Reconfigurable Functionality — The OS Perspective

Michael Dales
UNIVERSITY of GLASGOW
Monday 26\textsuperscript{th} June 2000
Motivation

• FPL is good
• Want to bring that goodness to the masses
• Attempts to add FPL into desktop machines
• Lot’s of good from at the low–level:
  – Using FPGA on PCI card
  – FPL and processor on one chip
  – FPL inside ALU of processor
Motivation — Proteus Architecture

- Another attempt to bring FPL inside the processor.

- Proteus Processor:
Motivation — Proteus Architecture

- Uses multiple small RFUs:

- Shared between applications
- Attempt to reduce contention
- Currently building s/w model of ProteanARM
Motivation — What’s currently missing?

- Easy to focus on syntax, not semantics
- Consider problems of desktop environment:

![Diagram](image-url)
Problems — Resource management

- Single large resource is bad
  - Single large FPL block a area for contention
  - Performance crosstalk between applications
- OS virtualises the resources
- Treat virtual circuitry like virtual memory?
Problems — Security

• Break virtual machine — expose hardware to apps.
• Problems:
  – Division of labour between user and kernel mode
  – FPGA Viruses
  – Denial of Service
• Solved partially in Extendable OS research
Problems — Context Explosion

- Context switching requires saving state
- Pure overhead
- More state = more overhead
- Typical modern OS may switch up to 100 times a second
- State in RFUs could affect this dramatically
Problems — Concurrency and Timing

- Concurrency hard to manage
- Loading circuits — what happens if interrupts
  - Delay interrupt?
  - Back off and restart?
Summary

- Lots of issues when consider what the OS needs
- Must support this model for desktop acceptance
- Needs low-level support
- Look at current OS support in processor design