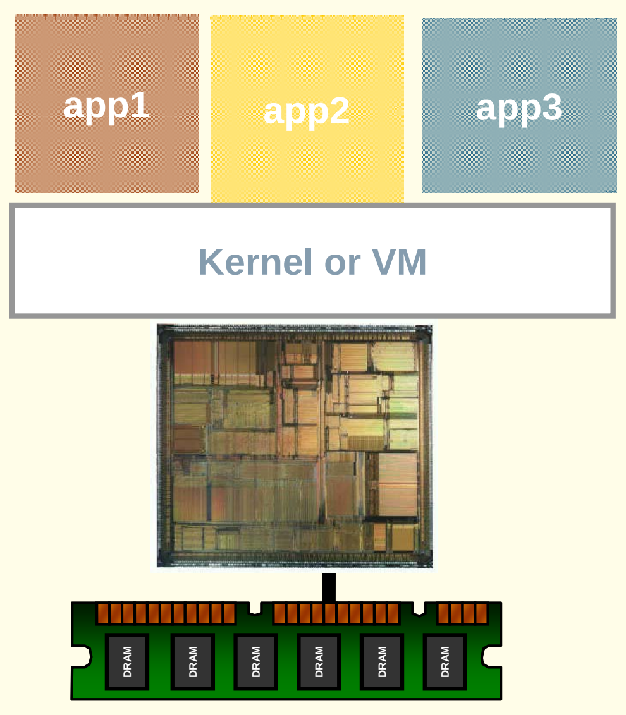
Tim Sherwood, Cynthia Irvine, Ryan Kastner, Thuy Nguyen, Tim Levin, Ted Huffmire

Adaptive Security and Separation in Reconfigurable Hardware

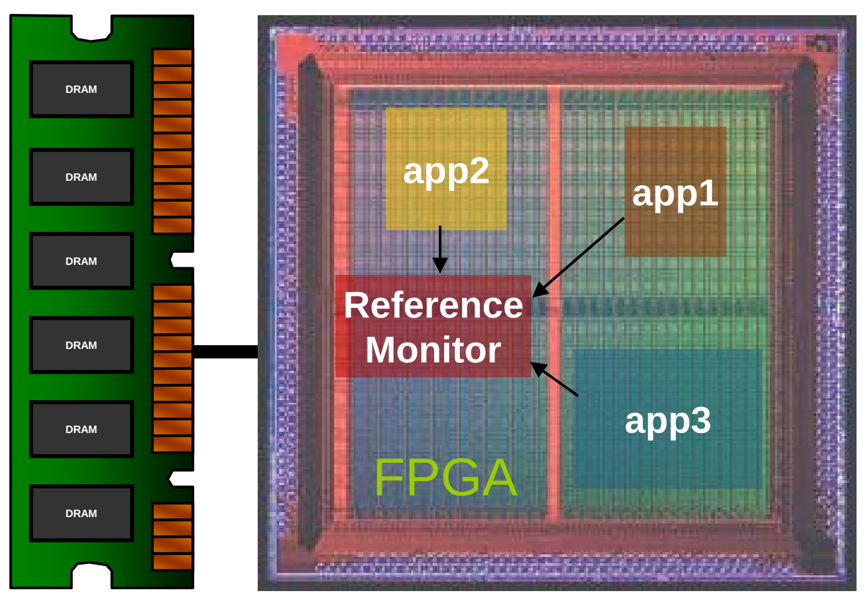
Blurring the line between software and hardware, reconfigurable devices strive to strike a balance between the raw high speed of custom silicon and the post-fabrication flexibility of programmable processors. This flexibility is a boon for embedded system developers, who can now rapidly prototype and deploy solutions that include a variety of ``soft IP cores" from different third-party vendors, with performance approaching that of custom silicon designs. However, in reality the various cores, which may share external resources such as memory, can possess divergent levels of trustworthiness and be provided by mutually suspicious vendors. The problem is that, unlike traditional software where resources are managed by an operating system, soft IP cores in reconfigurable devices necessarily have direct, fine grain control over the underlying hardware, and can intercept or even interfere with the operation of one another. We address this problem with a set of novel security primitives and the complementary use of both static and dynamic techniques for isolation of cores.



Isolation Option 1: Physically Separate Cores



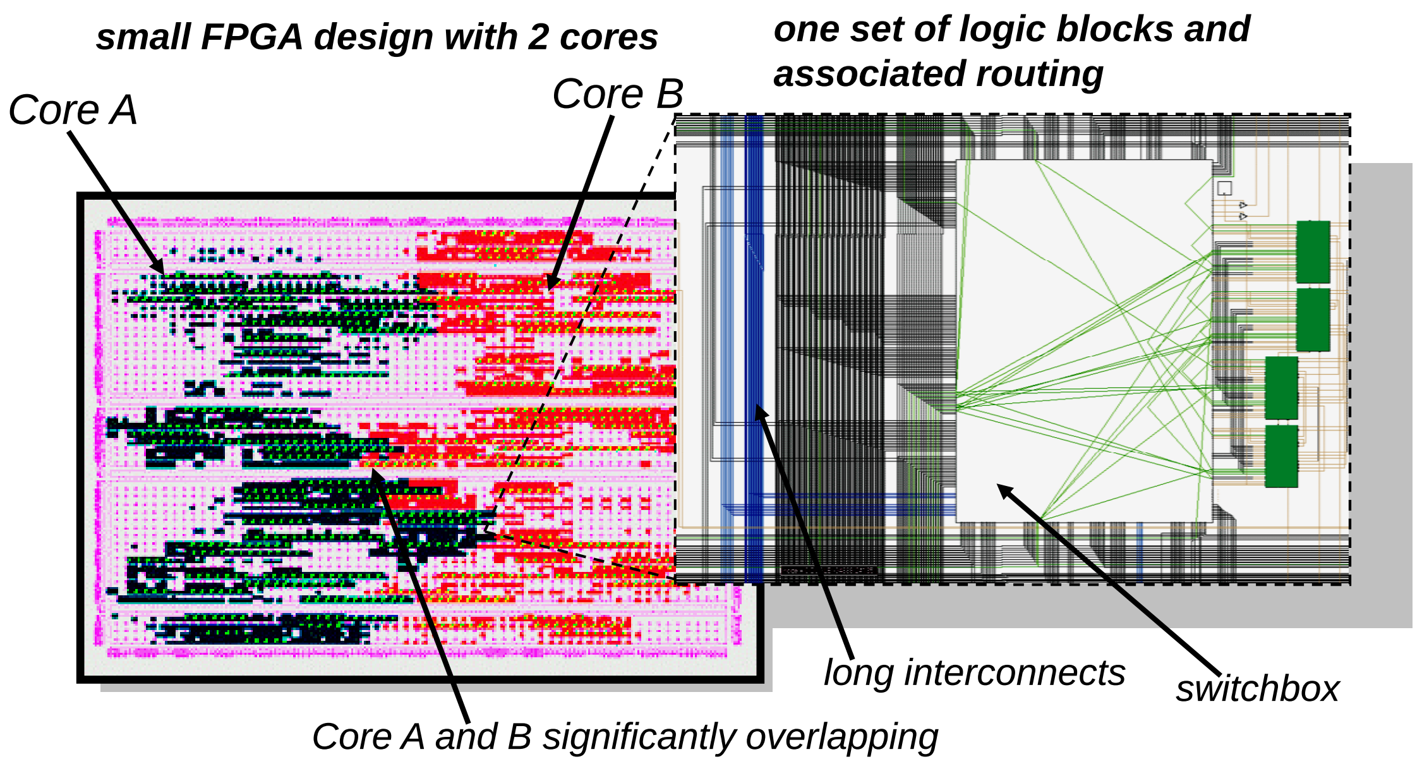
Isolation Option 2: On-Chip Separation Software



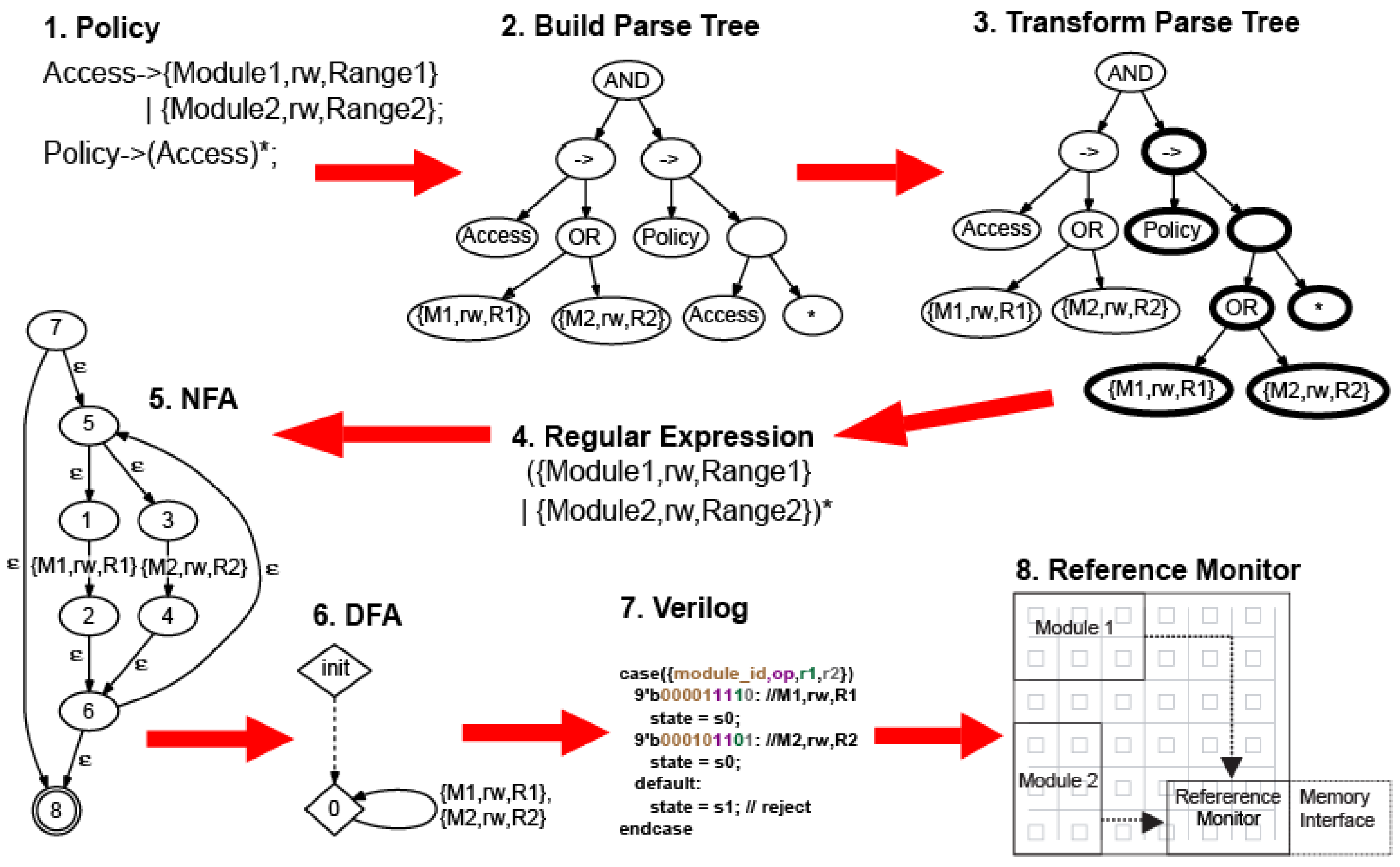
Isolation Option 3: RCsec

Physical Isolation: Moats and Drawbridges

Isolation is imperative to ensure the confidentiality and integrity of the data stored in an IP core. One of the goals of the RCSec Project is to support physical isolation of cores on reconfigurable hardware such as FPGAs. The robust routing structure and ease of reconfigurability on FPGAs makes isolation difficult. To achieve isolation, we propose a solution where each core is surrounded by a "moat" - an area that contains no logic. This ensures that the cores are physically isolated. However the cores must communicate amongst each other and with external peripherals. "Drawbridges" are a precisely defined communication path that cross the moats allowing the cores to communicate. We are developing a tool which allows us to check the bitstream of a design and verify that the moats are only crossed by specified drawbridges. Moats and Drawbridges allow for us to isolate the cores providing integrity and confidentiality of their data.


small FPGA design with 2 cores
one set of logic blocks and associated routing
Core A
Core B
long interconnects
switchbox
Core A and B significantly overlapping

Protecting and Separating Memory: Hardware Reference Monitors



```
case({module_id,op,r1,r2})
  9'b000011110: //M1,rw,R1
    state = s0;
  9'b000101101: //M2,rw,R2
    state = s0;
  default:
    state = s1; // reject
endcase
```

A key element of our isolation strategy is the use of a reconfigurable reference monitor to provide policy-driven memory protection. We have developed a memory protection mechanism capable of enforcing policies expressed as a formal language. A policy is a formal top-level specification of the legal sharing of memory among cores. We have developed a compiler that translates a policy of legal sharing to a hardware description of a reference monitor that enforces the policy. The hardware description is then synthesized into reconfigurable logic, which are directly transferred onto an FPGA. Testing has shown this approach to be efficient in terms of both processing time and space usage on the FPGA.

We are currently developing techniques to prevent the use of the internal states of the reference monitor as covert channels. Since any reference monitor is only as good as the policy it enforces, we are also developing tools for the analysis of candidate policies, and to make the process of expressing policies as precise and user-friendly as possible for embedded systems designers.

Future Research Directions

- **Multi-Core Systems**
  - Techniques to mediate the efficient and secure communication of large numbers of cores on a single chip
- **Further Integration of Security Primitives**
  - Cooperative application of spatial schemes, temporal schemes, and tagging to meet security requirements and minimize overhead
- **Reconfigurable Updates**
  - Understand the security implications of partial reconfiguration as it applies to dynamic updates
- **Channels and Information Leakage**
  - Investigate solutions to the problems of covert channels, side channels, and direct channels in FPGA-based embedded systems

National Science Foundation
WHERE DISCOVERIES BEGIN

CyberTrust PI Meeting
New Haven, CT
March 2008

UCSD
University of California Santa Barbara
NAVAL POSTGRADUATE SCHOOL