Demystifying and Mitigating Cross-Layer Deficiencies of Soft Error Protection in Instruction Duplication

Zhengyang He[†] Yafan Huang[†] Hui Xu[◊]

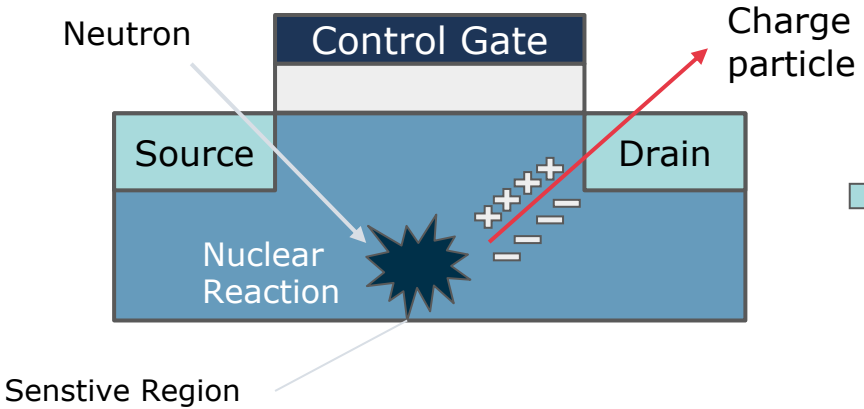
Dingwen Tao[‡] Guanpeng Li[†]

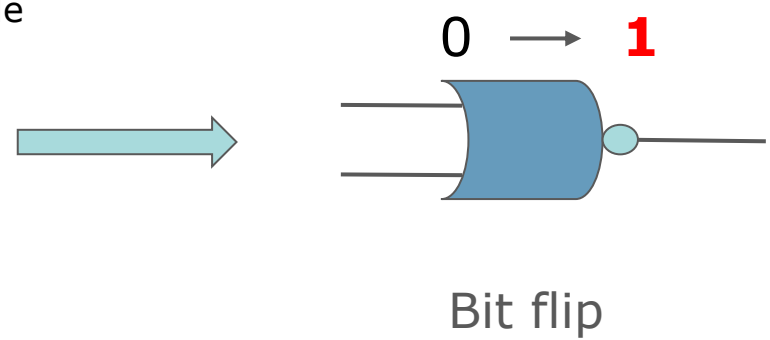
† The University of Iowa

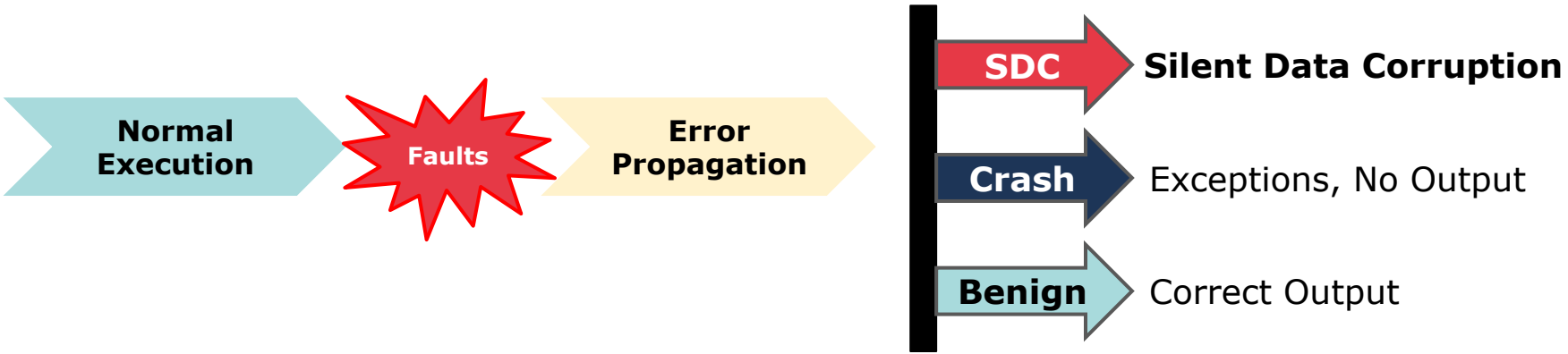
◊Fudan University

‡ Indiana University Bloomington

Soft Errors

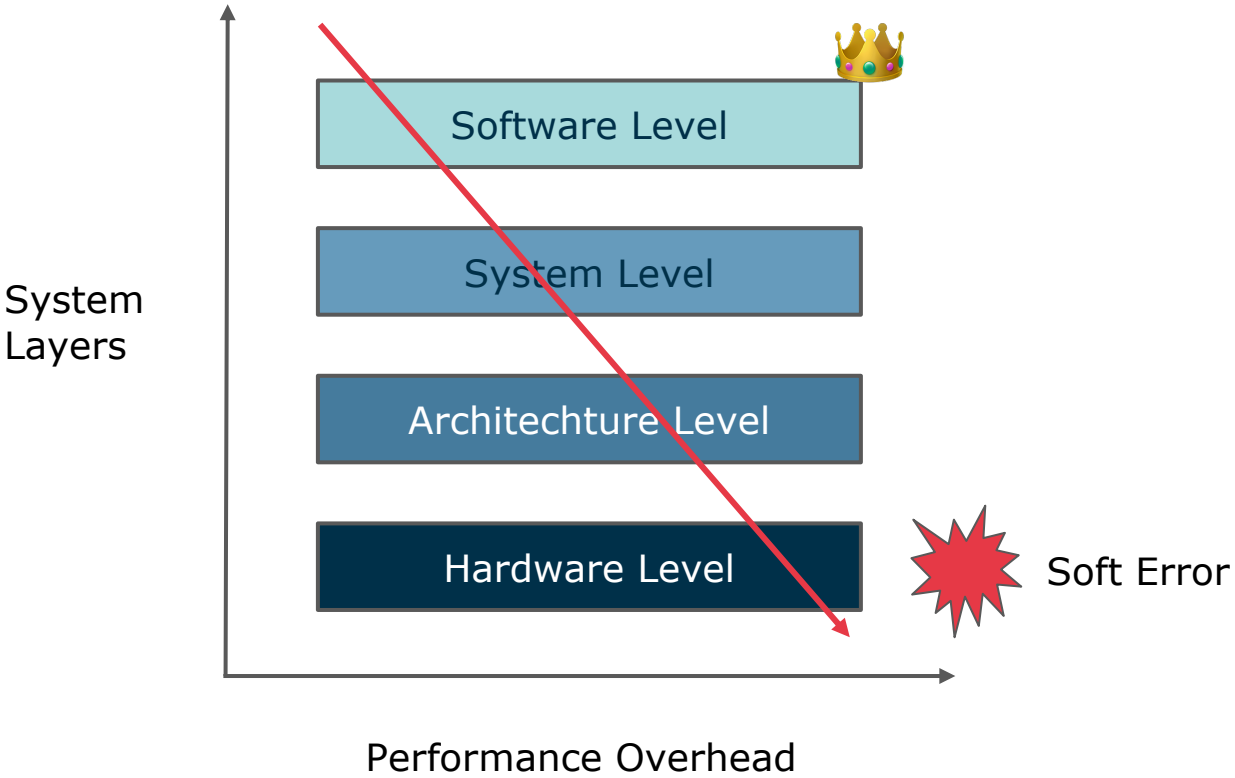

Neutron
Control Gate
Charge particle
Source
Drain
Nuclear Reaction
Senstive Region


0
1
Bit flip


Normal Execution
Faults
Error Propagation
SDC
Silent Data Corruption
Crash
Exceptions, No Output
Benign
Correct Output

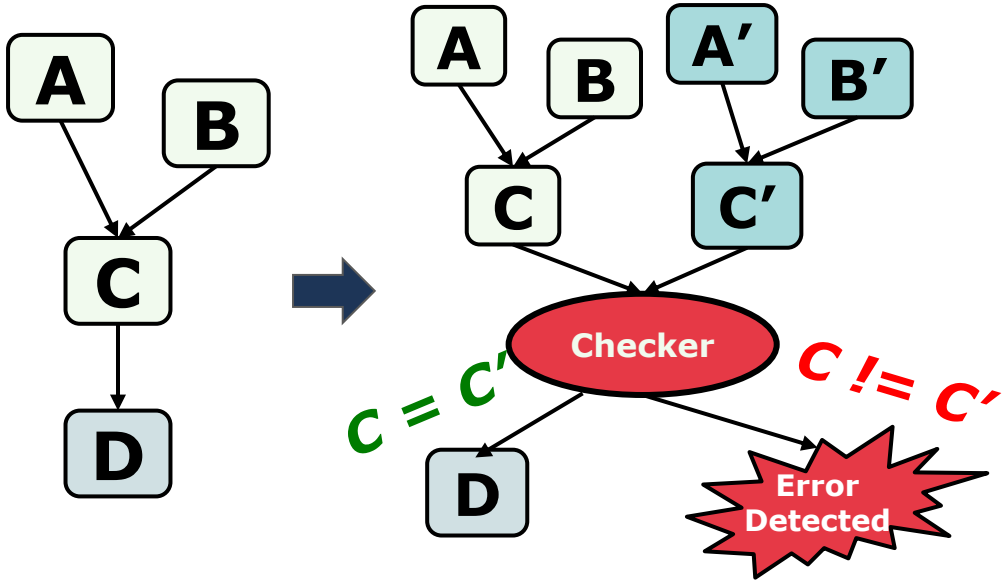
Software Solutions

- **Software solution** is more flexible and cost-effective.



Error Detection by Duplicating Instructions (EDDI)

- **EDDI** duplicates instruction at *compile time* and detects errors at *run time*.
- Compiler-level transformation, hence **program-agnostic**.



original code EDDI-protected code

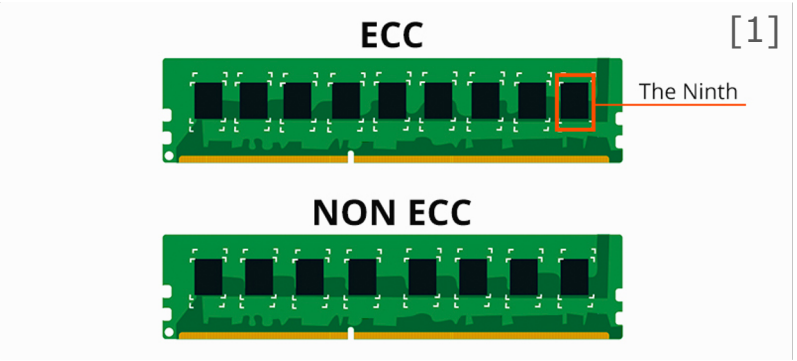
LLVM IR Level ✅

V.S.

Assembly Level

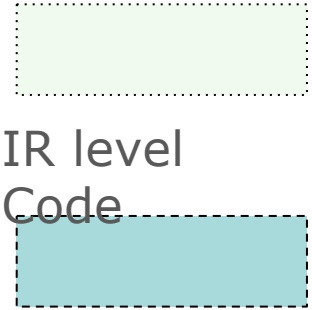
Fault Model

- Single-bit flip, which is accurate enough to evaluate SDC
- Errors in computation units/data path
- One fault per program execution
- Memory errors can be protected by ECC, so do not consider



[1]: https://community.fs.com/article/ecc-vs-non-ecc-memory-which-one-is-better.html

Fault Injection Methodology



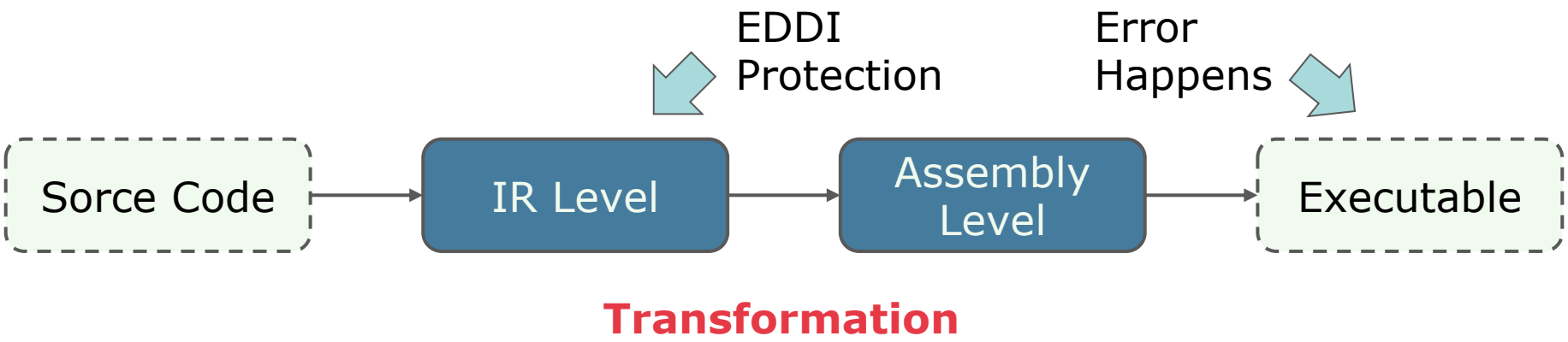
- ***IR***: Instructions that contains return value

```
%3 = icmp slt i32 %1, %2
%4 = load i32* %1, align 4
%5 = mul i64 1, %4
                    …
```

- ***Assembly***: Instructions whose computation destination is a register

```
test   $0x1,%al
sub    %ecx,%eax
mov  -0x40(%rbp),%rax
                    …
```

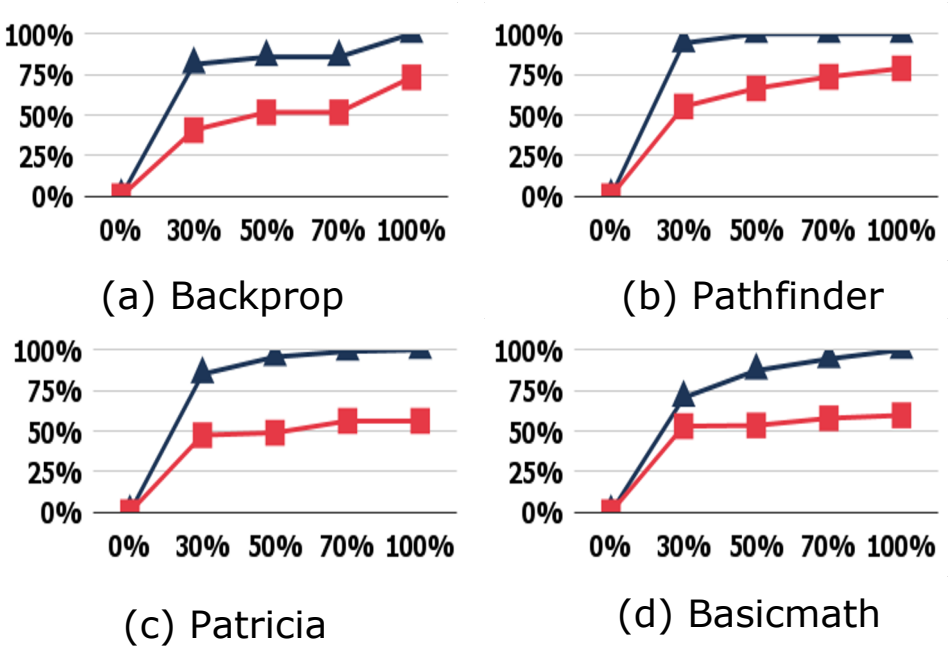
Motivation: Code transformation from IR to Assembly



IR-level EDDI may not provide full protection on Assembly-level!

Motivation: Code transformation from IR to Assembly

- Compare of: SDC coverage evaluation at IR and Assembly level



X-axis : Protection level

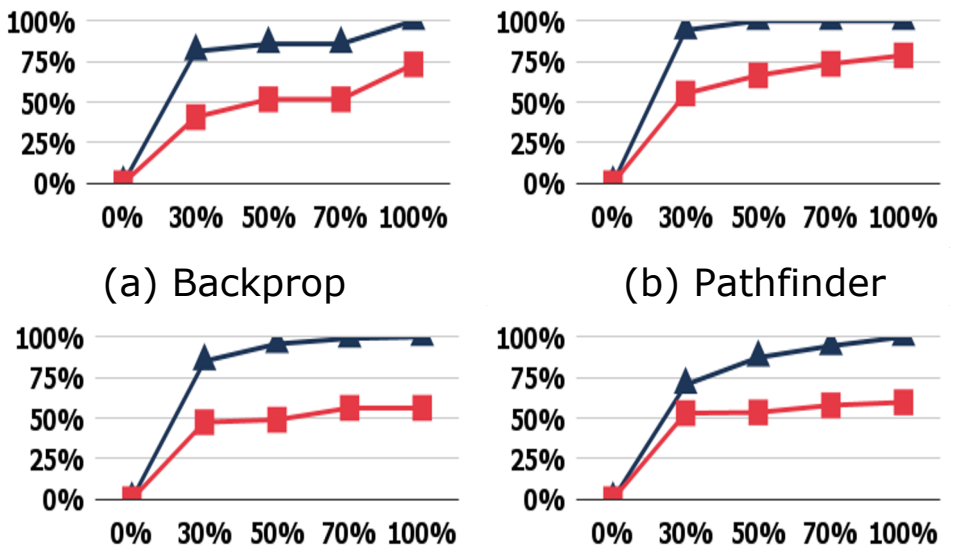
Y-axis : SDC coverage

Assembly level

IR level

Motivation: Code transformation from IR to Assembly

- Compare of: SDC coverage evaluation at IR and Assembly level



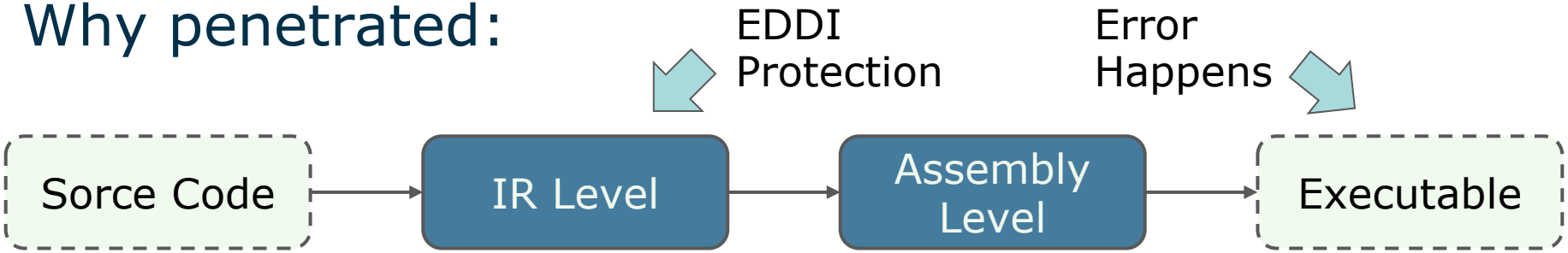
- **Application specific** SDC coverages
- **SDC coverage gap** in IR and Assembly protection

Assembly level

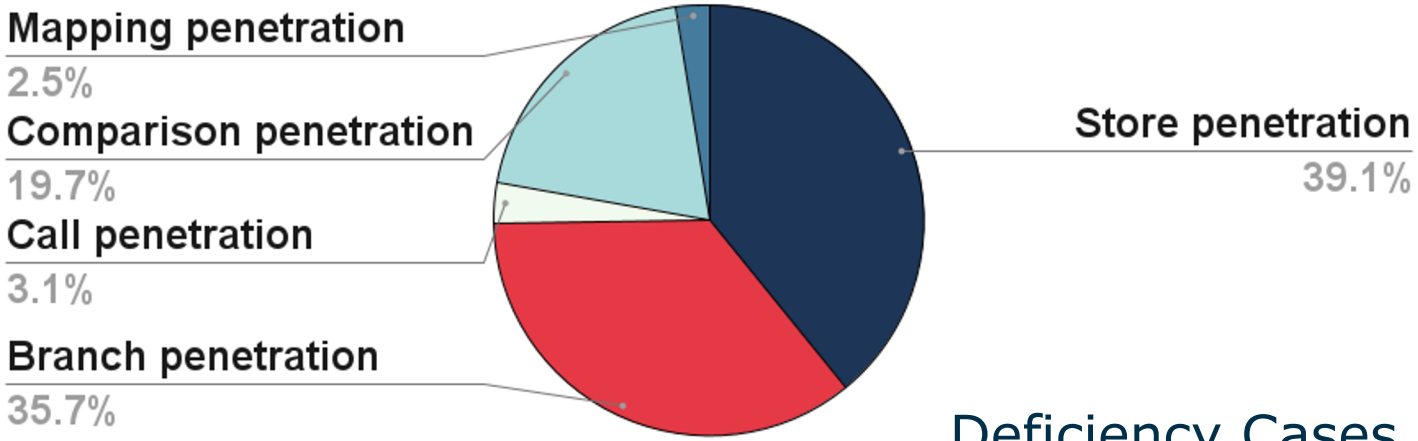
IR level

LLVM SDC coverage often **over-estimates** Assembly level benefits.

Why penetrated:

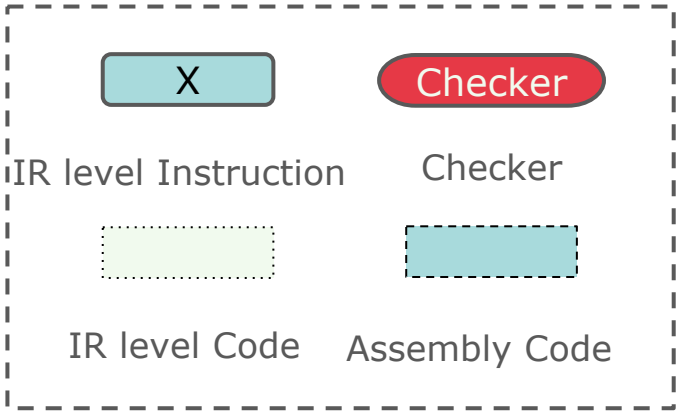


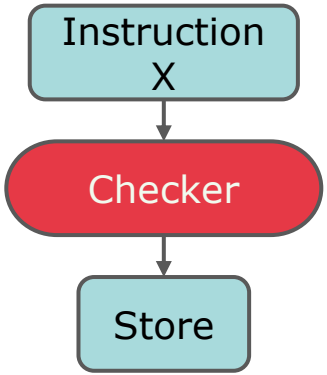
Transformation

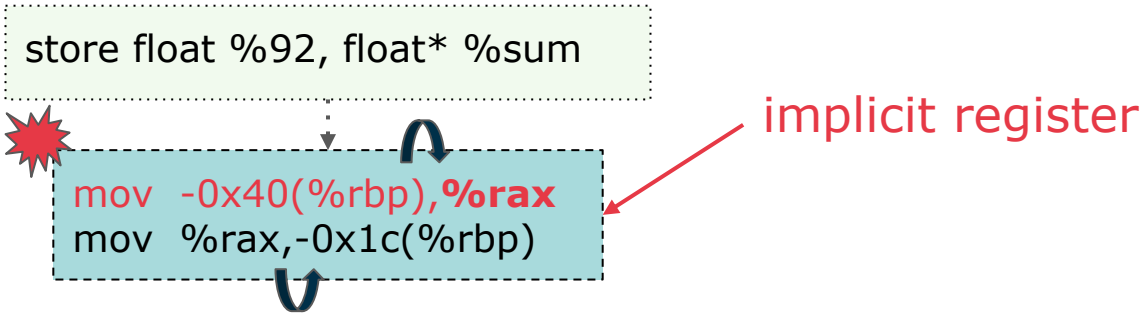


Deficiency Cases

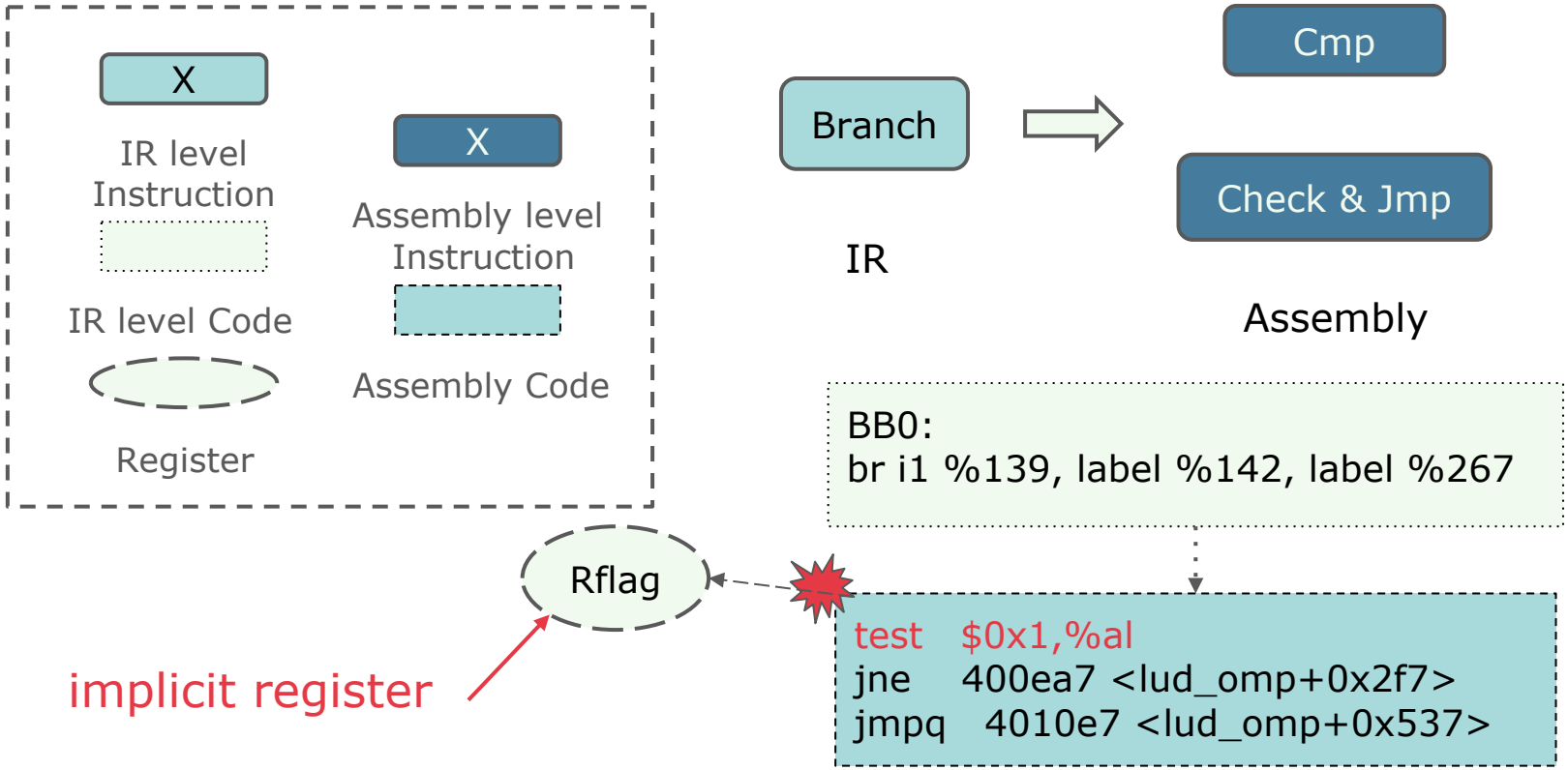
Why penetrated: **1. Store Penetration**



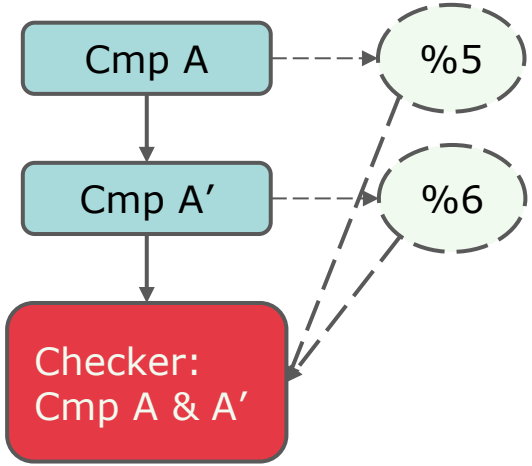


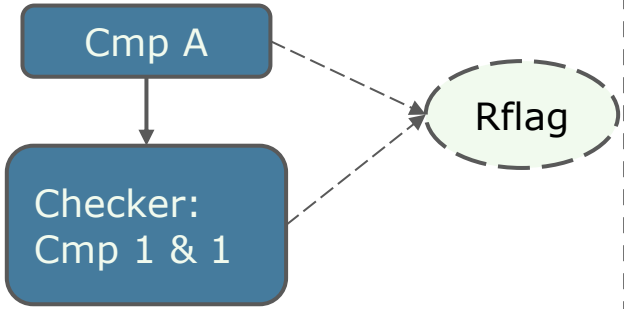


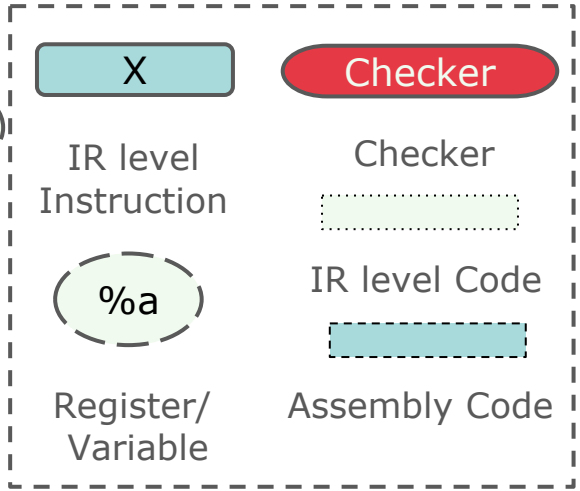
Why penetrated: **2. Branch Penetration**



Why penetrated: **3. Comparison Penetration**







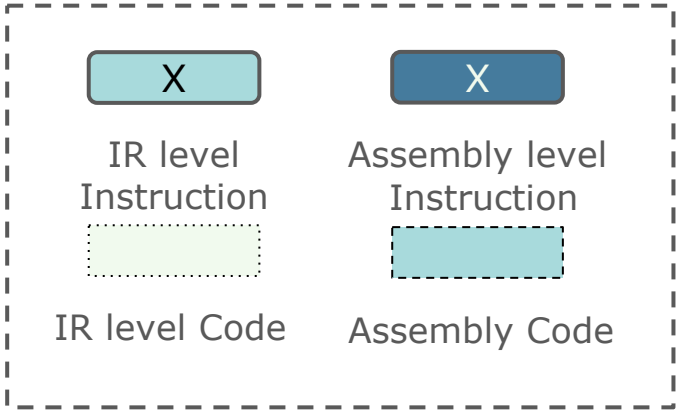
IR

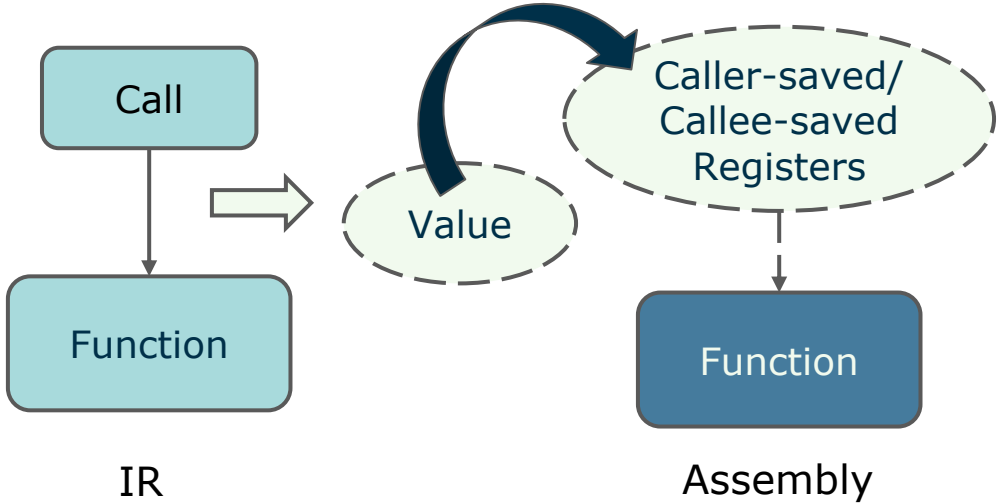
```
%5 = icmp slt i32 %1, %3
%6 = icmp slt i32 %2, %4
%check_cmp = icmp eq i1 %5, %6
```

Assembly

```
sub    %ecx,%eax
setl   %dl
mov    $0x1,%sil
test   %sil,%sil
```

Why penetrated: **4. Call Penetration**

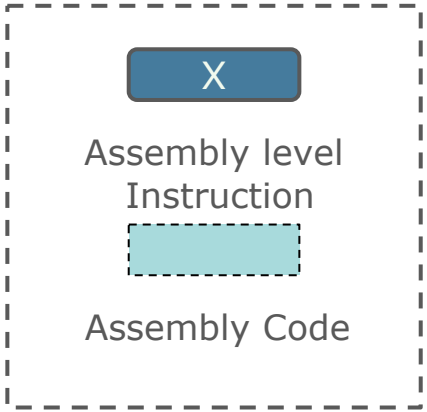


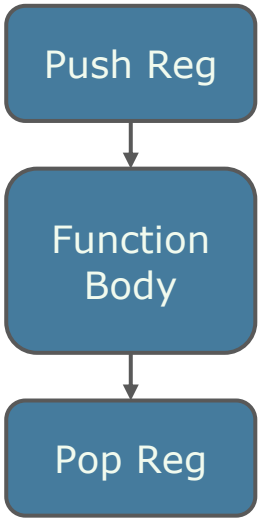


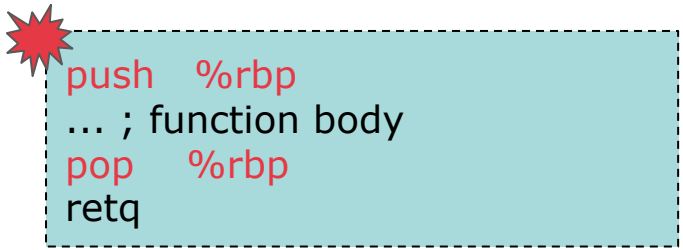

```
call void @_Z3runiPPc(i32 %4, i8** %7)
```

```
mov    -0x14(%rbp),%edi
mov    -0x20(%rbp),%rsi
callq  400f70 <_Z3runiPPc>
```

Why penetrated: **5. Mapping Penetration**

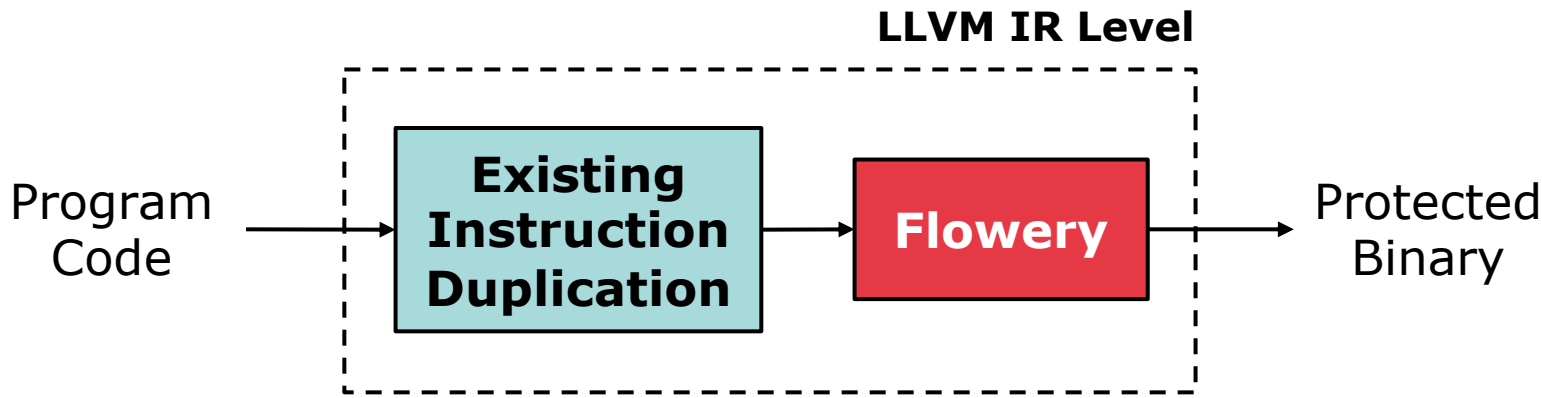






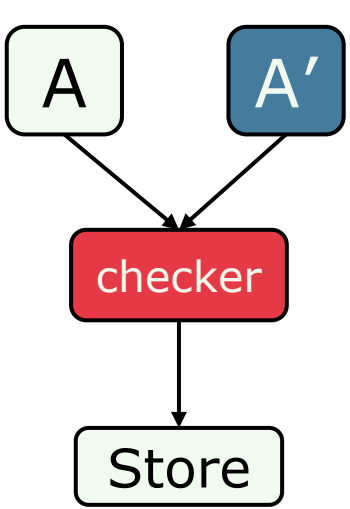
Flowery 🌸

- Goal: **Boost** assembly level protection via IR level
- Obsevations:
  - IR level offers simple instruction **tracking and modification**
    - IR level can alter assembly-level **register allocation**

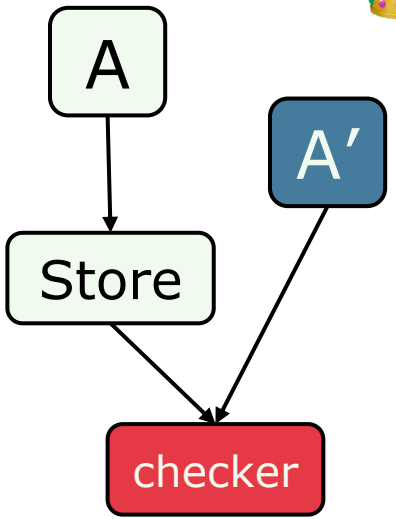


Flowery

- Eager Mode of Store

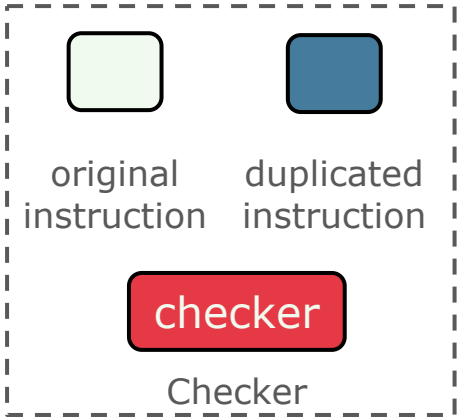


Original EDDI



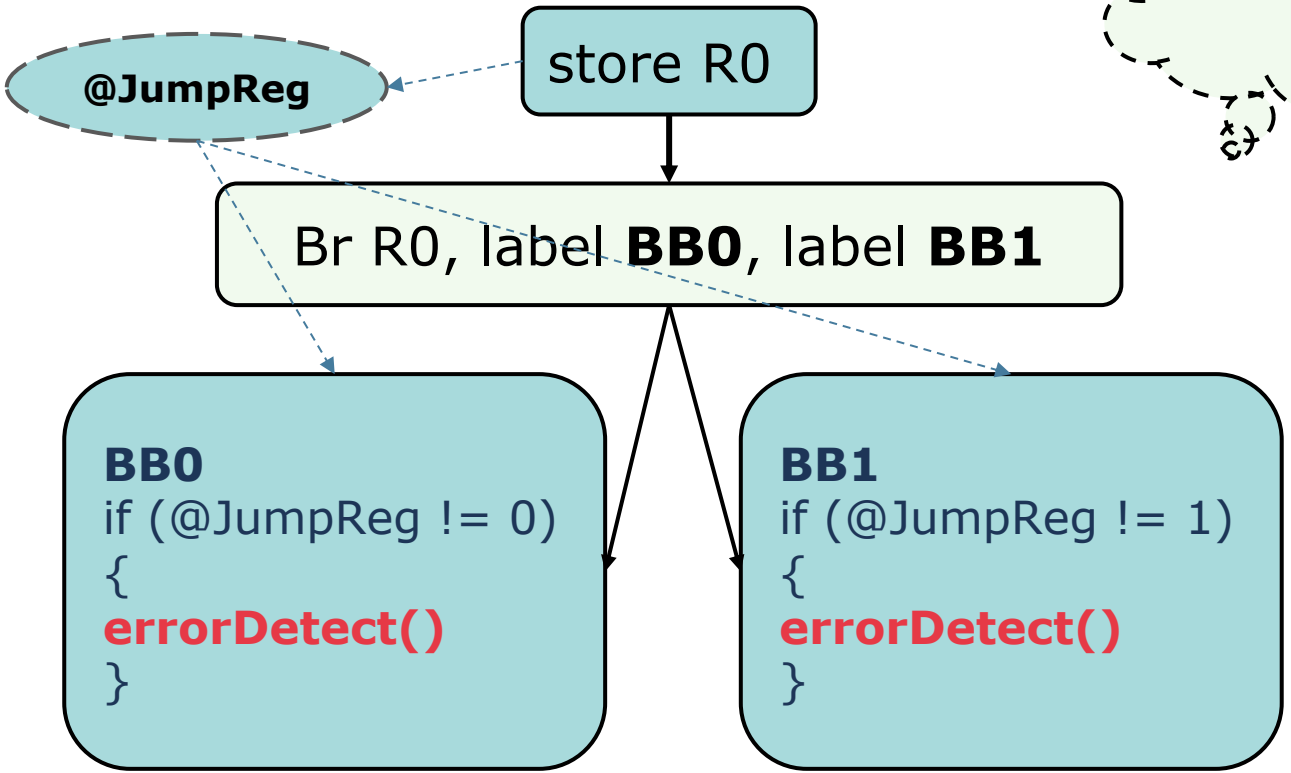
Eager Mode Checker Position

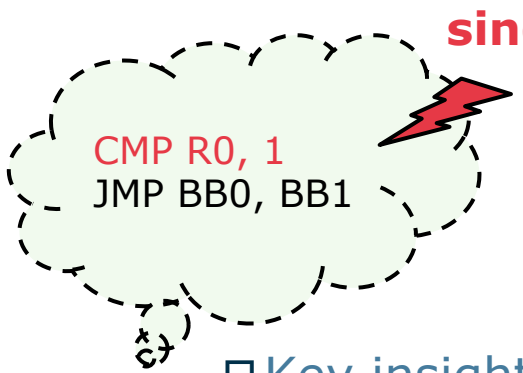
- Key insight: Reallocate the store instruction.
- Result: Move the temporary value to a register without extra computations.



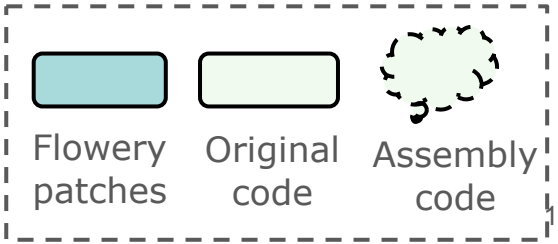
Flowery

- Postponed Branch Condition Check



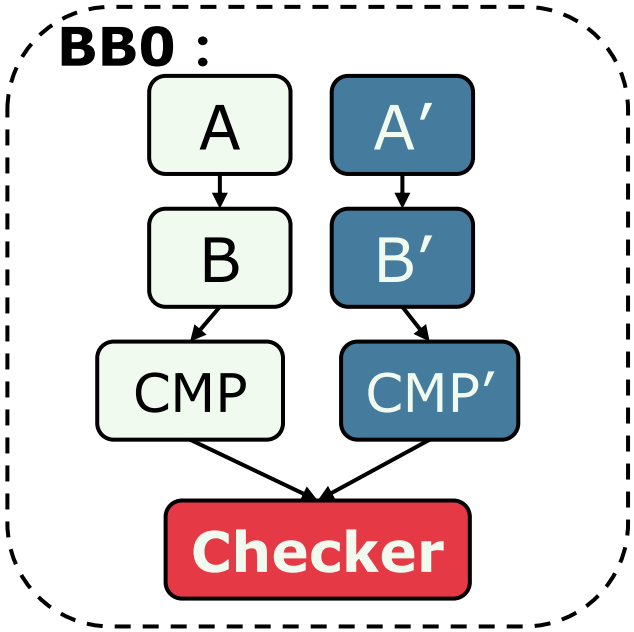


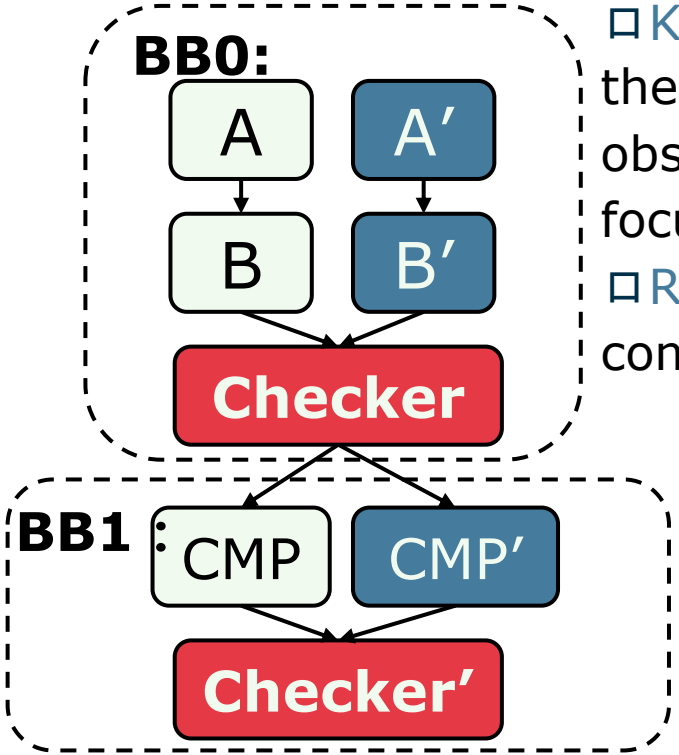
- Key insight: Check branch flag after it jumps.
- Result: Soft errors occur at branch can be detected.



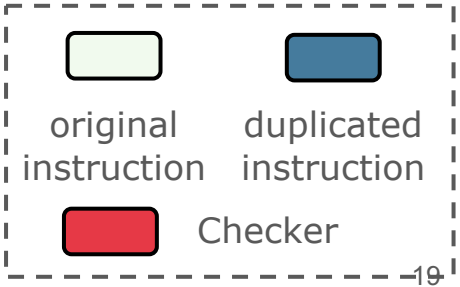
Flowery

- Anti-Comparison Duplication Optimization 👑





- Key insight: Disrupt the dependency chain to obstruct comparison-focused optimization.
- Result: Protect all comparison penetration.

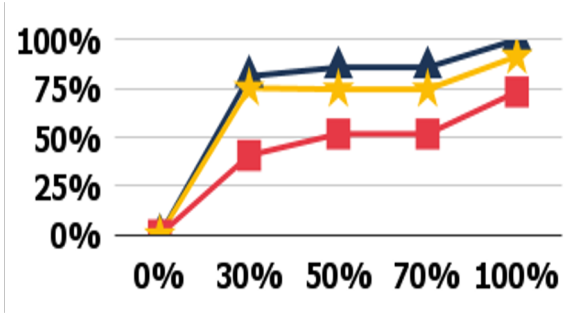


Evaluation

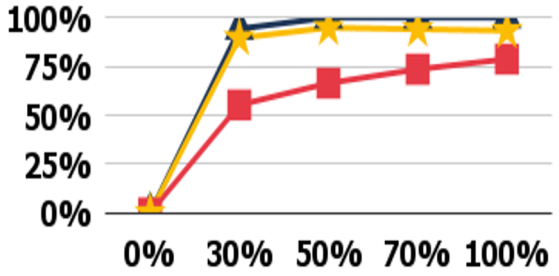
- **SDC coverage**

Optimized assembly level
Assembly level
IR level

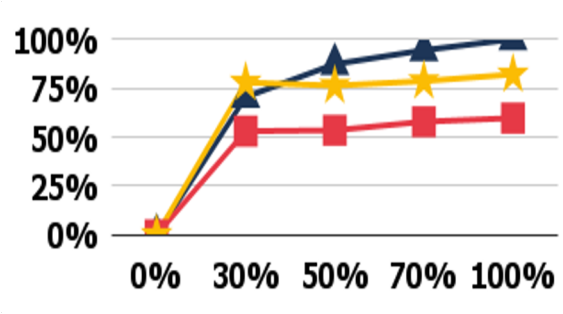
X-axis : protection level
Y-axis : SDC coverage



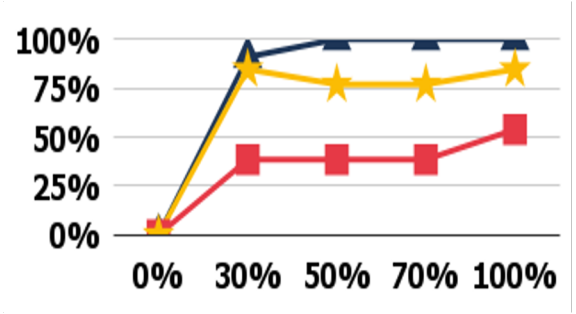
(a) Backprop



(b) Pathfinder



(d) Basicmath



(c) Patricia

EDDI Protection efficiency is significantly enhanced

Conclusion

- We observe **EDDI protection deficiencies** across IR and Assembly levels.
- There are **5 penetration cases** responsible for such deficiencies.
- We propose **Flowery**, which is a set of IR-level modifications.
- Flowery can **mitigate such deficiencies** with **no obvious overhead**.
- Open source: https://github.com/hyfshishen/SC23-FLOWERY

IOWA

FUDAN UNIVERSITY