A Fast Low-Level Error Detection Technique

Zhengyang He, Hui Xu, Guanpeng Li

Reza Adinepour

Amirkabir University of Technology
(Tehran Polytechnic)

Computer Engineering Department

January 10, 2025

Agenda

1. Introduction

    Problem & Solutions Overview

    EDDI Methods

    EDDI Methods (Cont.)

    Example Code

2. Main Contribution

3. Background

    Background

    Figure

4. Mathematics

5. Referencing

Problem & Solutions Overview

- **Problem:** Transient hardware faults (soft errors) due to shrinking transistor sizes and operating voltages.
- **Impact:** Soft errors can cause Silent Data Corruptions (SDCs), compromising system dependability.
- **Solutions:**
  1. Traditional: Hardware-based methods such as:
     - voltage guard bands
     - redundancy

     have high overhead in performance and energy consumption.
  2. Software-Based: Error Detection by Duplicating Instructions (EDDI)
     has been proposed as a flexible, resource-efficient alternative.

EDDI Methods

- **EDDI:** Duplicates instructions at compile time and checks for mismatches at runtime.

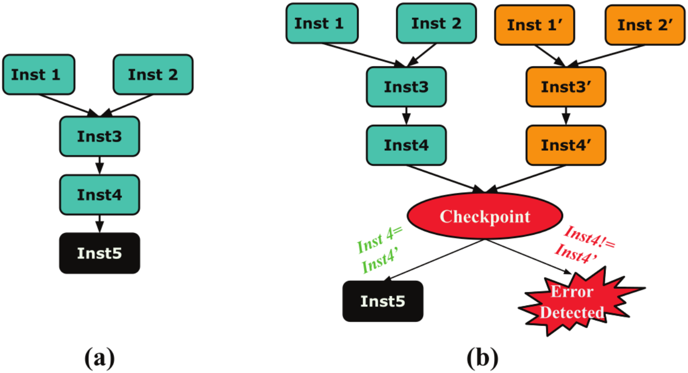


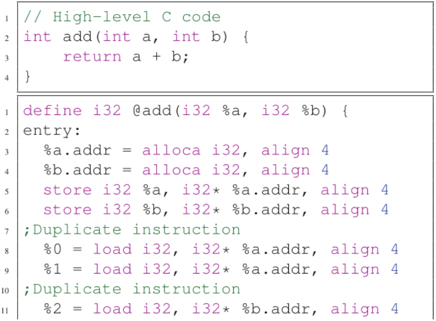
Figure: High-level idea of EDDI

EDDI Methods (Cont.)

- **Existing EDDI Methods:**
  1. Mostly at IR level

  reduced fault coverage when tested at the assembly level.
- **Problem with IR-Level EDDI:**
  - Fault coverage gaps at IR level.
  - Reduced effectiveness when evaluated at assembly level.
  - Underestimated error detection at lower levels.
  - Need for assembly-level implementation for better fault protection.

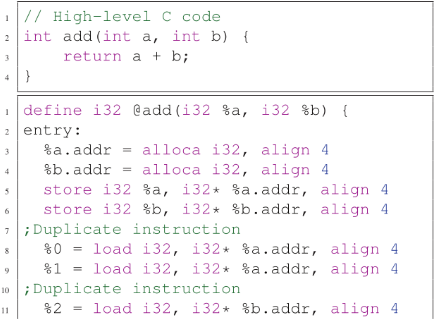
IR Code Example Using EDDI



```
// High-level C code
int add(int a, int b) {
    return a + b;
}
```

```
define i32 @add(i32 %a, i32 %b) {
entry:
  %a.addr = alloca i32, align 4
  %b.addr = alloca i32, align 4
  store i32 %a, i32* %a.addr, align 4
  store i32 %b, i32* %b.addr, align 4
;Duplicate instruction
  %0 = load i32, i32* %a.addr, align 4
  %1 = load i32, i32* %a.addr, align 4
;Duplicate instruction
  %2 = load i32, i32* %b.addr, align 4
```

Figure: (a)



```
// High-level C code
int add(int a, int b) {
    return a + b;
}
```

```
define i32 @add(i32 %a, i32 %b) {
entry:
  %a.addr = alloca i32, align 4
  %b.addr = alloca i32, align 4
  store i32 %a, i32* %a.addr, align 4
  store i32 %b, i32* %b.addr, align 4
;Duplicate instruction
  %0 = load i32, i32* %a.addr, align 4
  %1 = load i32, i32* %a.addr, align 4
;Duplicate instruction
  %2 = load i32, i32* %b.addr, align 4
```

Figure: (b)

Main Contribution

1. **Proposed Solution:**
   - FERRUM: Optimized assembly-level EDDI.
   - Enhancements: Utilizes SIMD and compiler optimizations.
   - Improves: Fault coverage and performance.
2. **Key Findings & Results:**
   - 28% gap in fault coverage (IR-level vs. assembly-level).
   - 100% fault coverage with FERRUM at assembly level.
   - Higher overhead in assembly-level EDDI; optimizations reduce it.
   - 52% reduction in runtime overhead with FERRUM, no loss in fault coverage.

Background

1. **Fault Model:** Single bit-flip faults; memory protected by ECC.
2. **Fault Simulation:** Assembly-level fault injection; beam testing infeasible.
3. **EDDI:** Instruction duplication, runtime comparison.
4. **Compilation:**
    - IR-level: Common, uses LLVM tools.
    - Assembly-level: Rare, closer to hardware.
5. **Platform:** x86 ISA.

Figure

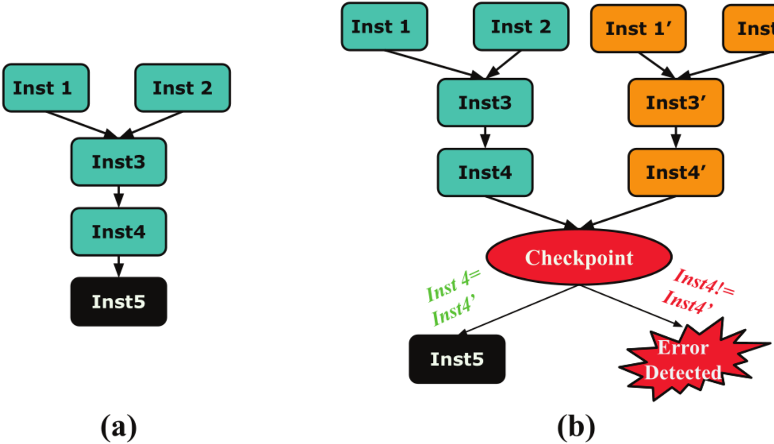


Figure: Creodocs logo.

Definitions & Examples

Definition

A prime number is a number that has exactly two divisors.

Example

- 2 is prime (two divisors: 1 and 2).
- 3 is prime (two divisors: 1 and 3).
- 4 is not prime (three divisors: 1, 2, and 4).

You can also use the `theorem`, `lemma`, `proof` and `corollary` environments.

Theorem, Corollary & Proof

Theorem (Mass–energy equivalence)

$E = mc^2$

Corollary

$x + y = y + x$

Proof.

$\omega + \phi = \epsilon$ □

Equation

$$\cos^3\theta = \frac{1}{4}\cos\theta + \frac{3}{4}\cos 3\theta \tag{1}$$

Verbatim

Example (Theorem Slide Code)

```
^^I^^I^^I\begin{frame}
^^I^^I^^I^^I\frametitle{Theorem}
^^I^^I^^I^^I\begin{theorem}[Mass--energy equivalence]
^^I^^I^^I^^I^^I$E = mc^2$
^^I^^I^^I^^I\end{theorem}
^^I^^I\end{frame}
```

Slide without title.

Citing References

An example of the `\cite` command to cite within the presentation:

This statement requires citation [Smith, 2022, Kennedy, 2023].

References

John Smith (2022)
Publication title
Journal Name 12(3), 45 – 678.

Annabelle Kennedy (2023)
Publication title
Journal Name 12(3), 45 – 678.

Acknowledgements

Smith Lab

- Alice Smith
- Devon Brown

Cook Lab

- Margaret
- Jennifer
- Yuan

Funding

- British Royal Navy
- Norwegian Government

The End

Questions? Comments?