Example Computer Architecture Research Project

Simon Moore

UNIVERSITY OF CAMBRIDGE
Computer Laboratory
Computer Architecture Group

Trustworthy Processor Design

- Motivation
  - Security/Trustworthiness is increasingly important
    - need the hardware to help enforce policies

- Hypothesis
  - Capsicum (next slide) demonstrated that capability based protection is good for fine-grained sandboxing of applications but with a performance cost
  - Hardware based capabilities will allow more security but with less performance overhead

Capsicum

Capsicum: practical capabilities for UNIX

Robert N. M. Watson
University of Cambridge

Jonathan Anderson
University of Cambridge

Ben Laurie
Google UK Ltd.

Kris Kennaway
Google UK Ltd.

Abstract

Capsicum is a lightweight operating system capability and sandbox framework planned for inclusion in FreeBSD 9. Capsicum extends, rather than replaces, UNIX APIs, providing new kernel primitives (sandboxed *capability mode* and *capabilities*) and a userspace sandbox API. These tools support compartmentalisation of monolithic UNIX applications into logical applications, an increasingly common goal supported poorly by discretionary and mandatory access control. We demonstrate our approach by adapting core FreeBSD utilities

significant technical limitations: current OS facilities are simply not

The acc

capability-

Access Co

(MAC). D

other: the

permissio

the object

system po

(e.g. "

USNIX Security
Best paper 2010

Observations from Capsicum

- Software designs that employ the principle of least privilege are neither easily nor efficiently represented in current hardware
- Kernels and programming language runtimes (TCBs) building directly on hardware in C are enormous and unsound
- Software TCB implementations embody artifacts of security policies rather than design principles

CAP Computer (1970s)

Checkered History of Capability Machines

- 1966: Dennis & Van Horn invent the term
- 1972: Plessey System 250 use hardware capabilities commercially
- 1976: Cambridge CAP Computer
- 1979: IBM System/38
- 1981: Intel iAPX 432 – embodiment of CISC

RISC Revolution!

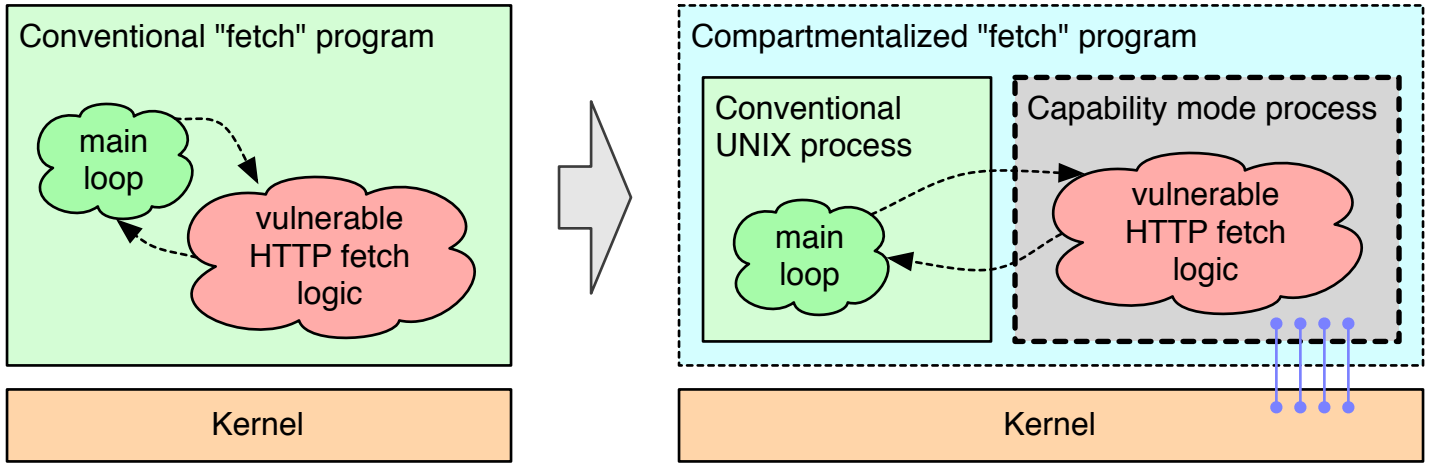
- 1999: EROS uses software capabilities
- 2010: Capsicum capability security model

A RISC approach to capabilities: CHERI

- Base system - our own 64 bit MIPS style core (BERI)
  + extensive regression test suite
- Implemented in Bluespec targeting FPGAs
- Running FreeBSD
  – complete UNIX setup
- Now adding capability mechanisms to hardware and OS

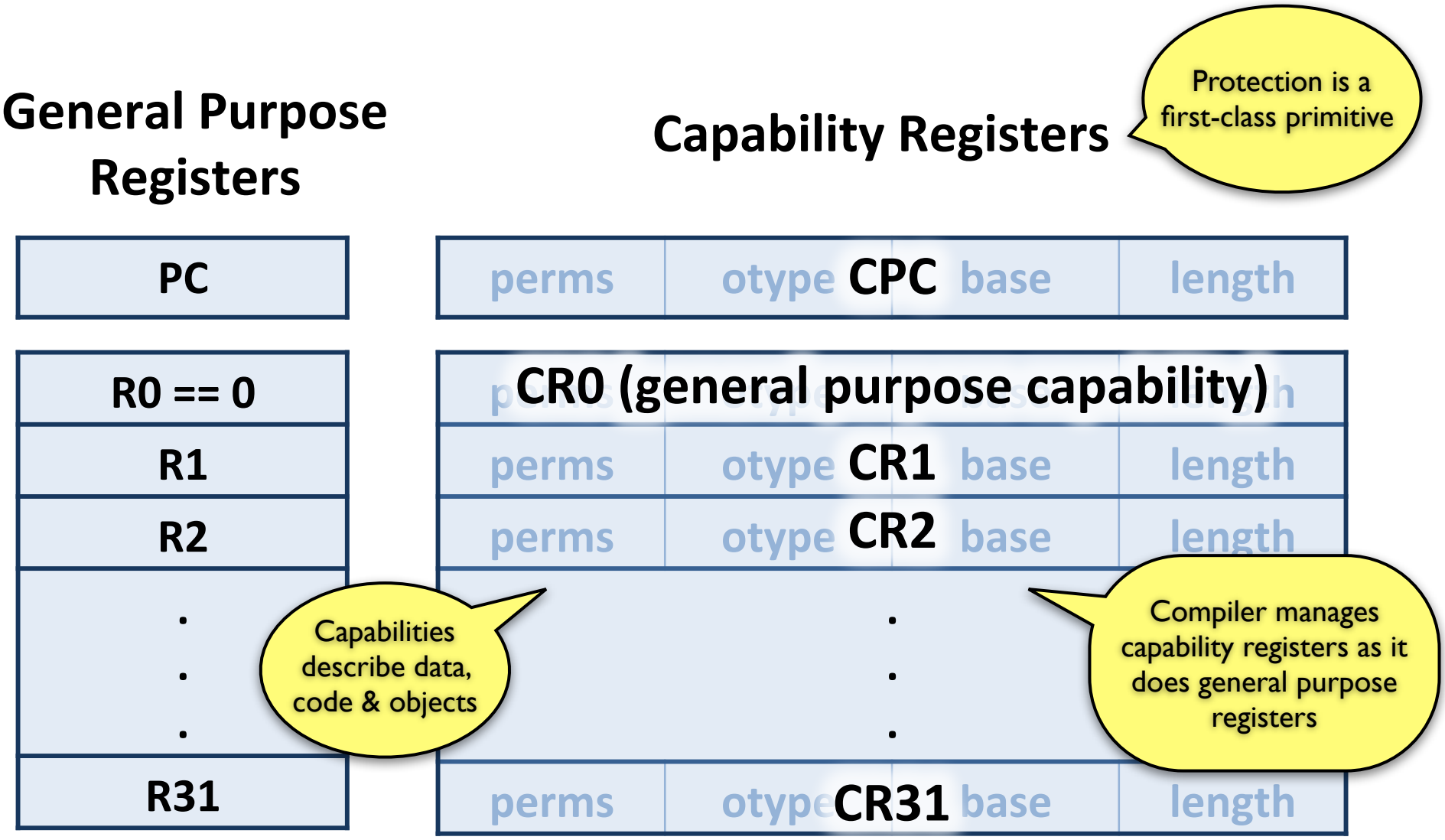
CHERI tablet demo platform

Software compartmentalisation

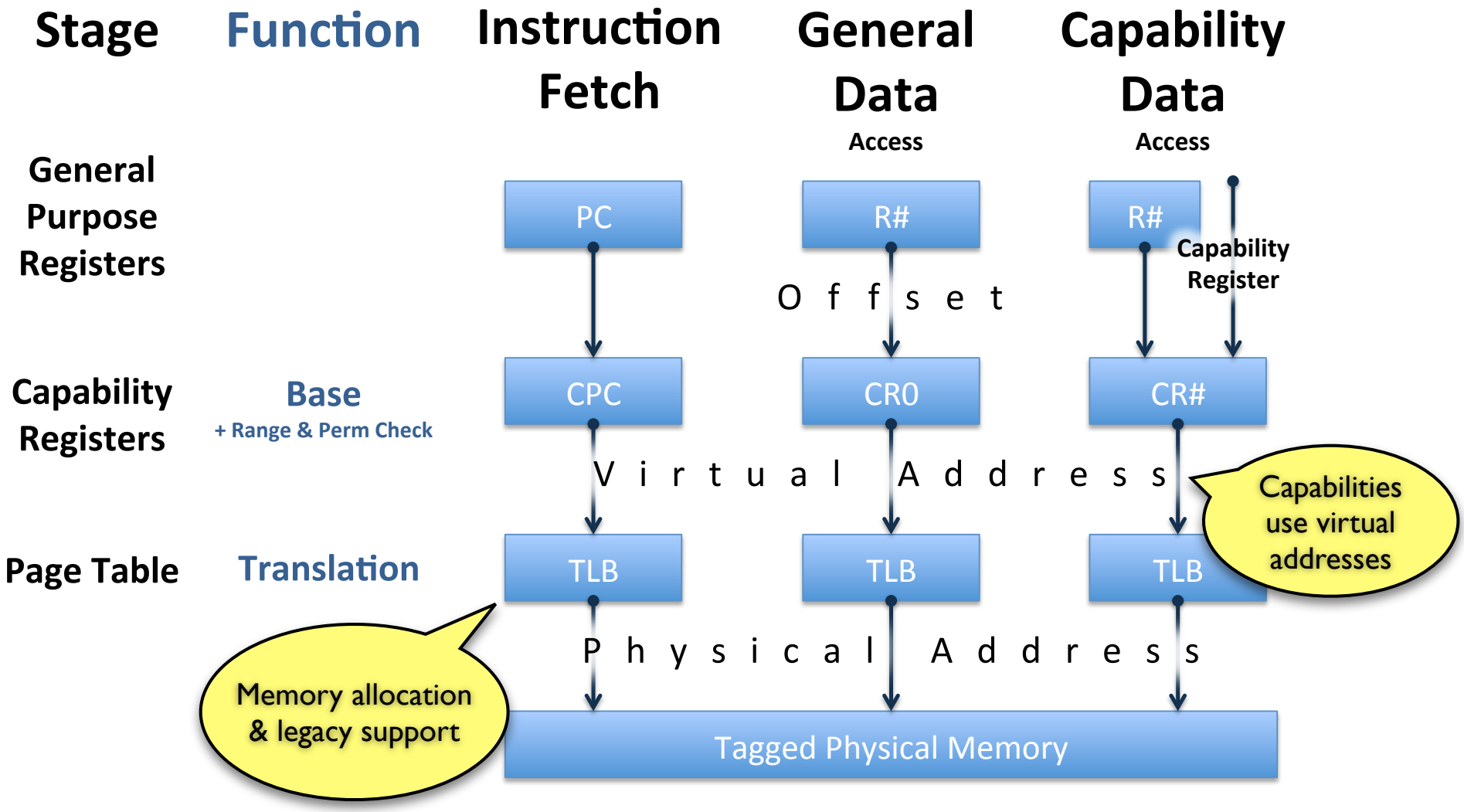


- Software compartmentalisation decomposes applications into many isolated components
- Each running with only the rights required to perform its function
- This implements the principle of least privilege

Capability Register Model



Memory Access



Summary

- CHERI = MIPS + capabilities
- Aiming to show that hardware-based fine-grained protection is a real winner for real applications

Conclusions

- There's lots of open research questions in computer architecture
- FPGAs provide an efficient "sand pit" for computer architecture research

Ph.D. positions, Part II projects, etc.

- Lots of opportunities available for bright well motivated individuals to join the team over the next few years

- Need people interested in:
  - computer architecture
  - operating systems/run-time systems
  - security
  - compilation techniques
  - etc…