**Design Virtualization for Mainstream Reconfigurable Computing**

James Coole, David Wilson, Austin Baylis, and Greg Stitt

### FPGA Development Woes

**RTL Design**
- Requires designers with expertise in digital design and specialized languages

**Low-level Debugging**
- Involves lengthy verification and cycle-by-cycle analysis of signal waveforms

**Limited Reuse**
- Device-specific bitstreams and vendor-specific tool flows inhibit design reuse

**Lengthy Compilation**
- Large designs may take up to days/weeks to place-and-route on FPGAs

### FPGA Overlays

**Virtual coarse-grained architectures intended to alleviate FPGA productivity issues through abstraction**

#### Application Layer
- App users write applications in high-level languages

#### Overlay Abstraction Layer
- Overlay appears to app as a reconfigurable circuit of app-level resources

#### FPGA Layer
- Overlay built on FPGA and abstracts FPGA resources

### Overlay+HLS Design Flow

1. **Application Development**
   - User develops application in high-level language

2. **High-level Synthesis**
   - High-level synthesis (HLS) compiles app into an intermediate representation (IR)

3. **Overlay Selection**
   - HLS selects or synthesizes an overlay that can support IR's resource needs

4. **Overlay PAR**
   - IR is virtually placed-and-routed (PAR) on overlay at runtime

### Results

**Hardware Security**
- Attacker must extract both overlay and FPGA bitstreams to steal IP
- Overlay instance may include countermeasures to protect overlay bitstream
- **Correlated Noise Generation**
  - Correlated noise generation protects overlay bitstream from side channel analysis
  - Correlated Noise Generation (CNG) successfully mitigates CPA attack
  - With CNG, no significant correlation forms
  - Highest correlation is at incorrect subkey guess

**Reliability**
- Use overlay PAR to dynamically and transparently apply redundancy to application circuit for reliability

### Experiments

Evaluation of OpenCL HLS onto overlays compared to RTL synthesis onto FPGAs using several computer-vision OpenCL kernels

- **Between Overlays**
  - Large design space between flexibility and overhead in overlay interconnect
  - Island-style interconnect in Intermediate Fabrics (IFs) enables flexible kernel mappings
  - Tailored interconnect in Supernets are up to 8x smaller than IFs

- **Overlays vs. RTL**
  - Overlay compilation orders of magnitude faster than RTL at reasonable overhead
  - Up to 13x faster compiles than direct-to-FPGA compilation @ 0.15s per kernel
  - 2.5x for system as a whole in 3.6 hours
  - Up to 79% lower area vs. separate datapaths (60% for system)

### Other Features

**Fast Compilation**
- The overlay design process uses multiple-levels of reconfiguration to enable iterative design flows
  - **Virtual Reconfiguration**
    - Overlays designed with innate reconfigurability to enable run-time changes
  - **Context Reconfiguration**
    - Overlay library stores overlay instances with different resources to meet different demands
  - **FPGA Reconfiguration**
    - Resort to synthesis only when there are no overlay instances that can meet requirements

**Design Reuse**
- **Design Portability**
  - Virtual bitstreams are portable to any FPGA that supports its respective overlay instance
- **Compilation Reuse**
  - Reuse overlay compilation runs for different applications with similar resource needs