Security Coverage Metrics for Information Flow at the System Level

Ece Nur Demirhan Coskun**

Sallar Ahmadi-Pour[U]

Muhammad Hassan[U]*

Rolf Drechsler[U]*

*Cyber-Physical Systems, DFKI GmbH
[U]Institute of Computer Science, University of Bremen
*ece.coskun@dfki.de

January 25, 2024
Incheon, South Korea

Bundesministerium für Bildung und Forschung

ECXL 01IW22002

SASPIT 16KIS1852K

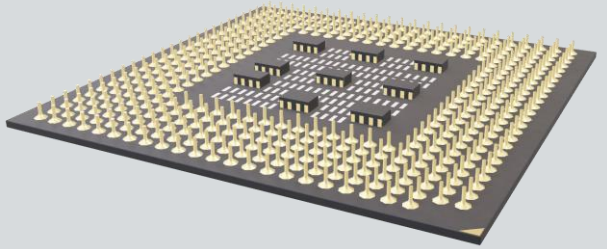
Scale4Edge 16ME0127

PaSVer 16ME0855

Overview of Presentation Content

- **Motivation**
  - ➢Security aspect of modern System-on-Chip (SOC) designs, Vulnerabilities
- **Preliminaries**
  - ➢Information Flow Tracking (IFT), Completeness Driven Development, Virtual Prototypes
  - ➢Threat Model
- **Aim**
  - ➢Security Coverage Metrics for System Level Information Flow
- **Methodology**
  - ➢Metrics Definition, Metrics Implementation: **SiMiT** (**S**tat**i**c + Dyna**mi**c IF**T**) Tool
- **Experimental Results**
  - ➢Car Engine Immobilizer with open source available RISC-V VP
- **Conclusion**

Motivation



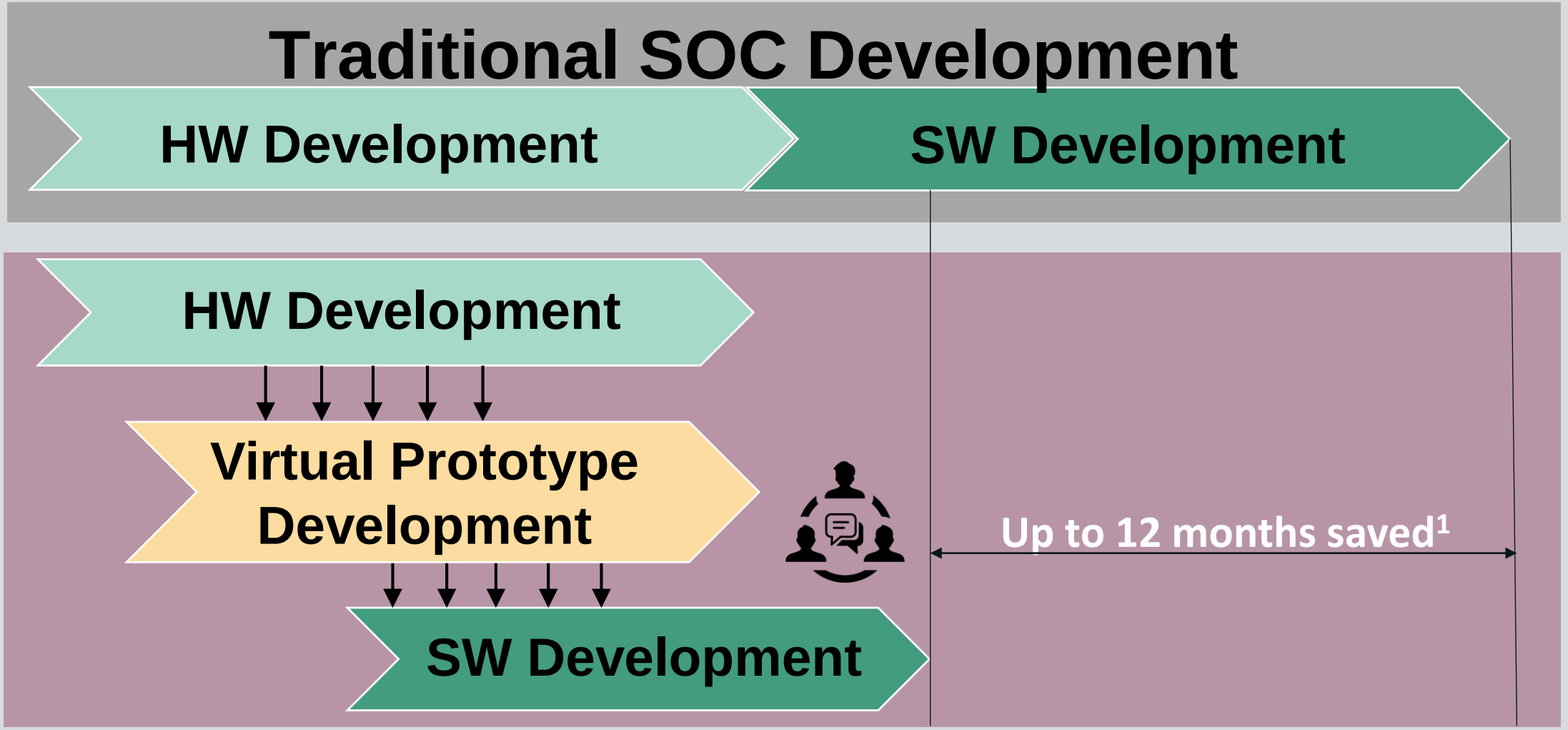
- **Modern SOC desings are everywhere.**
  - ➢IOT Devices, combining SW, HW, microcontrollers, microprocessors, and IPs.
  - ➢Seamless integration for mission critical applications.
- **Security of modern SOC desings are becoming increasingly important.**
  - ➢One bug ≈ Disastrous consequences.
  - ➢Real-time requirements, susceptible to attacks!
- **An Example: DoS Attacks on Roadside Units (RSU):**
  - ➢Disrupting communication between vehicles and/or RSUs.
  - ➢Shut down the network.
  - ➢NO road status information in time!

Security Validation Technique: Information Flow Tracking

- **IFT helps mitigate security vulnerabilities.**
  - Information Flow Policies
  - Security Properties:
    - Confidentiality
    - Integrity
    - Availabilitiy
- **Effectiveness of IFT?**
  - Accurate security property definition!
  - Ensure vulnerability detection for the targeted threat model!
- **Benefits:**
  - Assist verification engineers:
    - Better intuition to assess weaknesses,
    - Understand vulnerabilities,
    - Derive appropriate security properties.
- **How can we measure?**
  - **Security Coverage Metrics**
    - Qualitatively
    - Quantitatively

Virtual Prototypes (VPs)

★Golden reference ★Abstract SW models of HW ★SystemC/ AMS

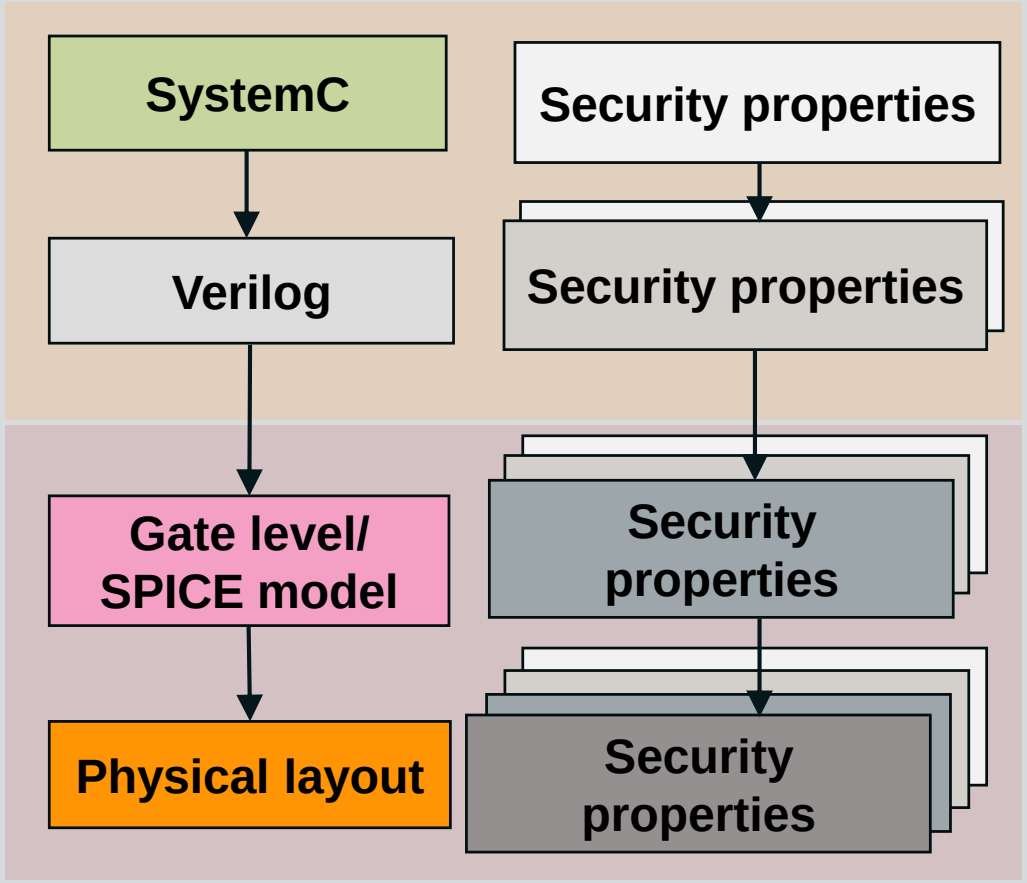


1. https://armkeil.blob.core.windows.net/developer/Files/pdf/white-paper/virtual-prototyping-soc-design.pdf

Completeness Driven Development (CDD)

Ensure competeness: Security aspect →

- verify the complete behavior of the design at each level of abstraction.
- Formerly used for functional correctness[2].



Security aware CDD concept.

[2] R. Drechsler, M. Diepenbeck, D. Große, U. Kühne, H. M. Le, J. Seiter, M. Soeken, and R. Wille, "Completeness-driven development," in International Conference on Graph Transformations, 2012, pp. 38–50.

Threat Model

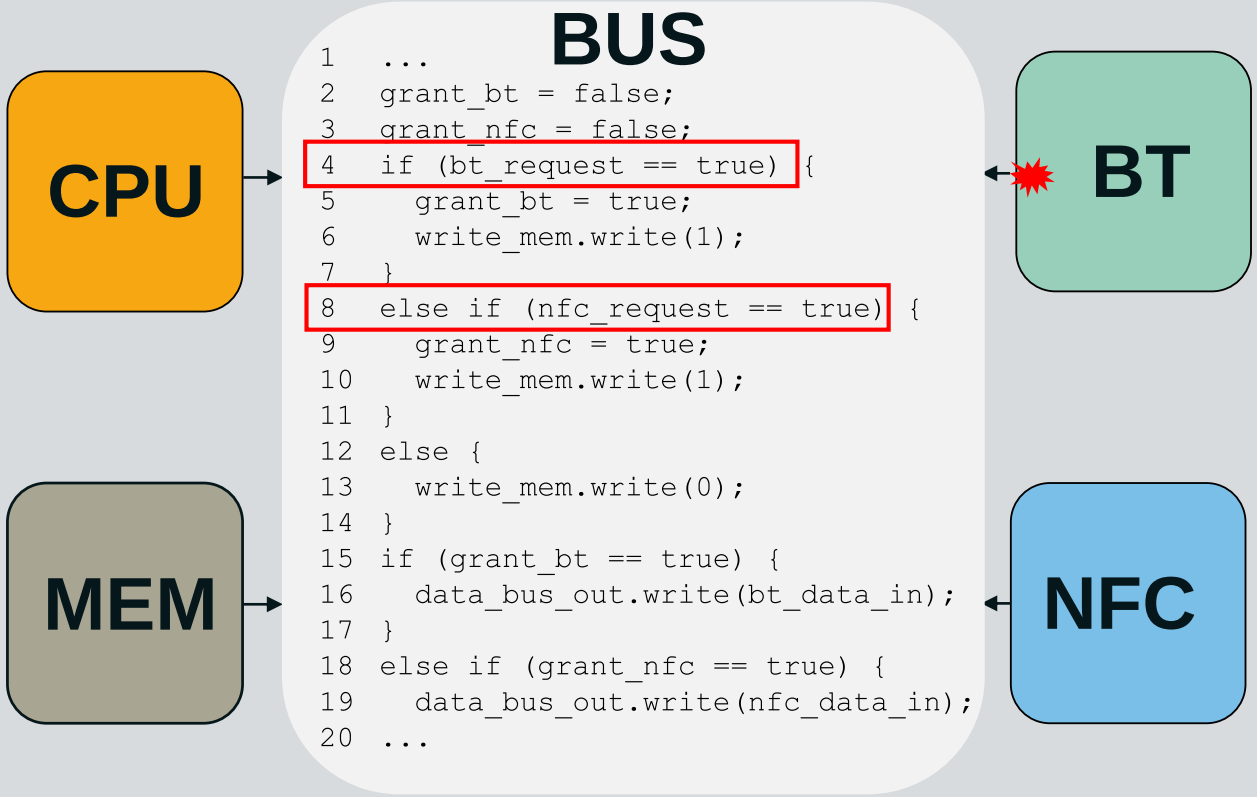
Availability:

- Availability: The timely access of shared resources.
- Availability issues: IPs overusing shared resources, making them inaccessible to other IPs.

➢ Improperly configured access control system!

➢ Detecting such relation is challenging without automated IFT tools!

Motivating Example

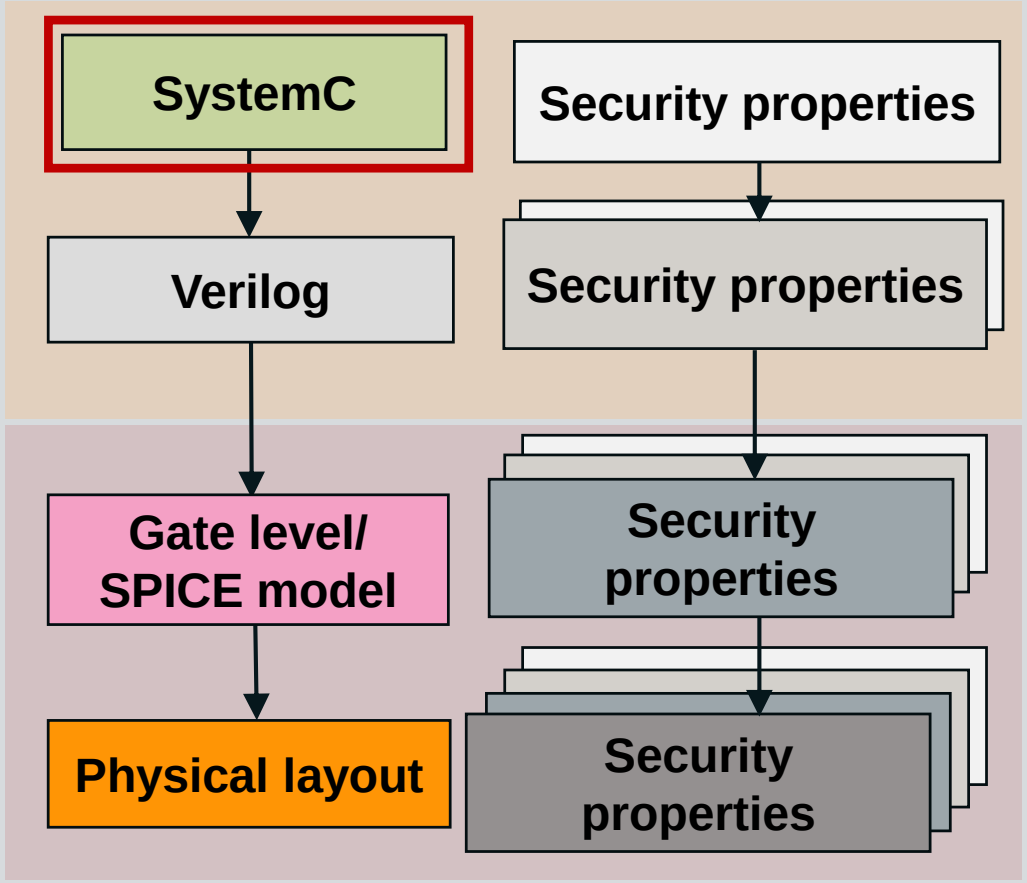


A simplified SystemC model of Keyless Entry System.

Security Coverage Metrics

- Sytem level Information Flow.
- Availability Threat Model.

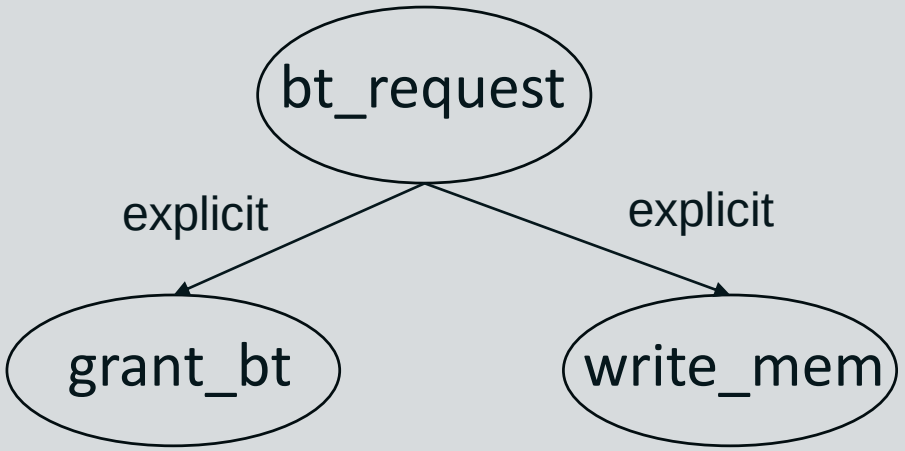
1. **Direct Signal Connectivity**
2. **Indirect Signal Connectivity**
3. **Partial Path Activation**
4. **Full Path Activation**
5. **Information Flow Rate**



Security aware CDD concept.

Direct Signal Connectivity (DSC)

- DSC metric determines whether a signal A and a signal B are directly, i.e. explicitly connected.



BUS

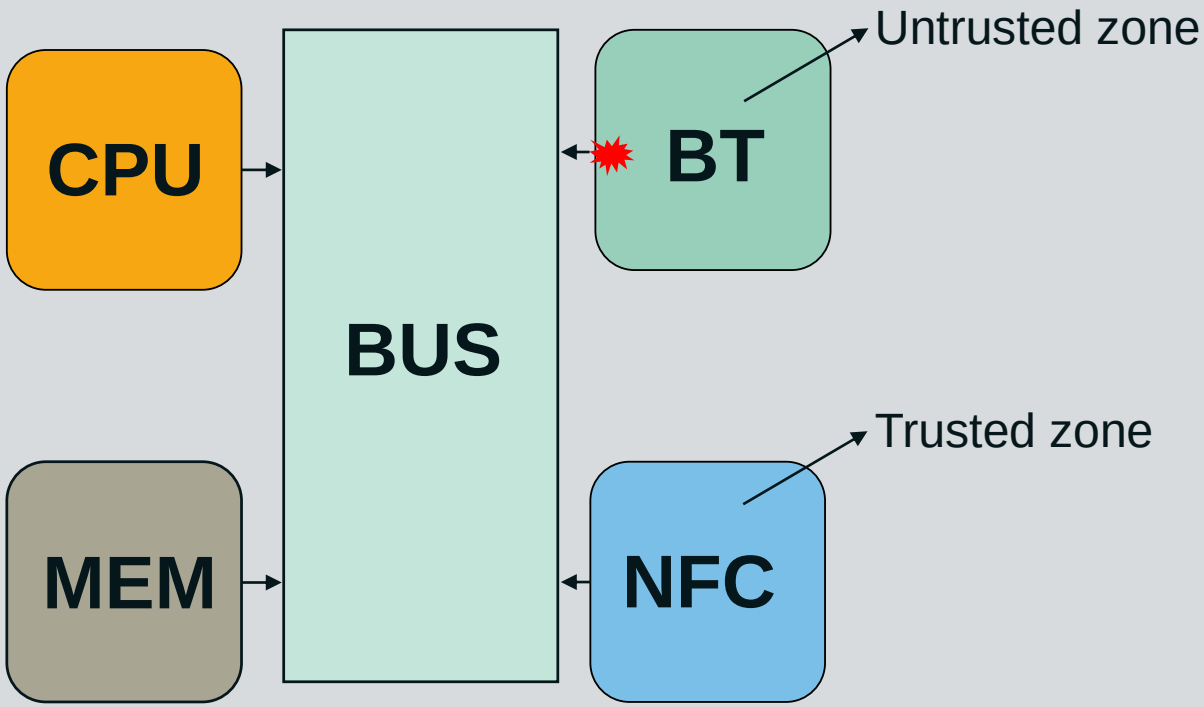
```
1   ...
2   grant_bt = false;
3   grant_nfc = false;
4   if (bt_request == true) {
5     grant_bt = true;
6     write_mem.write(1);
7   }
8   else if (nfc_request == true) {
9     grant_nfc = true;
10    write_mem.write(1);
11  }
12  else {
13    write_mem.write(0);
14  }
15  if (grant_bt == true) {
16    data_bus_out.write(bt_data_in);
17  }
18  else if (grant_nfc == true) {
19    data_bus_out.write(nfc_data_in);
20  ...
```

Code excerpt of Bus IP.

Indirect Signal Connectivity (ISC)

- ISC metric determines whether a signal A and a signal B are indirectly connected.
- Implicit Information Flow (IIF), more subtle compared to the EIF, could result in the **unavailability** of the IPs.

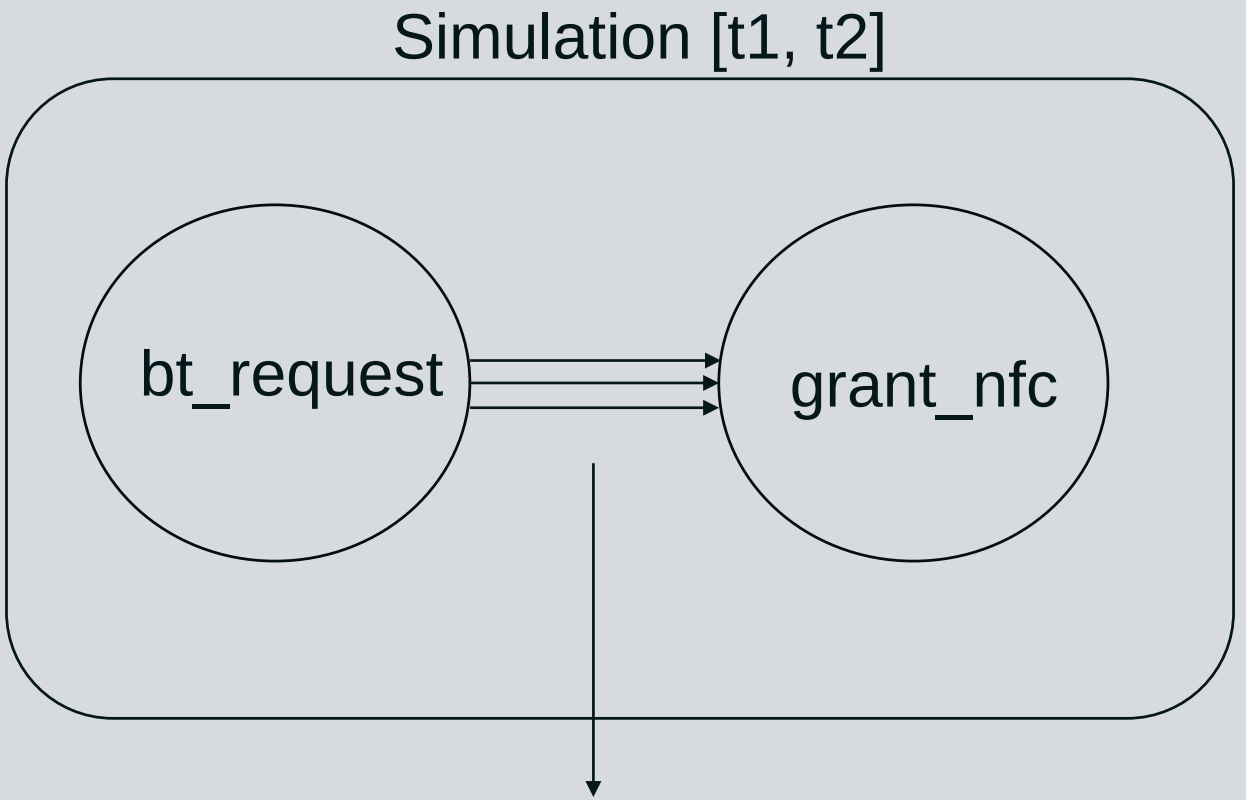
- The IIF might occur between two modules, where one belongs to the **trusted zone** and the other to the **untrusted zone**, sharing memory.



A simplified SystemC model of Keyless Entry System.

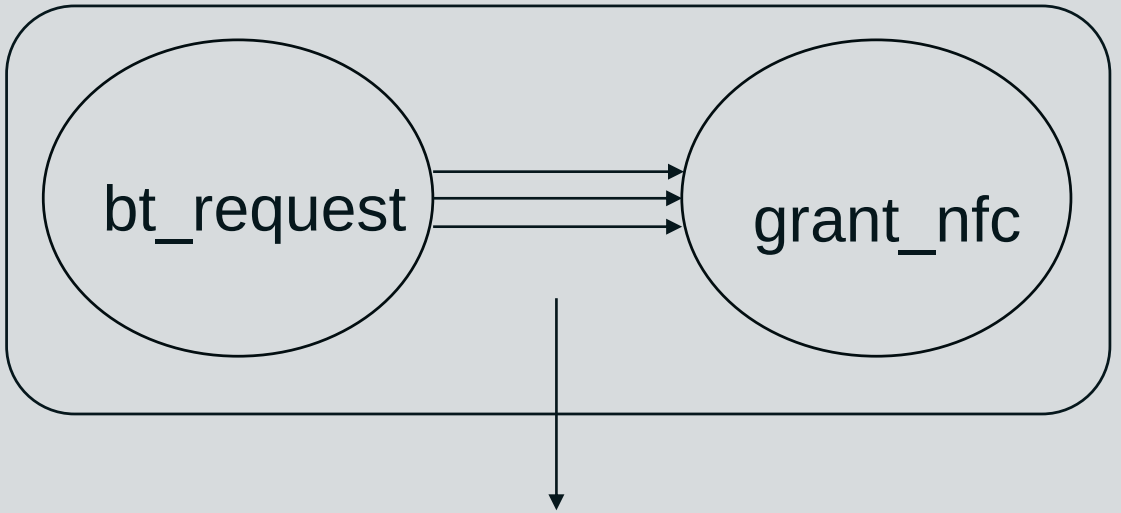
Partial Path Activation (PPA)

- A definite flow: beyond mere connectivity; activation of paths!
- Activation of a path: signal A successfully propagates to signal B in a simulation for a given time interval.



Full Path Activation (FPA)

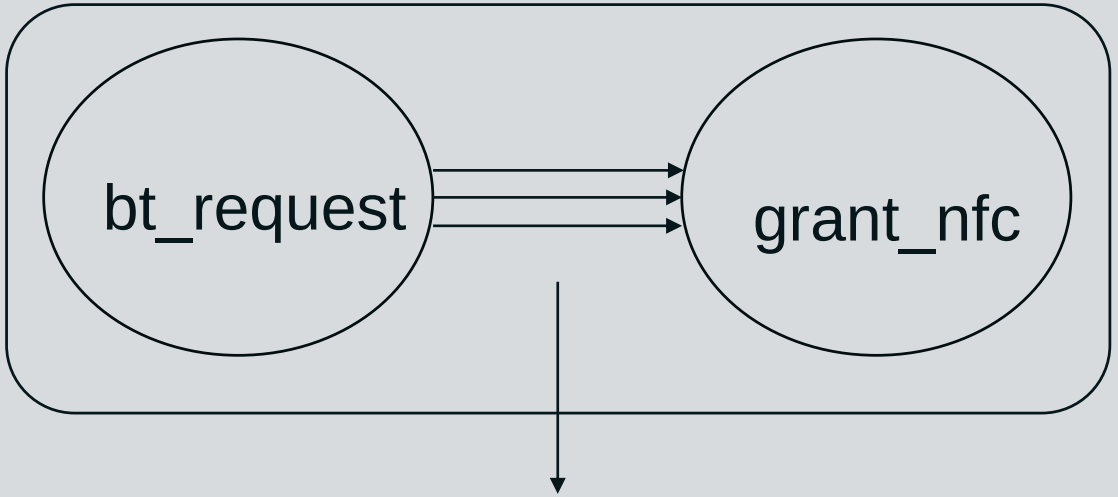
- FPA quantifies the total path activation for the whole execution time.



How many of paths are activated throughout the entire execution time?

Information Flow Rate (IFR)

- IFR quantifies the occurrence of an information flow from a signal A to a signal B within a given time frame.



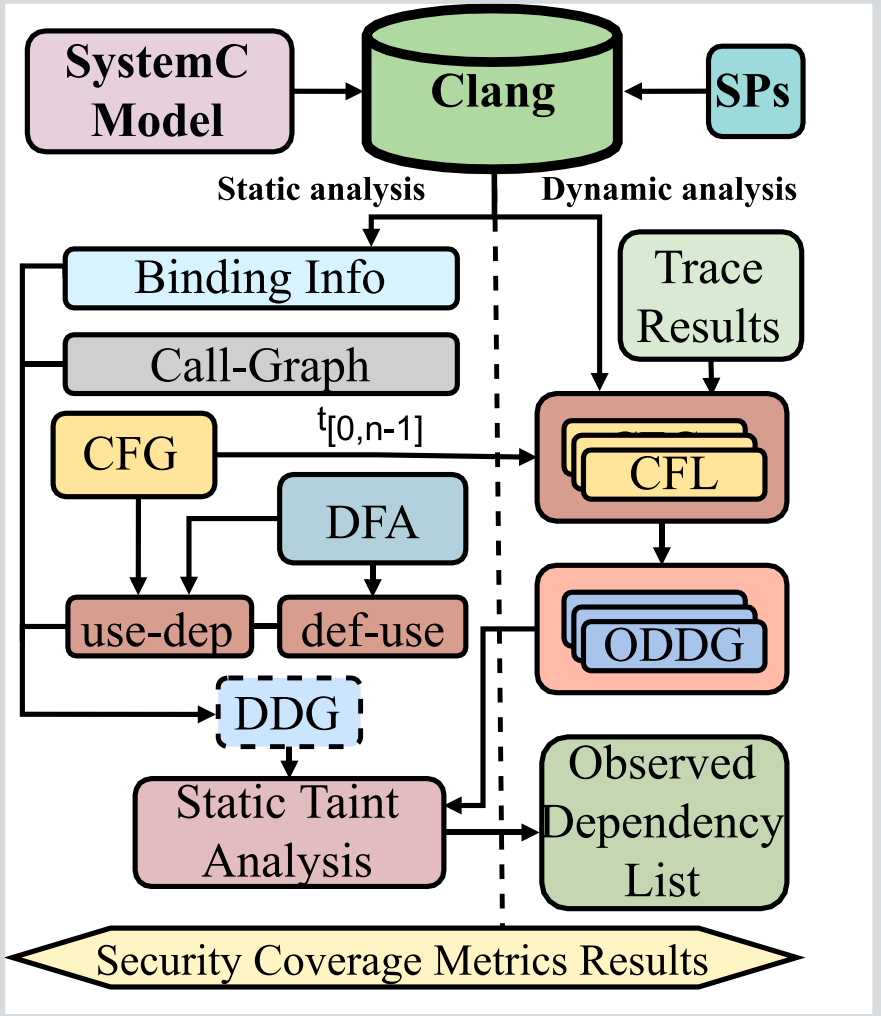
How many times does signal A reaches signal B during the time interval?

SiMiT: A static+dynamic IFT tool

- Security Properties (SPs): guarantees outputs are Always Available (AA) when needed.

SP={(SI,SO)|SI∈ {in1=HS,..},SO∈ {out1=AA,..}}

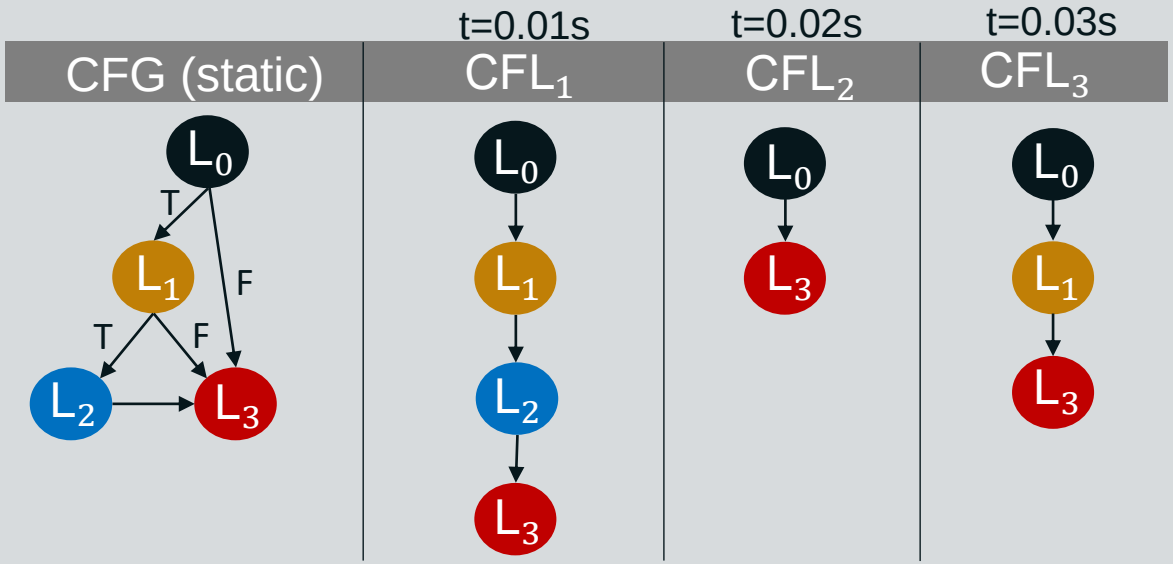
- Binding Information: connects modules.
- Call-Graph: coordinates analysis.
- Control Flow Graph(CFG): helps understanding the relationship between various statements.

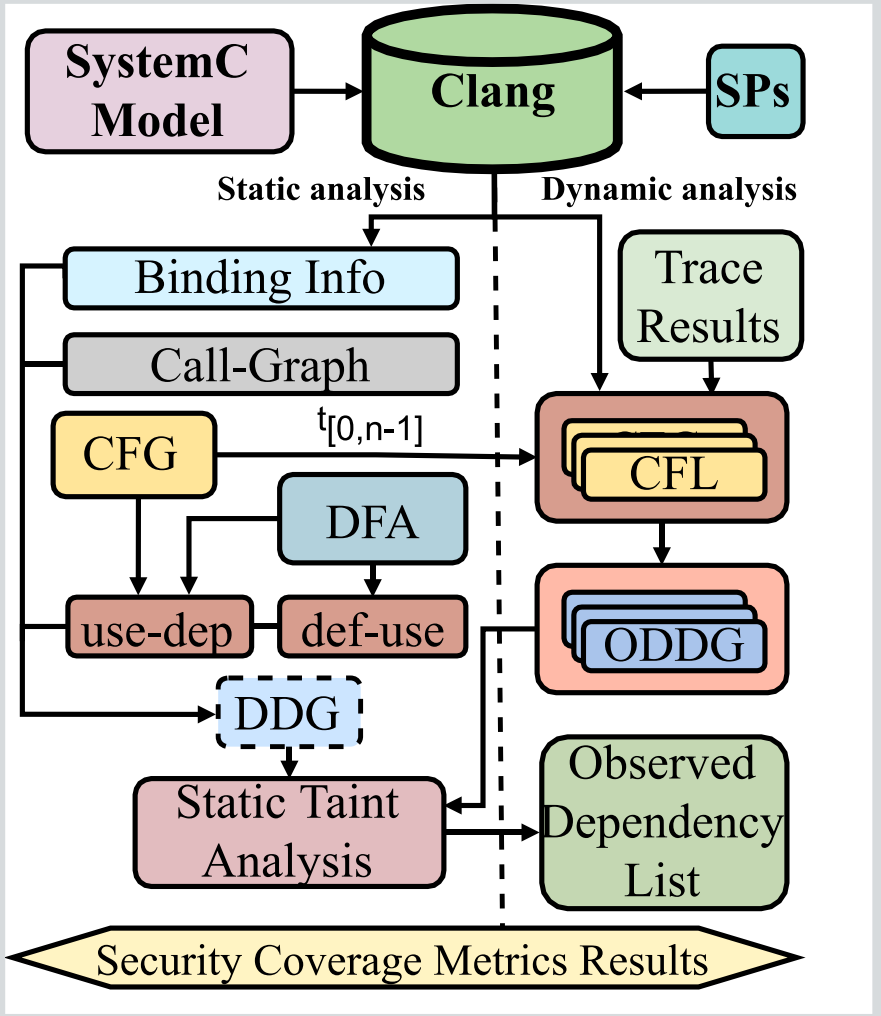


Overview of SiMiT Tool.

SiMiT: A static+dynamic IFT tool

- Trace: A log file records “visited nodes” incl. file names and line numbers (L) for each time step.
- Control Flow List (CFL): added traversed nodes from the CFG’s node.

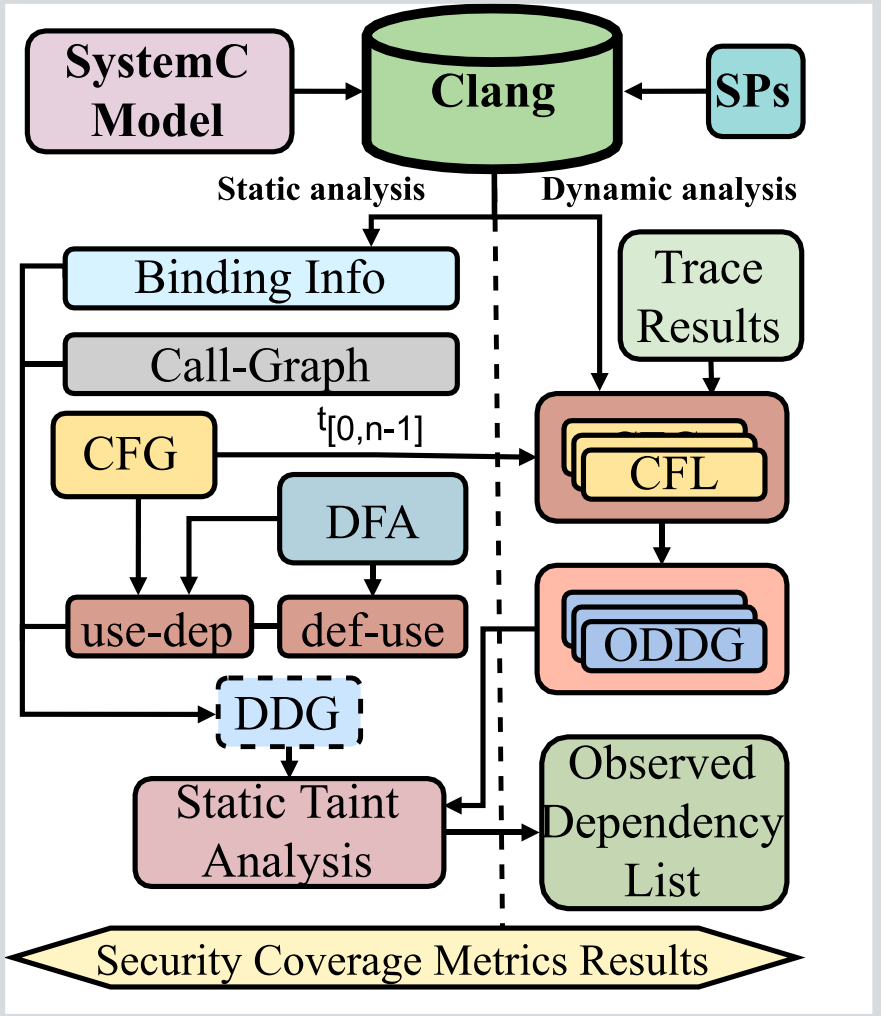




Overview of SiMiT Tool.

SiMiT: A static+dynamic IFT tool

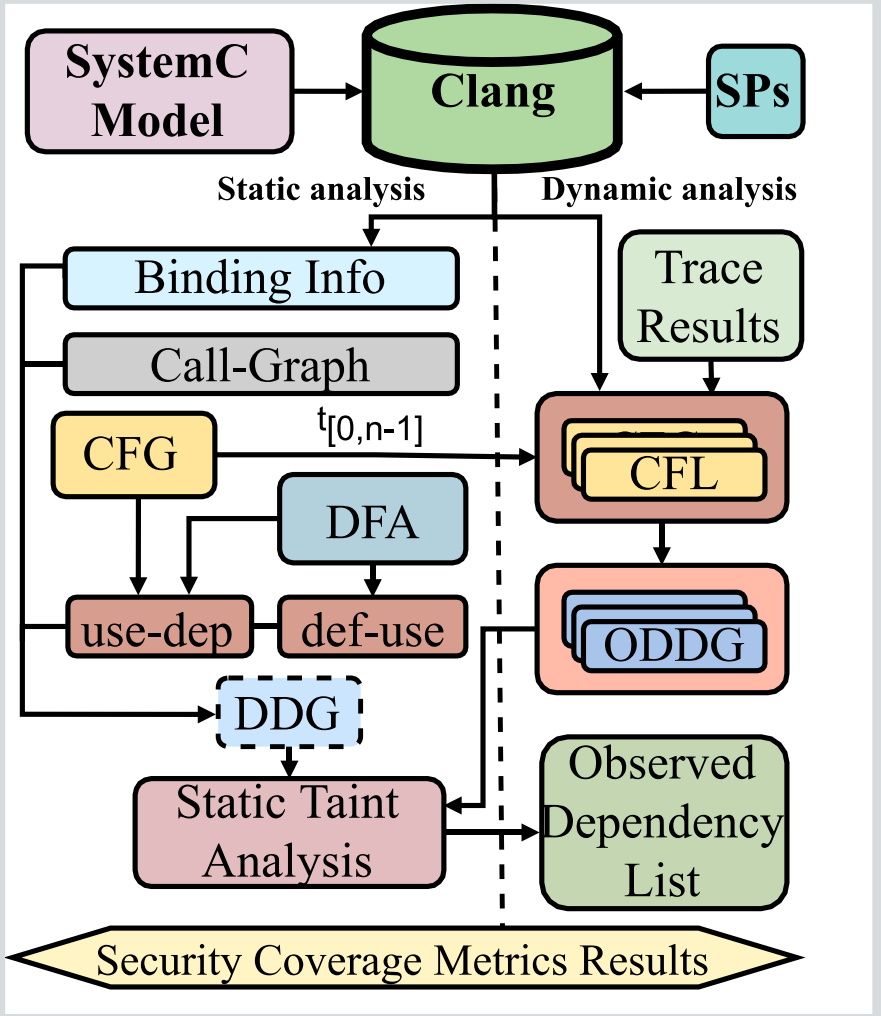
- Data Flow Analysis (DFA): constructs def-use, use_dep pairs.
- def-use: “for each defined variable, which uses may potentially utilize its values.”
- use-dep: dependence for variables in the conditional statements of CFG blocks, where definitions in the possible succesors to the conditional statements are stored.
- Data Dependency Graph (DDG): maps the relationships among variables, incl. signals, ports, variables.



Overview of SiMiT Tool.

SiMiT: A static+dynamic IFT tool

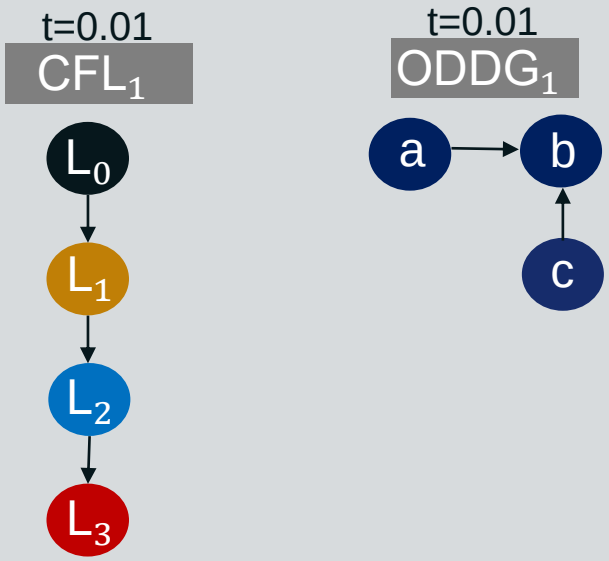
- Data Flow Analysis (DFA): constructs def-use, use_dep pairs.
- def-use: “for each defined variable, which uses may potentially utilize its values.”
- use-dep: “for each variable in conditional statements, which variables are assigned in the succesors to the condition.”
- Data Dependency Graph (DDG): maps the relationships among variables, incl. signals, ports, variables.

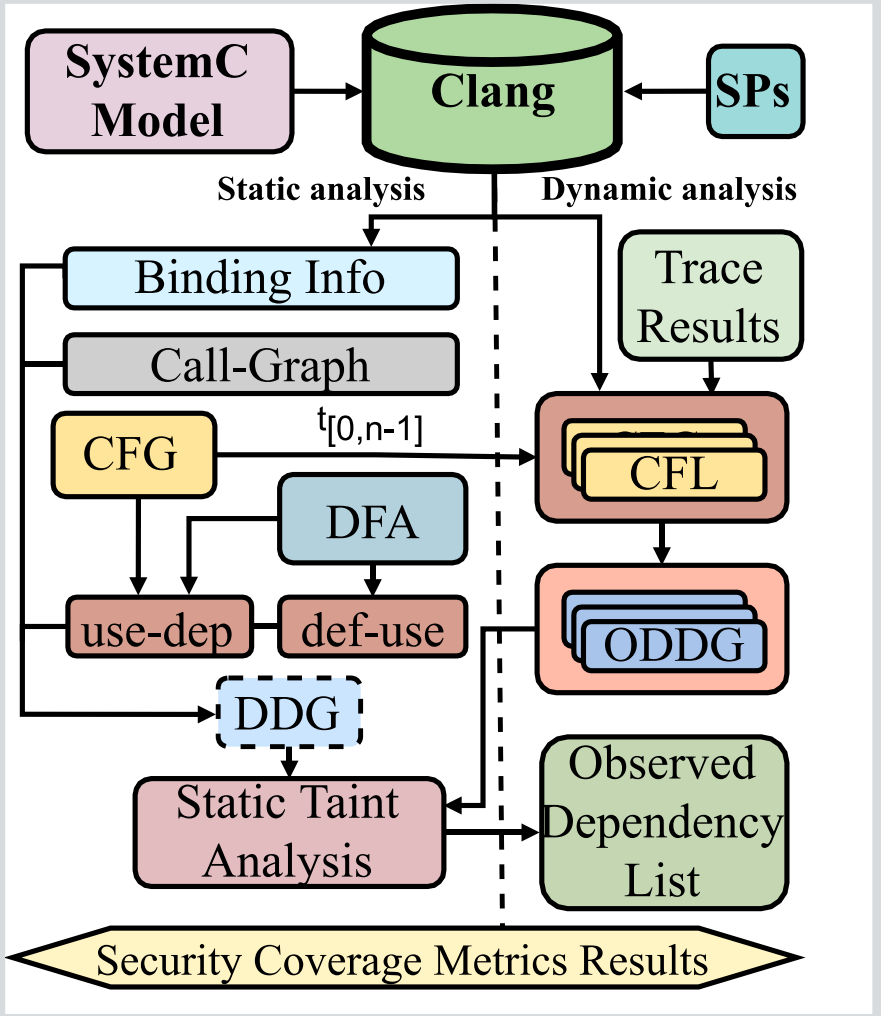


Overview of SiMiT Tool.

SiMiT: A static+dynamic IFT tool

- Observed Data Dependency Graph: formed using the relevant CFLs for each time steps, rather than the CFG.

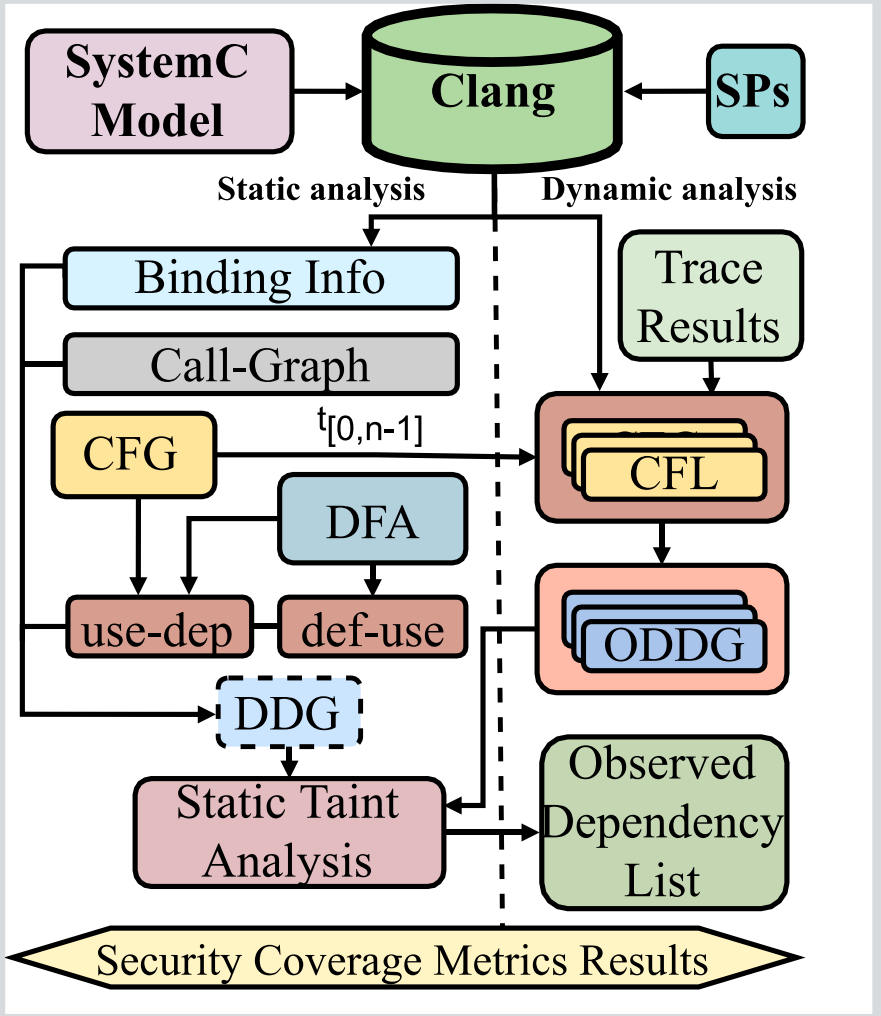




Overview of SiMiT Tool.

SiMiT: A static+dynamic IFT tool

- Static Taint Analysis: generates the **Observed Dependency List** (ODL). It begins with a tainted source and incrementally includes variables by executing a Depth First Search (DFS) that utilizes dependence data from each ODDG.



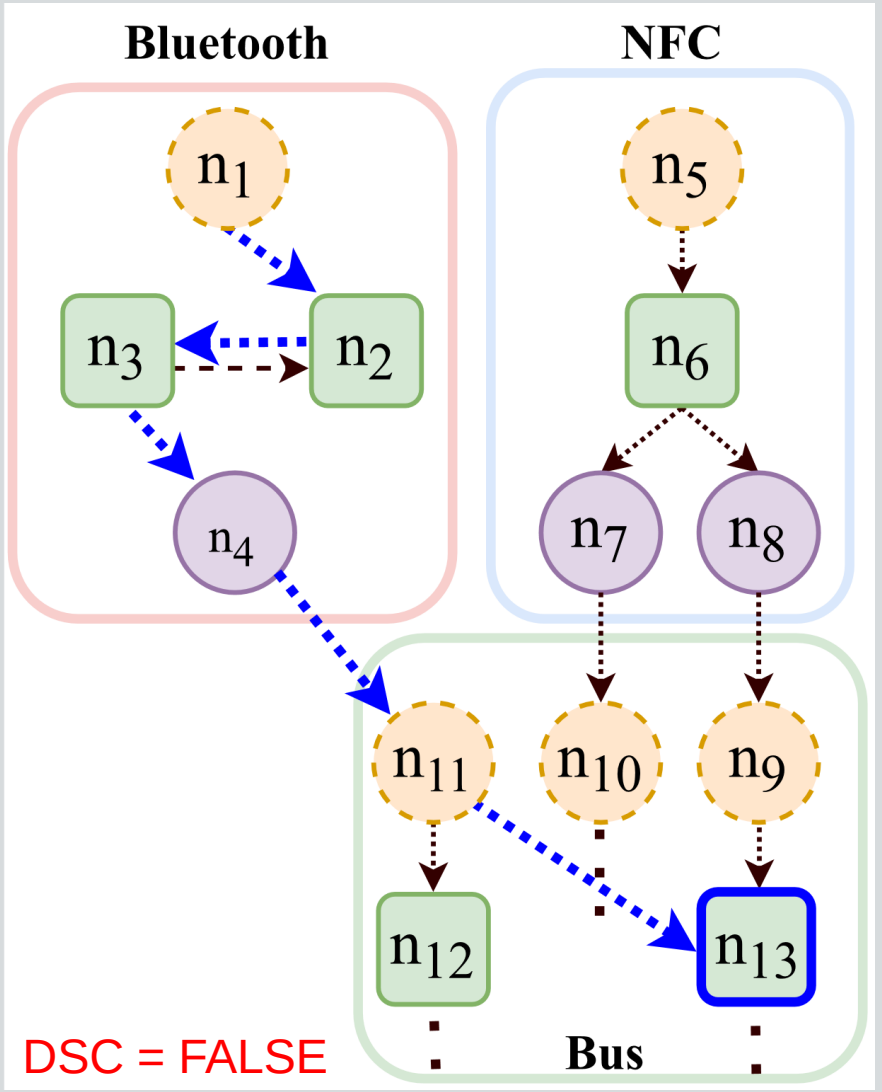
Overview of SiMiT Tool.

Implementation of Security Coverage Metrics

$$SP = (\{\text{bt\_enable\_in} = \text{HS}\}, \{\text{grant\_nfc} = \text{AA}\})$$

1. Direct Signal Connectivity:

- **Identify EIF**; find variables influenced by Secure Inputs (SI).
- Secure List of SI (SLSI): Employ forward tracing from SI node to a Secure Output (SO) node.
- Identify sensitive control variables in SLSI.
- For all child nodes of the conditional nodes with sensitive control variables, if any left-hand side variable is in SO, there is EIF.

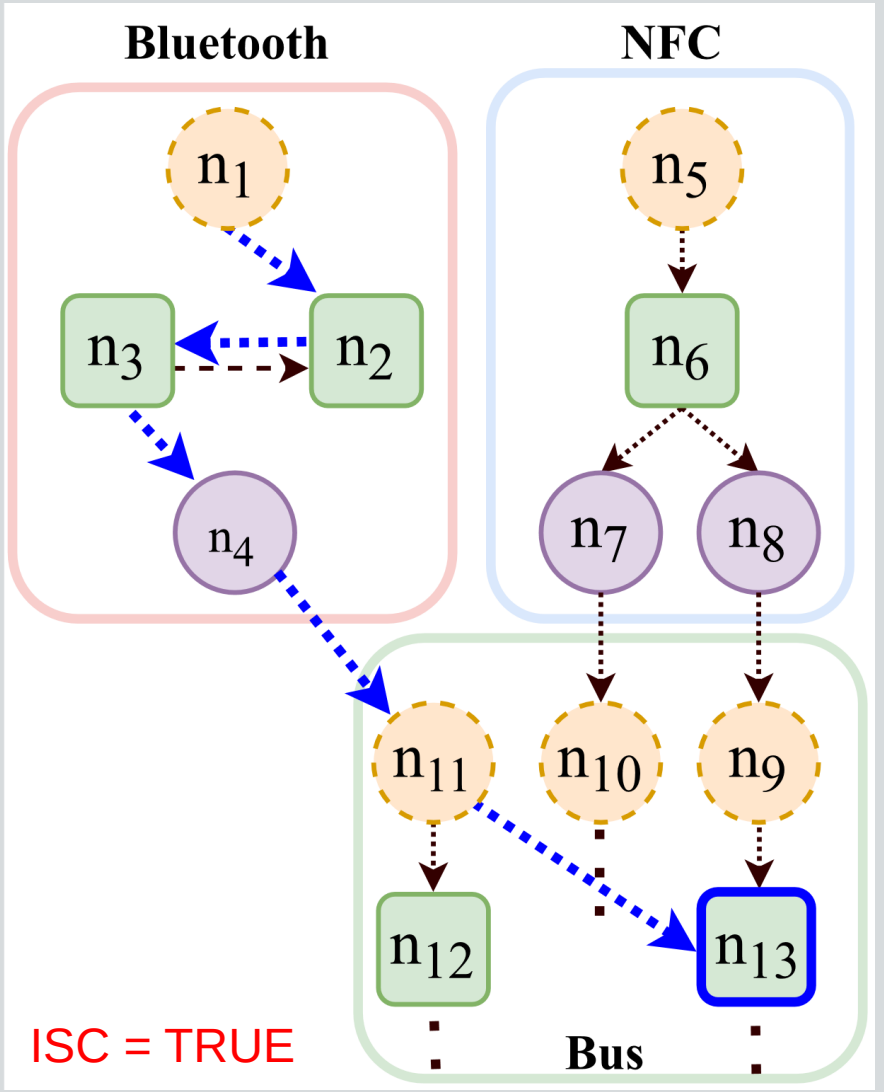


Implementation of Security Coverage Metrics

$$SP = (\{\text{bt\_enable\_in} = \text{HS}\}, \{\text{grant\_nfc} = \text{AA}\})$$

2. Indirect Signal Connectivity:

- **Identify IIF:**
  - Secure List of SO: Employ backward tracing to extract assignment statements explicitly linked to AA outputs.
  - For all child nodes of the conditional nodes with sensitive control variables, if any left-hand side variable is in SLSO, there is IIF.



Implementation of Security Coverage Metrics

$$SP = (\{\text{bt\_enable\_in} = \text{HS}\}, \{\text{grant\_nfc} = \text{AA}\})$$

3. Partial Path Activation:

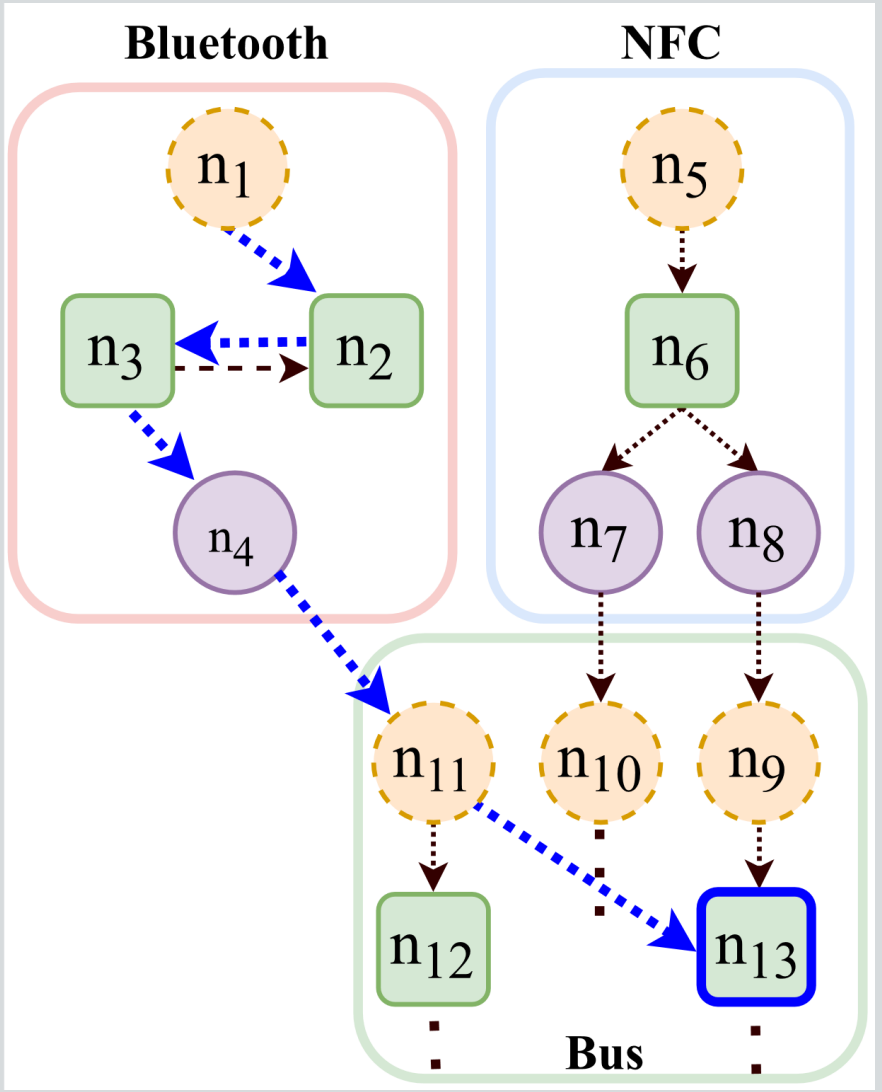
ODDG is used to find activated paths for each time step.

$$FPA(\text{HS}, \text{AA}, \text{t}_1, \text{t}_2) = \frac{n_{PPA}(HS, AA, t_1, t_2)}{n_P(HS, AA)} \begin{matrix} \rightarrow 1 \\ \rightarrow 1 \end{matrix}$$

4. Full Path Activation :

Quantify the total path activation for the whole execution time of the testbench.

$$FPA(\text{HS}, \text{AA}) = \frac{n_{FPA}(HS, AA)}{n_P(HS, AA)} \begin{matrix} \rightarrow 1 \\ \rightarrow 1 \end{matrix}$$



Implementation of Security Coverage Metrics

$$SP = (\{\text{bt\_enable\_in} = \text{HS}\}, \{\text{grant\_nfc} = \text{AA}\})$$

5. Information Flow Rate:

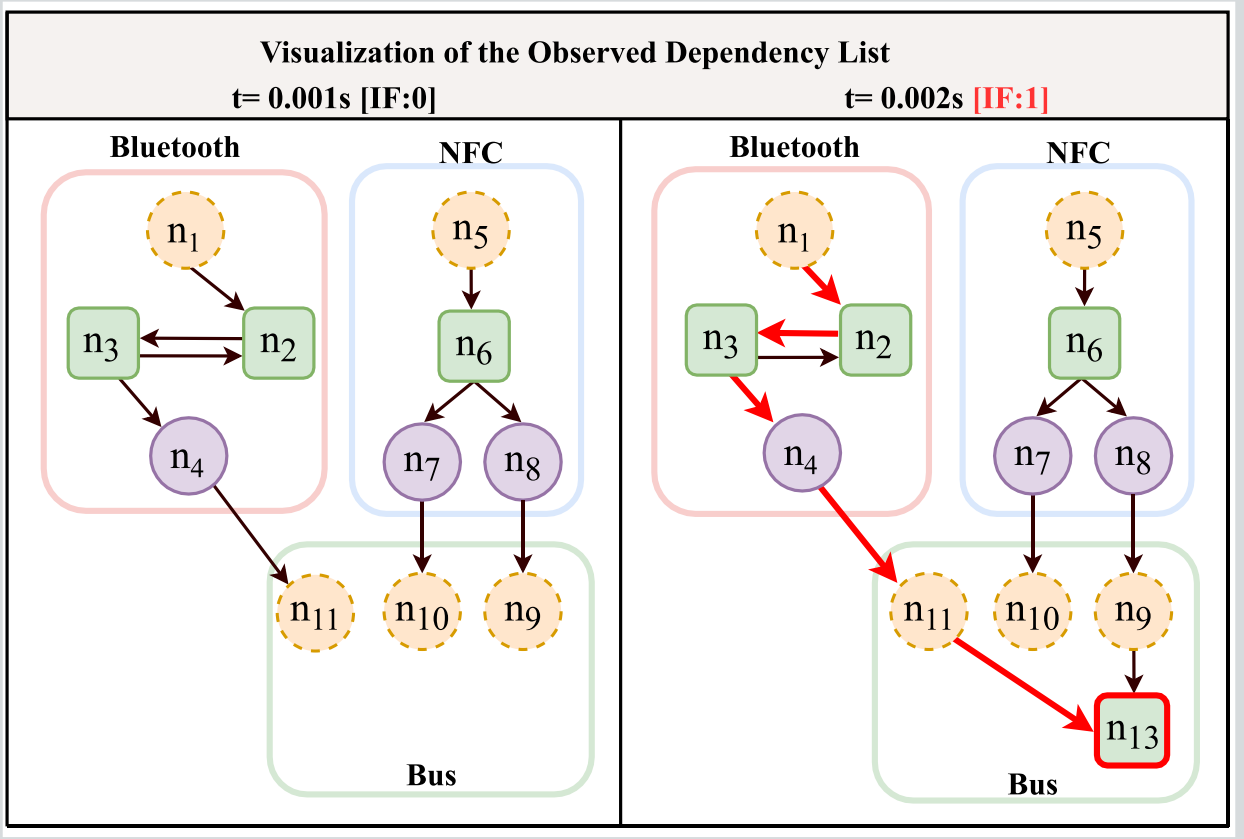
ODL is scrutinized to calculate the percentage of IFR:

$$IFR(\text{HS}, \text{AA}, s_1, s_2) = \frac{NF(HS, AA, s_1, s_2)}{1 + s_2 - s_1} \cdot 100$$

100

1

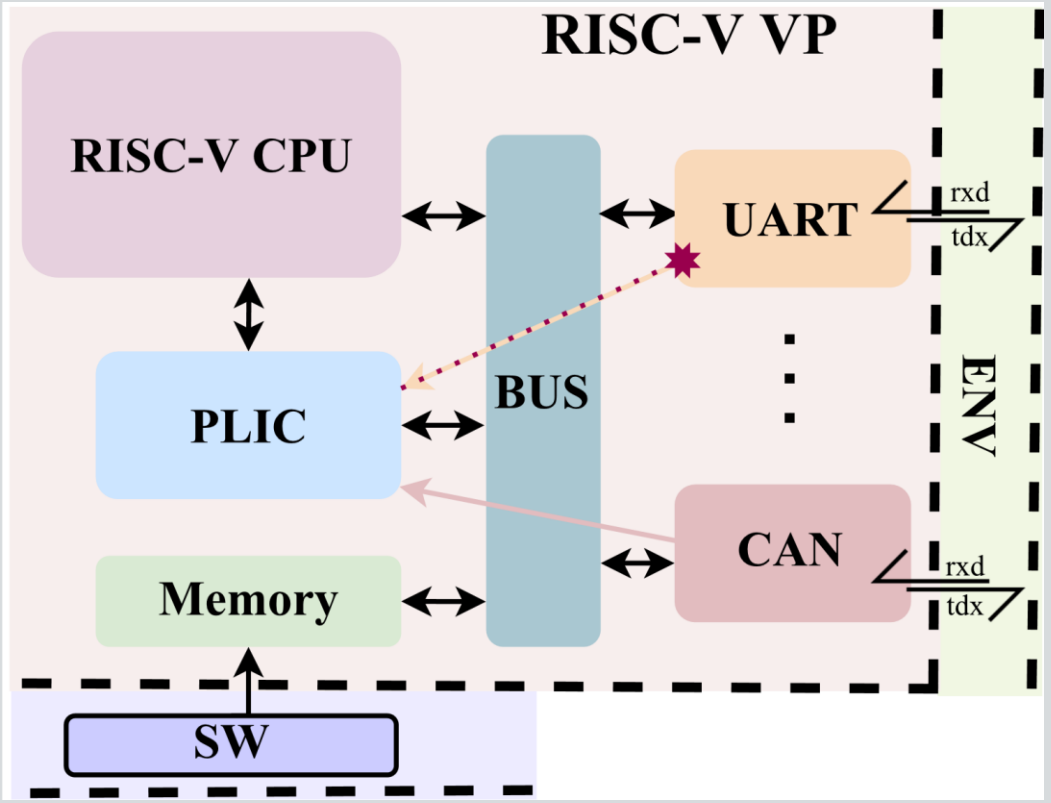
It happens 2 times in 10 ms : 2%



Experimental Results

Car Engine Immobilizer with open source available RISC-V VP:

- SystemC with TLM 2.0 Modeling.
- Misprioritizing the UART over the CAN ~
  - CPU delays, hindering communication with peripherals using critical tasks.
  - Usually does not impact normal use due to limited access to the ECU's debug interface.
  - However, CAN interrupt handling must be distinct from UART tasks to avoid signal interference.



The RISC-V VP model of a Car Engine Immobilizer

Experimental Results

$$SP = (\{\text{plic\_uart} = \text{HS}\}, \{\text{SO}_\text{s} = \text{AA}\})$$

No.	AA-tagged SOs	DSC	ISC	PPA	FPA	IFR (%)
SP_1	interrupt_can	FALSE	TRUE	1	1	$3 \cdot 10^{-4}$
SP_2	hart_config	FALSE	TRUE	1	1	$7.79 \cdot 10^{-3}$
SP_3	c.m.f.m	FALSE	TRUE	1	1	$7.78 \cdot 10^{-3}$

- 15 ms trace observed 196,313 samples to assess this effect.
- SiMiT identified multiple SPs influenced by UART signals, highlighting a potential impact on CAN message availability.
- For example, SP_1 failed:
  - interrupt_can; implicitly dependent through 6 paths to plic_uart via controlling variables.
  - SiMiT observed 6 of these 6 paths from plic_uart to interrupt_can were activated.
  - interrupt_can was reached by plic_uart in 6 of the observed samples.

Conclusion

- Definition of Security Coverage Metrics for System Level Information Flow
- Static and Dynamic Information Flow Techniques of SiMiT
- Implementation of Security Coverage Metrics in SiMiT
- Demonstration using experiments on an open source RISC-V Case study

Security Coverage Metrics for Information Flow at the System Level

Ece Nur Demirhan Coskun**

Sallar Ahmadi-Pour[U]

Muhammad Hassan[U]*

Rolf Drechsler[U]*

*Cyber-Physical Systems, DFKI GmbH
[U]Institute of Computer Science, University of Bremen
*ece.coskun@dfki.de

January 25, 2024
Incheon, South Korea

Bundesministerium für Bildung und Forschung

ECXL 01IW22002

SASPIT 16KIS1852K

Scale4Edge 16ME0127

PaSVer 16ME0855