int aFunction(const __capability int *x)
{
// This will abort at run time:
*(__capability int) x = 42;
return *x;
}


Capability Hardware Enhanced RISC Instructions (CHERI) provide a fine-grained protection system to address the problem of complementary existing virtual memory-based systems by providing efficient and more programmatically support for applications compartmentalization.

• Uses a reduced instruction set computer (RISC) approach, providing tools for compiler and operating system platform while minimizing hardware complexity.

• Designed to support systems that incorporate memory-mapped I/O, allowing high-level programming to use memory-mapped I/O.

• Provides efficient support for accessing different OS applications, reflecting diverse OS, programming languages, and application requirements.

CHERI and CHERI2 hardware prototypes.

We have developed a fine-grained protection system based on hardware-supported address space protection, allowing read and write access to protected memory regions.

• Uses a reduced instruction set computer (RISC) approach, providing tools for compiler and operating system platform while minimizing hardware complexity.

• Provides efficient support for accessing different OS applications, reflecting diverse OS, programming languages, and application requirements.

CHERI ISA-level testing and verification

CHERI ISA-level testing and verification includes both the bit and ISA level. This requires significant setup and infrastructure.

We have created a PVS model of the CHERI ISA to reason about security properties and check the expected behavior. We have developed an extensive ISA test set, which includes thousands of tests. We have implemented a fuzzing suite to detect hardware flaws and security exploits.

Formal verification of the CHERI microprocessor

In collaboration with Bluescale, Inc, we have provided a translation from BVS intermediate representations to a high-level Modelica-based constraint language.

This provides a way to reason about design that is more efficient and expressive than the CHERI ISA-level testing and verification.

We have developed an extensive ISA level test suite, which includes thousands of tests. We have implemented a fuzzing suite to detect hardware flaws and security exploits.

An incrementally adoptable hybrid capability system model

CHERI evaluates capabilities prior to MMU virtual address translation, giving such UNIS processes, as well as higher security contexts. Conventional untrusted software is reflected in a 1990s design consensus regarding virtualization and protections.

A further advantage of CHERI is that it introduces less total memory use than existing systems. This allows for significantly faster and more efficient software development.

CHERI extensions to Capsicum

We have added compiler built-in support for CHERI features to the Capsicum intermediate representation, which is a high-level Modelica-based constraint language.

This provides a way to reason about design that is more efficient and expressive than the CHERI ISA-level testing and verification.

We have developed an extensive ISA level test suite, which includes thousands of tests. We have implemented a fuzzing suite to detect hardware flaws and security exploits.

CHERI extensions to LAVM/LVM

We have added compiler built-in support for LAVM/LVM extensions, which are a high-level Modelica-based constraint language.

This provides a way to reason about design that is more efficient and expressive than the CHERI ISA-level testing and verification.

We have developed an extensive ISA level test suite, which includes thousands of tests. We have implemented a fuzzing suite to detect hardware flaws and security exploits.

CHERIBSID: Adapting UNIX for hardware capabilities

CHERIBSID is a framework for adapting UNIX for hardware capabilities. The framework provides a portable implementation of the CHERI extension techniques to support UNIX.

• Provides a portable implementation of the CHERI extension techniques to support UNIX.

• Provides a portable implementation of the CHERI extension techniques to support UNIX.

• Provides a portable implementation of the CHERI extension techniques to support UNIX.

In summary, we have developed a fine-grained protection system based on hardware-supported address space protection, allowing read and write access to protected memory regions.

We have created a PVS model of the CHERI ISA to reason about security properties and check the expected behavior. We have developed an extensive ISA test set, which includes thousands of tests. We have implemented a fuzzing suite to detect hardware flaws and security exploits.

Formal verification of the CHERI microprocessor

In collaboration with Bluescale, Inc, we have provided a translation from BVS intermediate representations to a high-level Modelica-based constraint language.

This provides a way to reason about design that is more efficient and expressive than the CHERI ISA-level testing and verification.

We have developed an extensive ISA level test suite, which includes thousands of tests. We have implemented a fuzzing suite to detect hardware flaws and security exploits.

An incrementally adoptable hybrid capability system model

CHERI evaluates capabilities prior to MMU virtual address translation, giving such UNIS processes, as well as higher security contexts. Conventional untrusted software is reflected in a 1990s design consensus regarding virtualization and protections.

A further advantage of CHERI is that it introduces less total memory use than existing systems. This allows for significantly faster and more efficient software development.

CHERI extensions to Capsicum

We have added compiler built-in support for CHERI features to the Capsicum intermediate representation, which is a high-level Modelica-based constraint language.

This provides a way to reason about design that is more efficient and expressive than the CHERI ISA-level testing and verification.

We have developed an extensive ISA level test suite, which includes thousands of tests. We have implemented a fuzzing suite to detect hardware flaws and security exploits.

CHERI extensions to LAVM/LVM

We have added compiler built-in support for LAVM/LVM extensions, which are a high-level Modelica-based constraint language.

This provides a way to reason about design that is more efficient and expressive than the CHERI ISA-level testing and verification.

We have developed an extensive ISA level test suite, which includes thousands of tests. We have implemented a fuzzing suite to detect hardware flaws and security exploits.