

GENEE: A Benchmark Generator for Static Analysis Tools of Energy-Constrained Cyber-Physical Systems

CPS-IoTBench 2019 (April 15, 2019)

Christian Eichler, Peter Wägemann, and Wolfgang Schröder-Preikschat

Department of Computer Science 4 (Distributed Systems and Operating Systems)



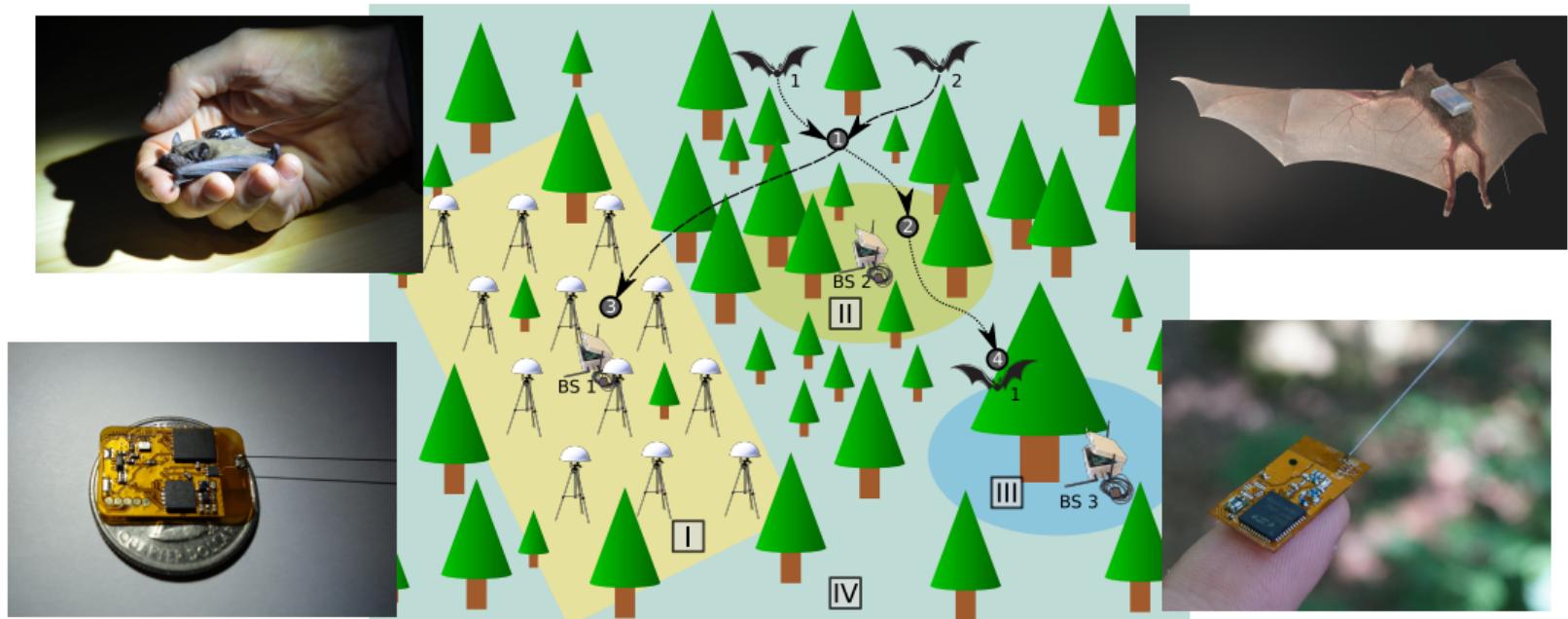
SCHR 603/9-2
SCHR 603/13-1
SCHR 603/14-2
SCHR 603/15-1



EU EFRE funds
0704/883 25

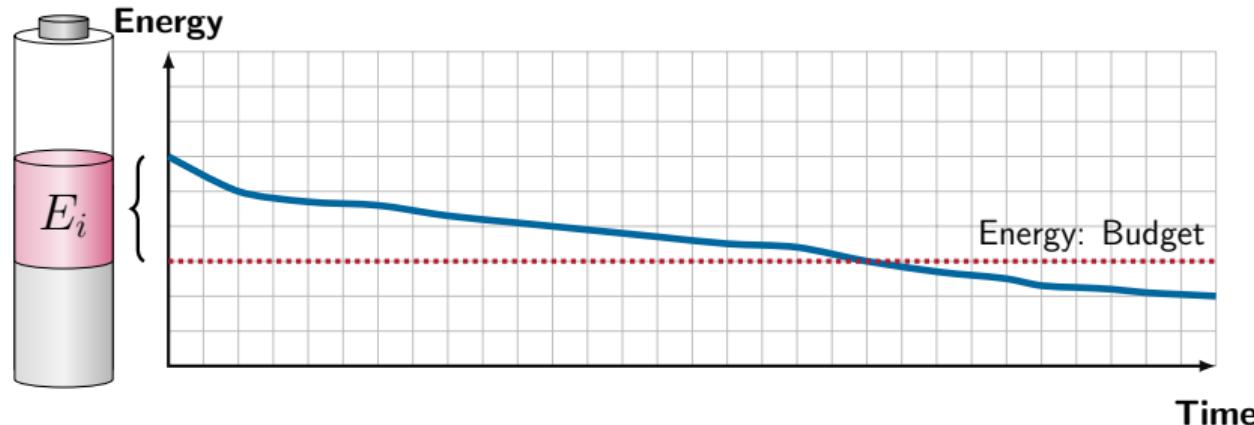


FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING



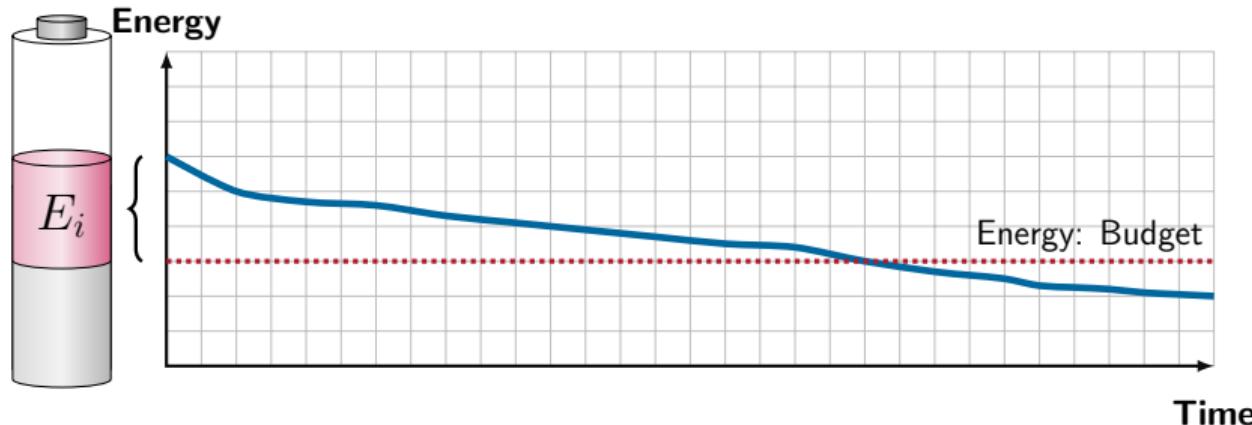
Biological challenge: understand social behavior of bats

Motivation: Operation under Energy Constraints



- Continuous collection of data (bat meetings, sensor data)
- Store data to non-volatile memory (close to depletion)
- **Guarantee completing operations: worst-case energy consumption (WCEC)**

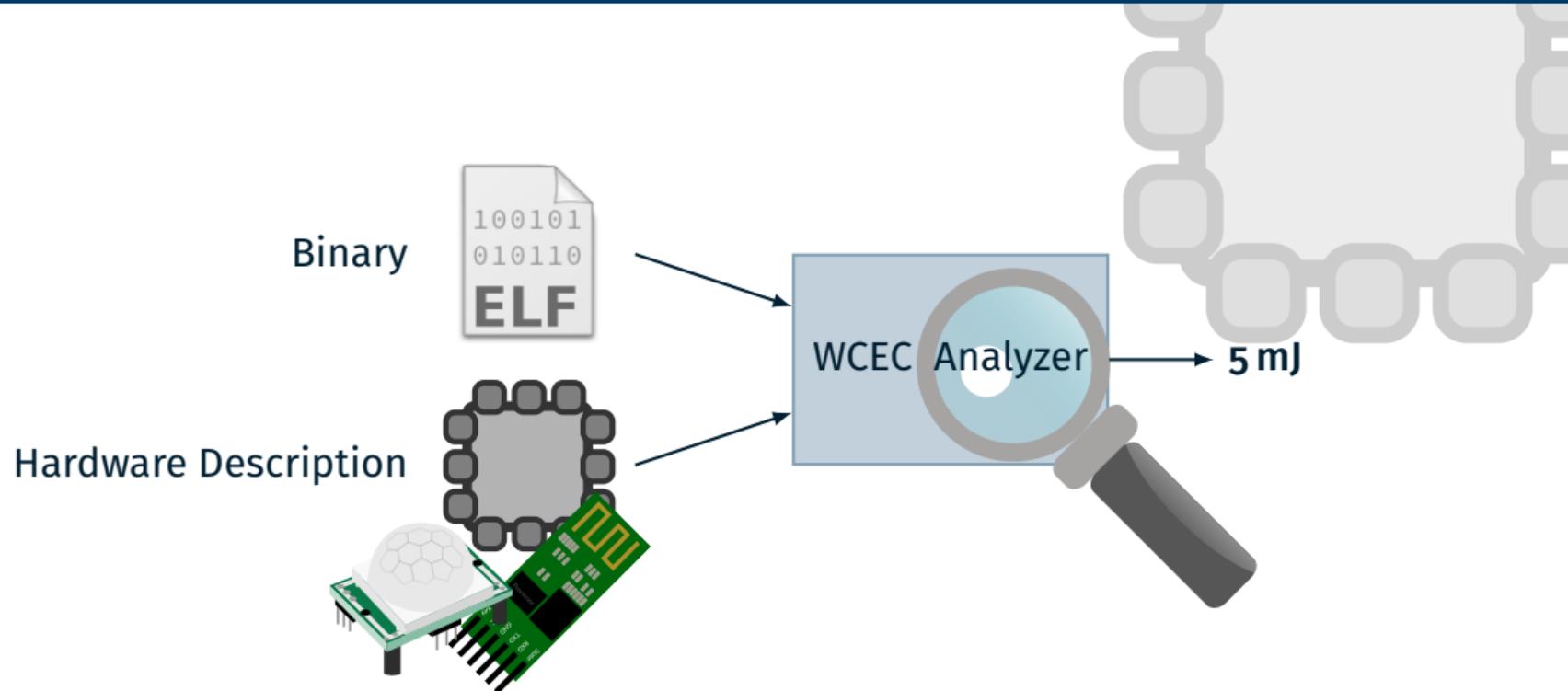
Motivation: Operation under Energy Constraints



- Continuous collection of data (bat meetings, sensor data)
- Store data to non-volatile memory (close to depletion)
- **Guarantee completing operations: worst-case energy consumption (WCEC)**

☞ WCEC required for scheduling decisions

Motivation: Worst-Case Energy-Consumption Analysis (1)



Motivation: Worst-Case Energy-Consumption Analysis (2)

```
int f(int x, int y) {  
    if(0 == x) { g(); }  
    else { h(); }
```

Energy consumption of f?

```
for(int i = 0; i < x; ++i) {  
    send_value(y);  
}  
  
if(0 != x) { m(); }  
}
```

Motivation: Worst-Case Energy-Consumption Analysis (2)

```
int f(int x, int y) {  
    if(0 == x) { g(); }  
    else { h(); }  
}
```

Energy consumption of f?

- Input dependency

```
for(int i = 0; i < x; ++i) {  
    send_value(y);  
}  
  
if(0 != x) { m(); }  
}
```

Motivation: Worst-Case Energy-Consumption Analysis (2)

```
int f(int x, int y) {  
    if(0 == x) { g(); }  
    else { h(); }
```

Energy consumption of f?

```
for(int i = 0; i < x; ++i) {  
    send_value(y);  
}
```

```
if(0 != x) { m(); }  
}
```

- Input dependency
- Loops

Motivation: Worst-Case Energy-Consumption Analysis (2)

```
int f(int x, int y) {  
    if(0 == x) { g(); }  
    else { h(); }  
}
```

```
for(int i = 0; i < x; ++i) {  
    send_value(y);  
}
```

```
if(0 != x) { m(); }  
}
```

Energy consumption of f?

- Input dependency
- Loops
- Mutual-exclusive paths

Motivation: Worst-Case Energy-Consumption Analysis (2)

```
int f(int x, int y) {  
    if(0 == x) { g(); }  
    else { h(); }  
  
    for(int i = 0; i < x; ++i) {  
        send_value(y);  
    }  
  
    if(0 != x) { m(); }  
}
```

Energy consumption of f?

- Input dependency
- Loops
- Mutual-exclusive paths
- Hardware

Different input values
→ different paths
→ **different energy consumptions**

Motivation: Worst-Case Energy-Consumption Analysis (2)

```
int f(int x, int y) {  
    if(0 == x) { g(); }  
    else { h(); }  
  
    for(int i = 0; i < x; ++i) {  
        send_value(y);  
    }  
  
    if(0 != x) { m(); }  
}
```

Energy consumption of f?

- Input dependency
- Loops
- Mutual-exclusive paths
- Hardware

Different input values
→ different paths
→ **different energy consumptions**

☞ WCEC Analyzers make pessimistic assumptions

Motivation: Baselines



Uses of systems with known **actual WCEC**:

Motivation: Baselines



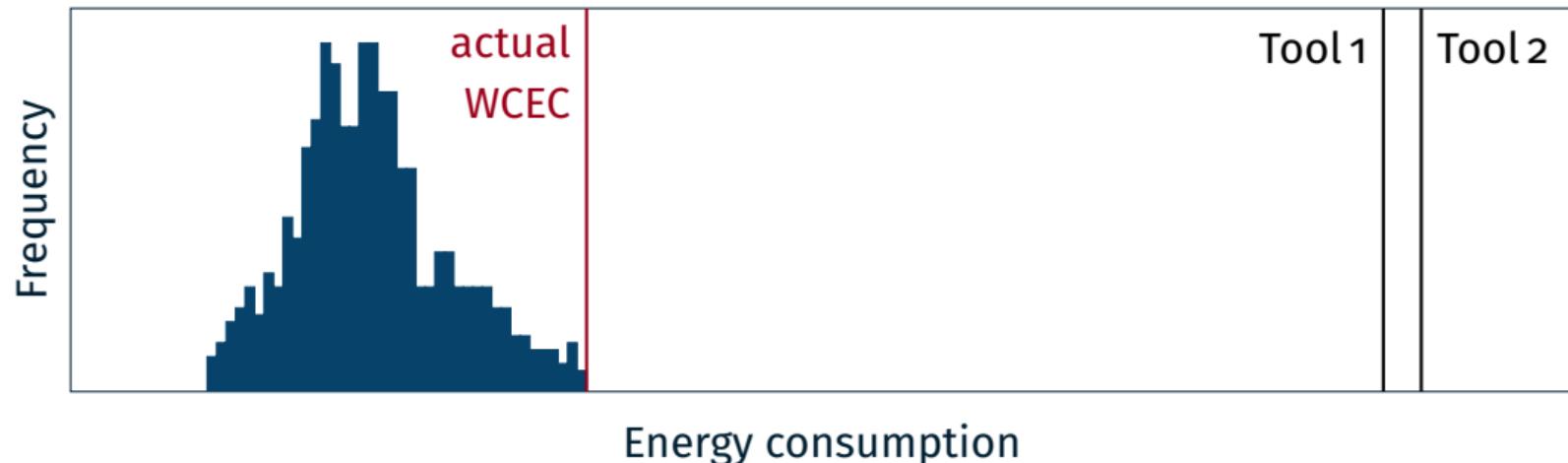
Uses of systems with known **actual** WCEC:

Motivation: Baselines



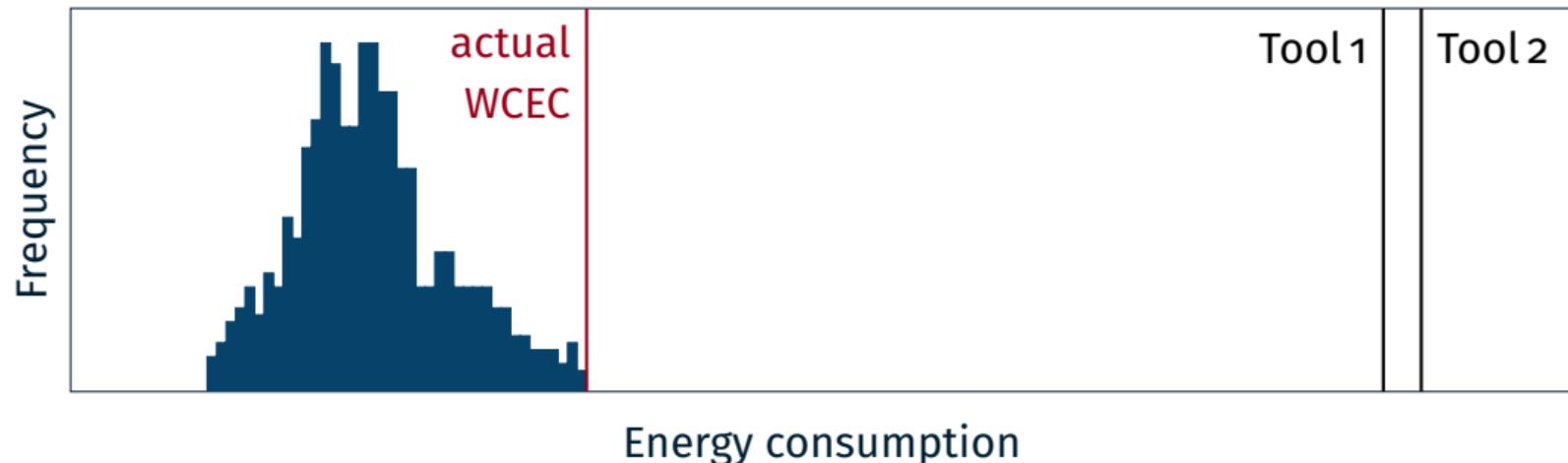
Uses of systems with known **actual** WCEC:

Motivation: Baselines



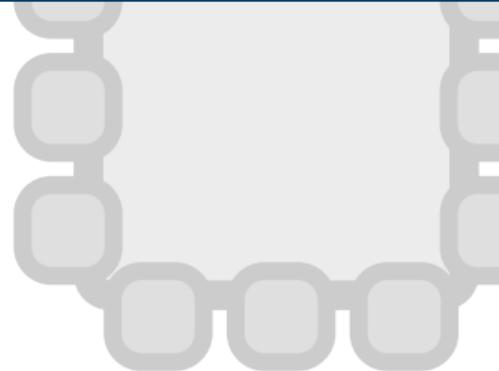
Uses of systems with known **actual WCEC**:
Estimate the accuracy of analyzers on an absolute scale

Motivation: Baselines



Uses of systems with known **actual WCEC**:
Estimate the accuracy of analyzers on an absolute scale

⇒ **Absolute baselines** for comprehensive evaluations needed



- Single-core processor with limited complexity
- Multiple devices
 - Software-controlled via device syscalls
 - State-dependent (constant) power demand
- Execution path depends on input value

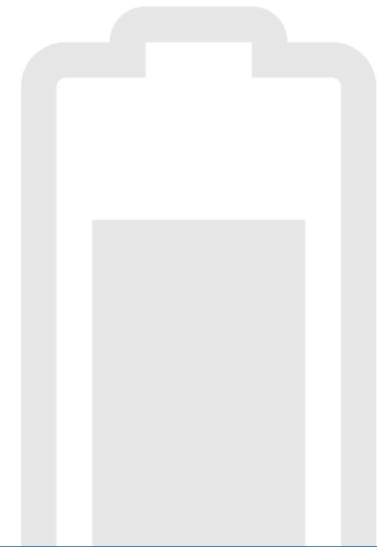
Outline

Motivation

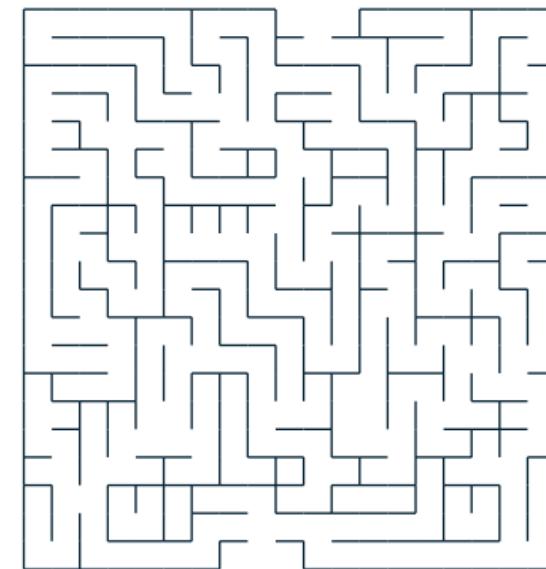
The GENEE Benchmark Generator

Evaluation

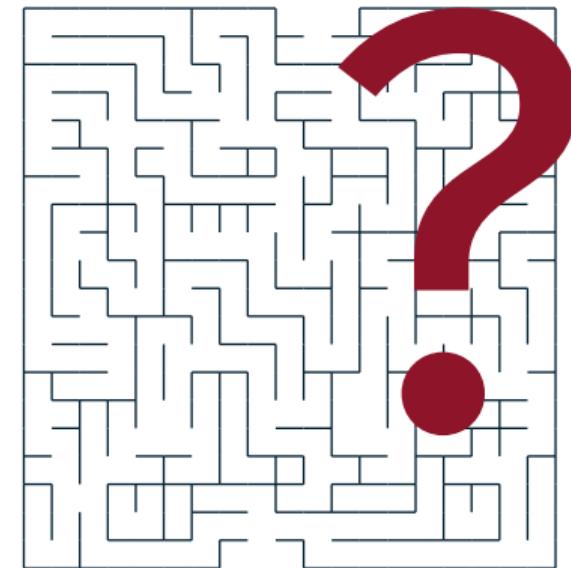
Conclusion



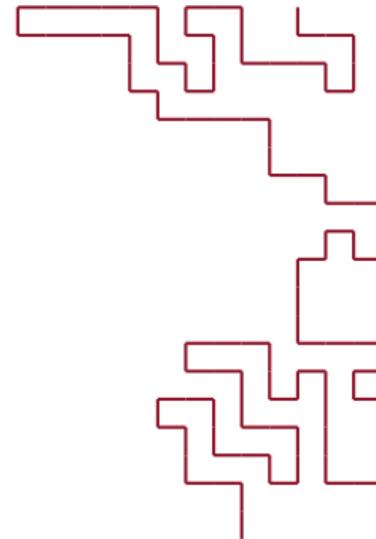
GENEE in a Nutshell



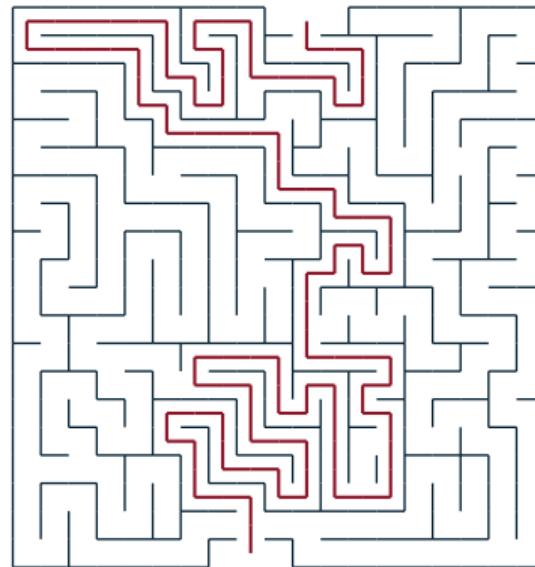
GENEE in a Nutshell



GENEE in a Nutshell



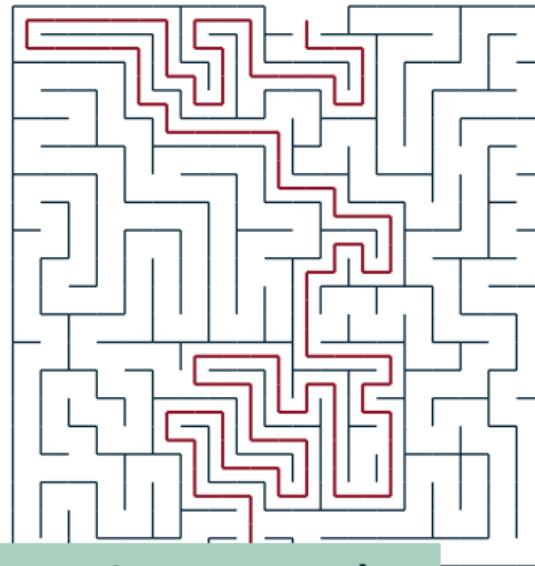
GENEE in a Nutshell



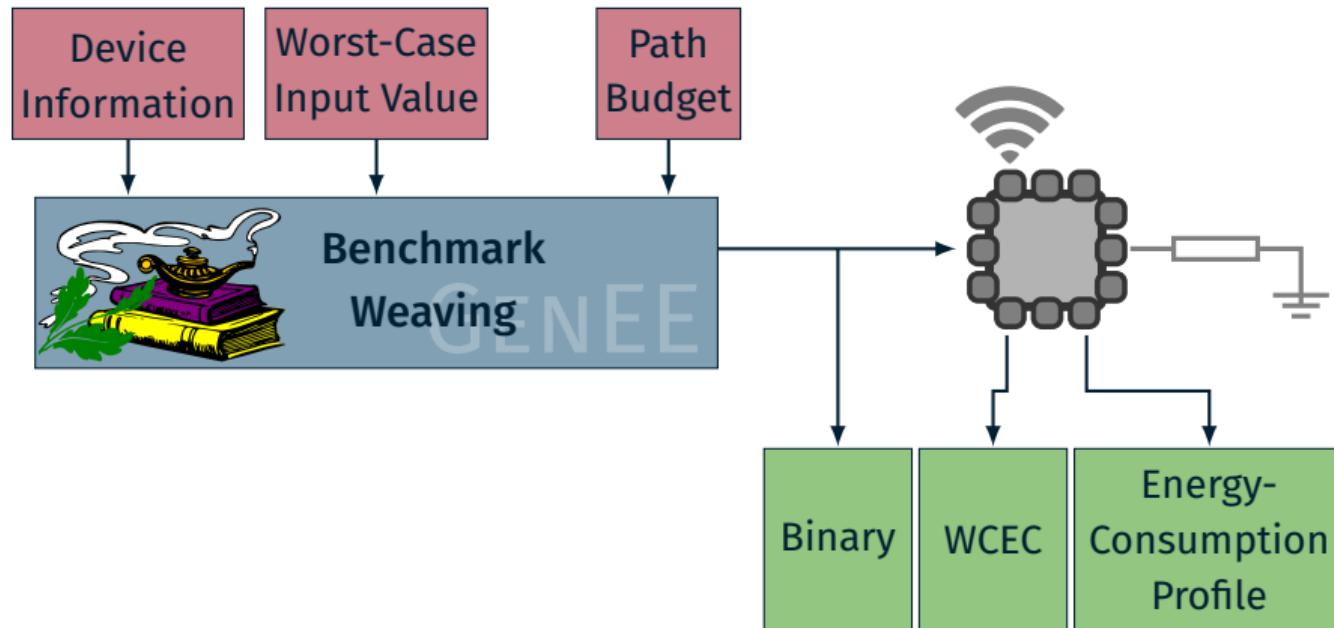
GENEE in a Nutshell



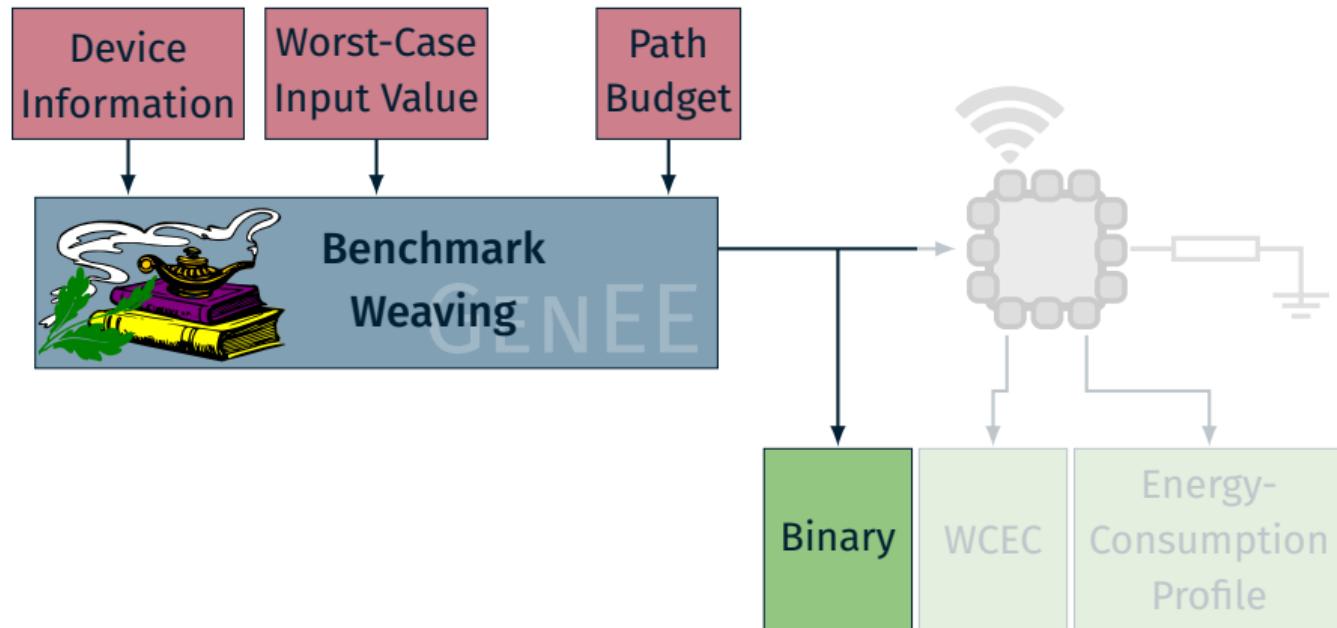
⇒ Worst-case path **known by construction**



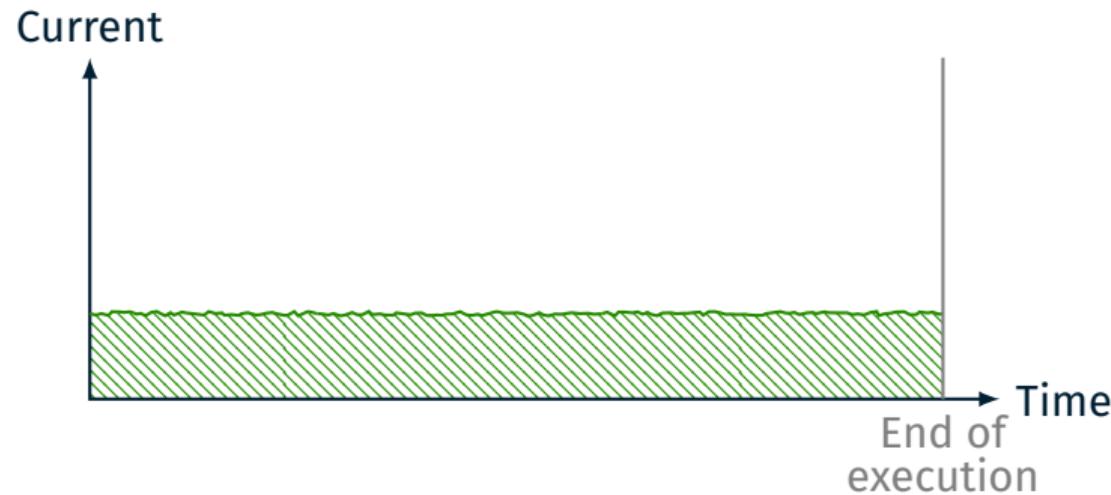
GENEE: Overview



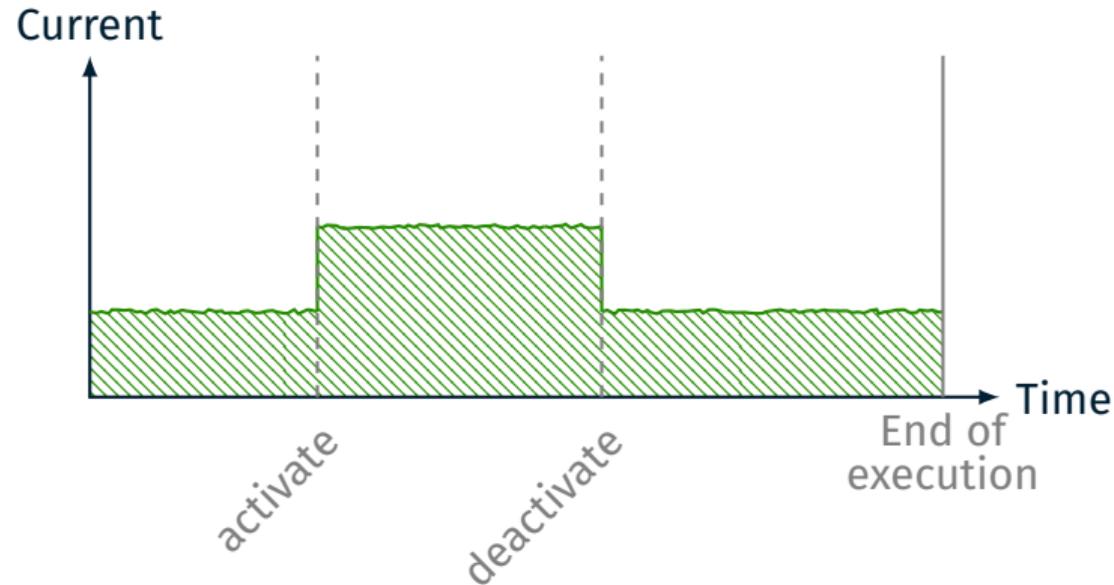
GENEE: Overview



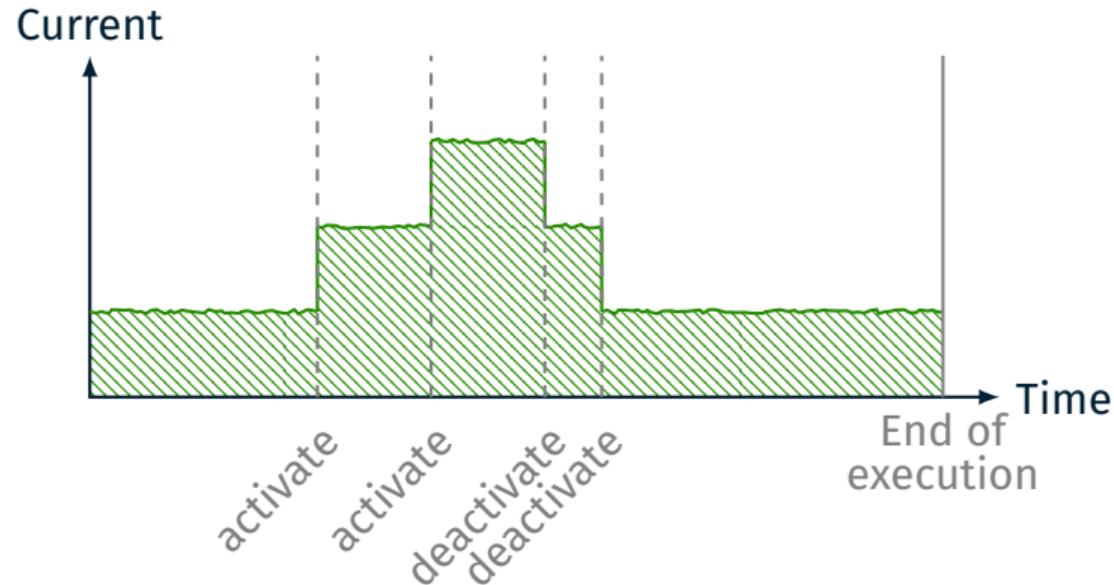
Benchmarks with Known (Worst-Case) Energy Consumption



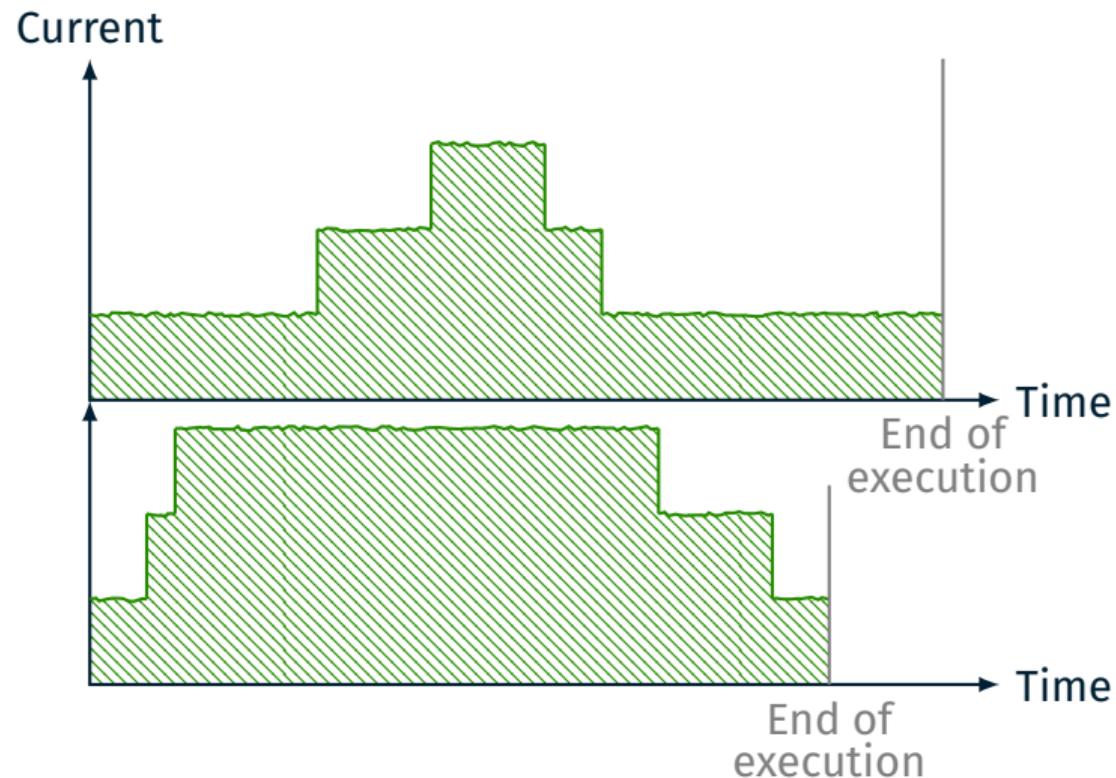
Benchmarks with Known (Worst-Case) Energy Consumption



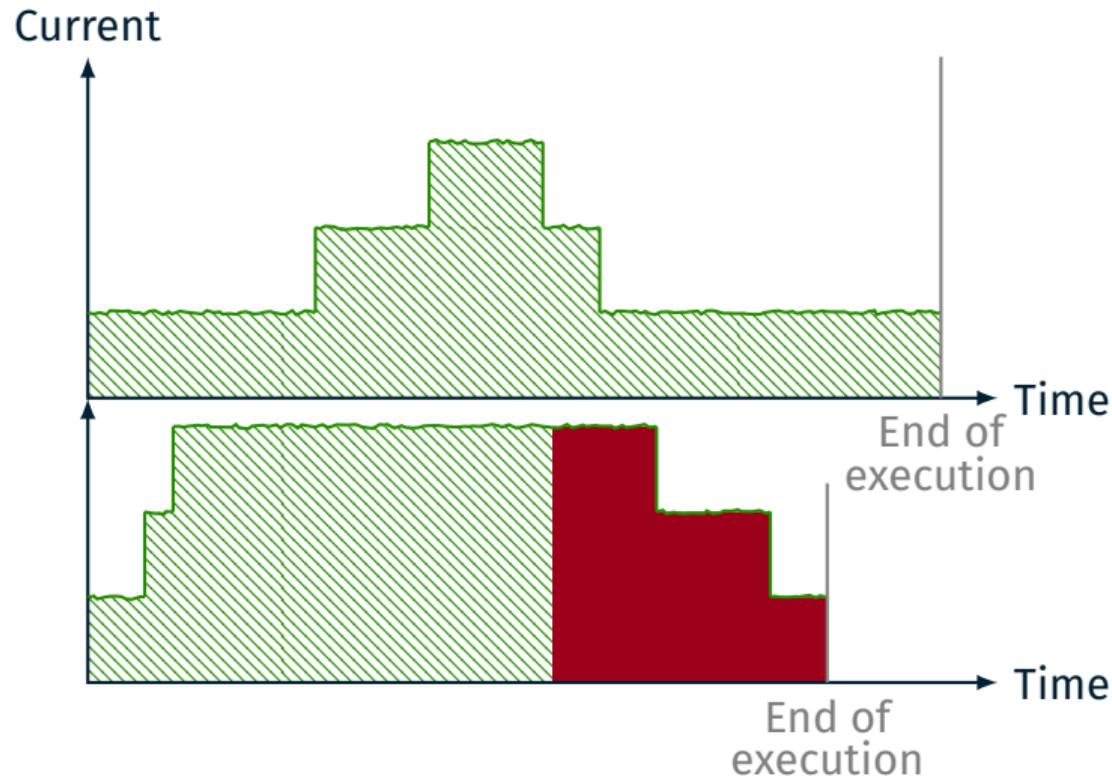
Benchmarks with Known (Worst-Case) Energy Consumption



Benchmarks with Known (Worst-Case) Energy Consumption

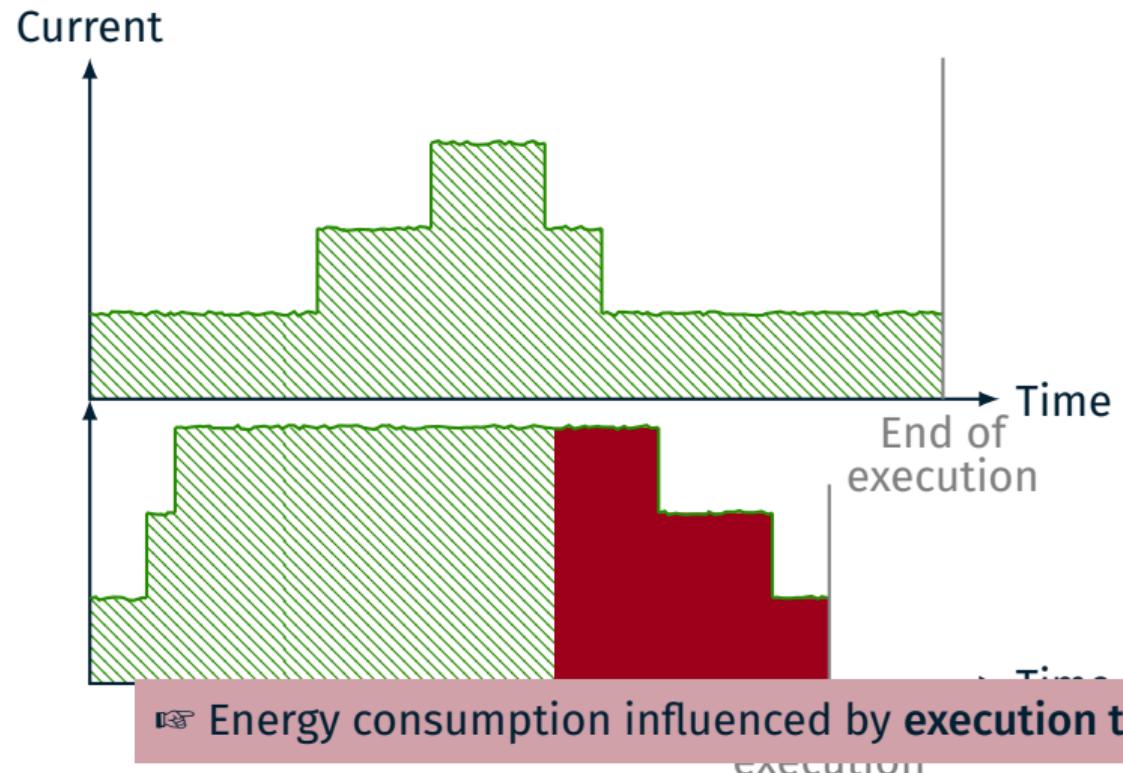


Benchmarks with Known (Worst-Case) Energy Consumption



Execution time alone
is not a measure for
energy consumption

Benchmarks with Known (Worst-Case) Energy Consumption



Execution time alone
is not a measure for
energy consumption

Path Budget: Generating the Longest Execution Path

- Estimates number of executed assembly instructions along longest path

Path Budget: Generating the Longest Execution Path

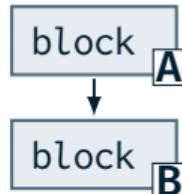
- Estimates number of executed assembly instructions along longest path
- *Goal:* Longest path is known by construction

Path Budget: Generating the Longest Execution Path

- Estimates number of executed assembly instructions along longest path
- *Goal:* Longest path is known by construction
- **Instructions have varying execution times**

Path Budget: Generating the Longest Execution Path

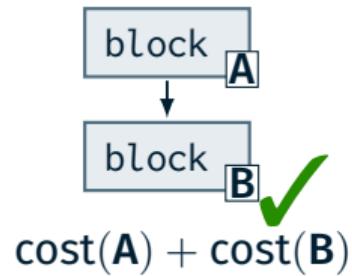
- Estimates number of executed assembly instructions along longest path
- *Goal:* Longest path is known by construction
- **Instructions have varying execution times**



$\text{cost(A)} + \text{cost(B)}$

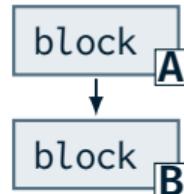
Path Budget: Generating the Longest Execution Path

- Estimates number of executed assembly instructions along longest path
- *Goal:* Longest path is known by construction
- **Instructions have varying execution times**

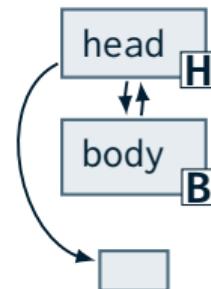


Path Budget: Generating the Longest Execution Path

- Estimates number of executed assembly instructions along longest path
- *Goal:* Longest path is known by construction
- **Instructions have varying execution times**



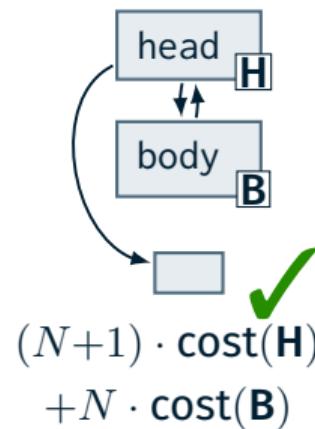
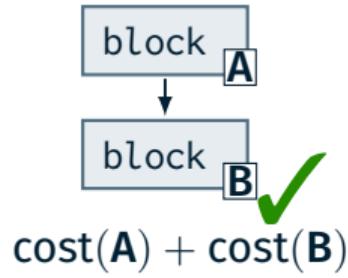
$\text{cost(A)} + \text{cost(B)}$



$$(N+1) \cdot \text{cost(H)} \\ + N \cdot \text{cost(B)}$$

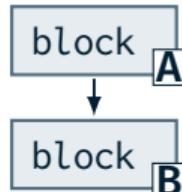
Path Budget: Generating the Longest Execution Path

- Estimates number of executed assembly instructions along longest path
- *Goal:* Longest path is known by construction
- **Instructions have varying execution times**

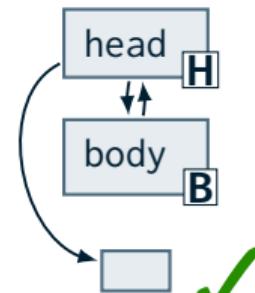


Path Budget: Generating the Longest Execution Path

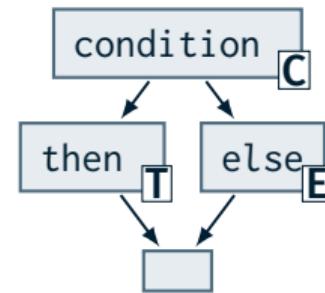
- Estimates number of executed assembly instructions along longest path
- *Goal:* Longest path is known by construction
- **Instructions have varying execution times**



$$\text{cost(A)} + \text{cost(B)}$$



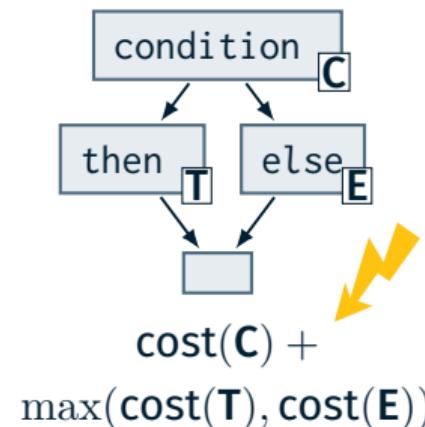
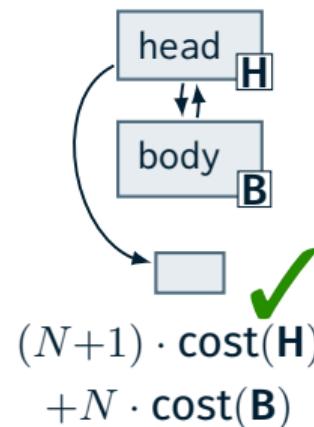
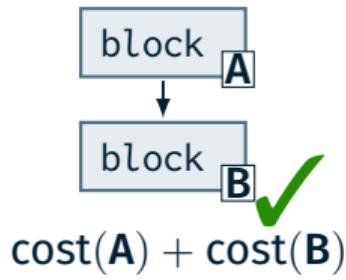
$$(N+1) \cdot \text{cost}(H) + N \cdot \text{cost}(B)$$



$$\text{cost(C)} + \max(\text{cost(T)}, \text{cost(E)})$$

Path Budget: Generating the Longest Execution Path

- Estimates number of executed assembly instructions along longest path
- *Goal:* Longest path is known by construction
- **Instructions have varying execution times**



Path Budget: Overweighting Branches (1)

- Pitfalls: Modern hardware features (e.g., caches, pipelining)
- Requirement: worst case(short path) \leq best case(long path)
- Timing of an instruction: $t_{\text{instr}} = t_{\text{icache}} + t_{\text{dcache}} + t_{\text{pipeline}} + t_{\text{calc}}$

Path Budget: Overweighting Branches (1)

- Pitfalls: Modern hardware features (e.g., caches, pipelining)
- Requirement: worst case(short path) \leq best case(long path)
- Timing of an instruction: $t_{\text{instr}} = t_{\text{icache}} + t_{\text{dcache}} + t_{\text{pipeline}} + t_{\text{calc}}$

and r4, r5

$t_{\text{and},\min} = 1$

ldr r0, [r0, 42]

$t_{\text{ldr},\max} = 7$

$\Rightarrow 7 \times \text{and}$ compensates $1 \times \text{ldr}$ in the worst case

Path Budget: Overweighting Branches (1)

- Pitfalls: Modern hardware features (e.g., caches, pipelining)
- Requirement: worst case(short path) \leq best case(long path)
- Timing of an instruction: $t_{\text{instr}} = t_{\text{icache}} + t_{\text{dcache}} + t_{\text{pipeline}} + t_{\text{calc}}$

and r4, r5

$t_{\text{and},\min} = 1$

ldr r0, [r0, 42]

$t_{\text{ldr},\max} = 7$

$\Rightarrow 7 \times \text{and}$ compensates $1 \times \text{ldr}$ in the worst case

- Find maximum of worst-case timings between all instructions

\Rightarrow Minimum overweighting factor: $\mathcal{F} = 7$

Path Budget: Overweighting Branches (1)

- Pitfalls: Modern hardware features (e.g., caches, pipelining)
- Requirement: worst case(short path) \leq best case(long path)
- Timing of an instruction: $t_{\text{instr}} = t_{\text{icache}} + t_{\text{dcache}} + t_{\text{pipeline}} + t_{\text{calc}}$

and r4, r5

$t_{\text{and},\min} = 1$

ldr r0, [r0, 42]

$t_{\text{ldr},\max} = 7$

$\Rightarrow 7 \times \text{and}$ compensates $1 \times \text{ldr}$ in the worst case

- Find maximum of worst-case timings between all instructions

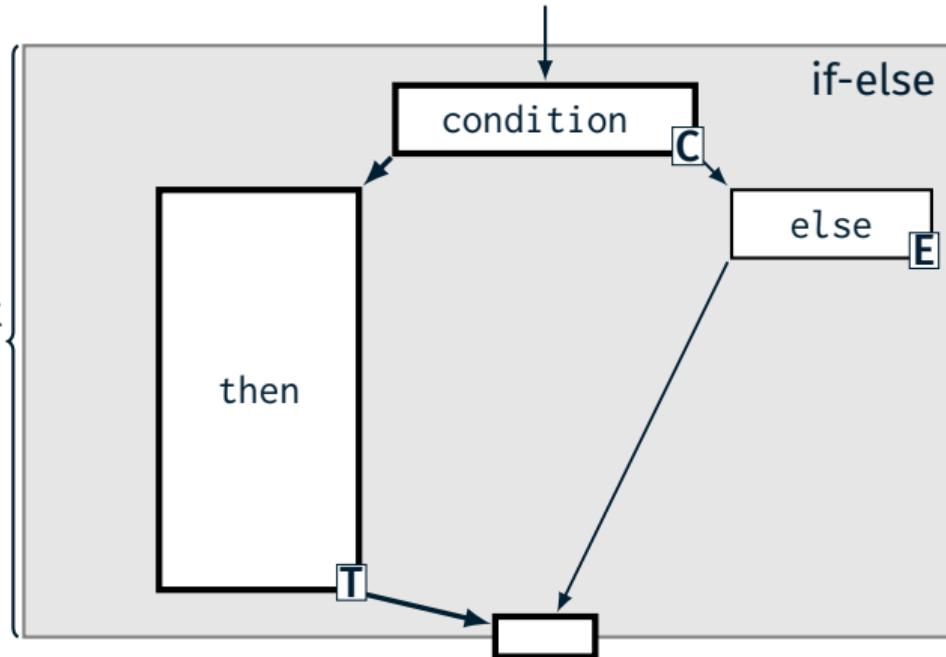
\Rightarrow Minimum overweighting factor: $\mathcal{F} = 7$

\Rightarrow Choose sufficiently large: $\mathcal{F} = 25$

Path Budget: Overweighting Branches (2)

Generate if-else with budget 1010 and overweighting factor $\mathcal{F} = 25$

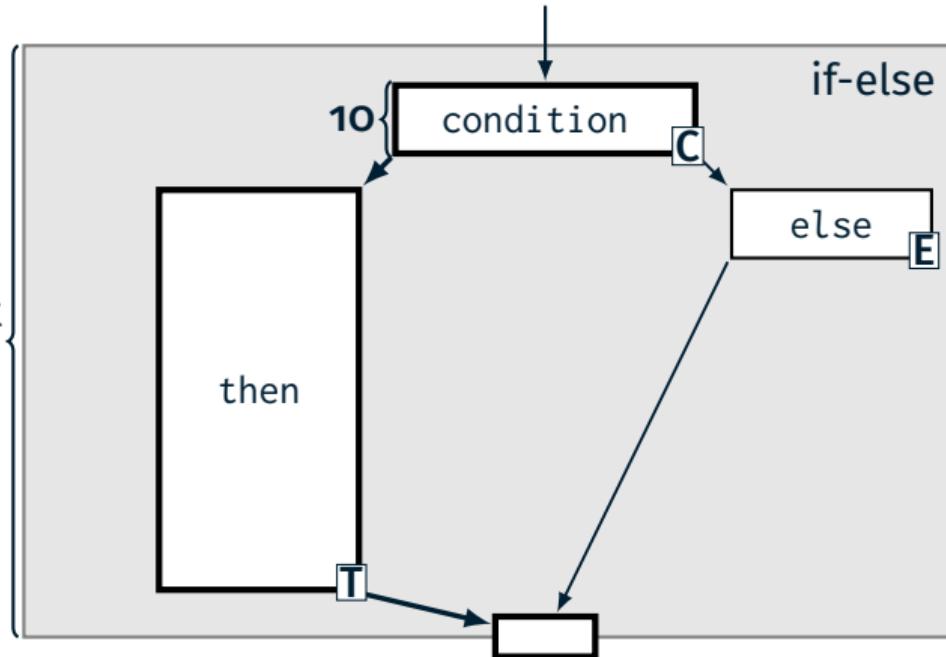
Budget
1010



Path Budget: Overweighting Branches (2)

Generate if-else with budget 1010 and overweighting factor $\mathcal{F} = 25$

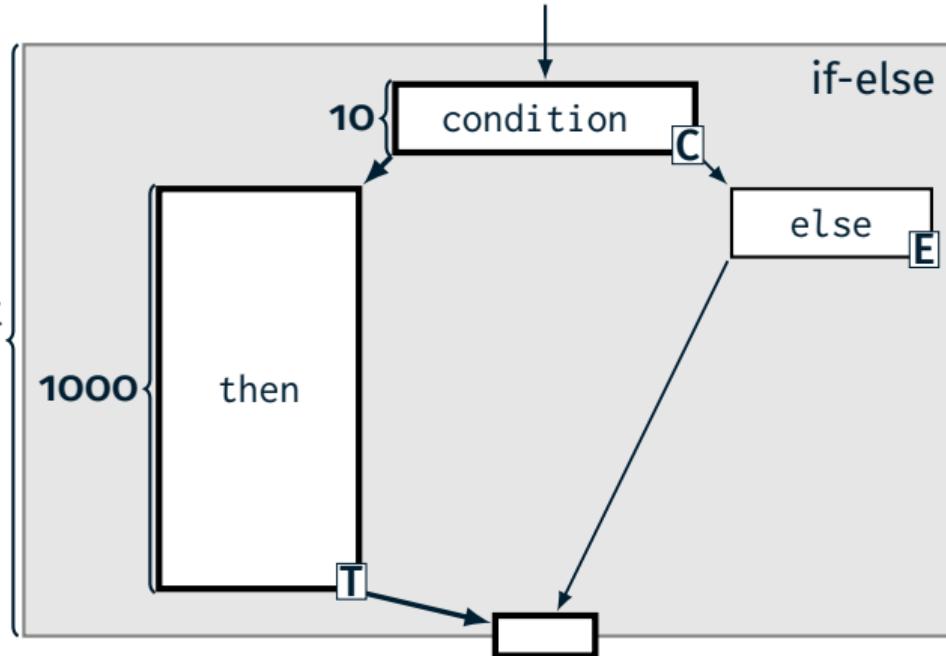
Budget
1010



Path Budget: Overweighting Branches (2)

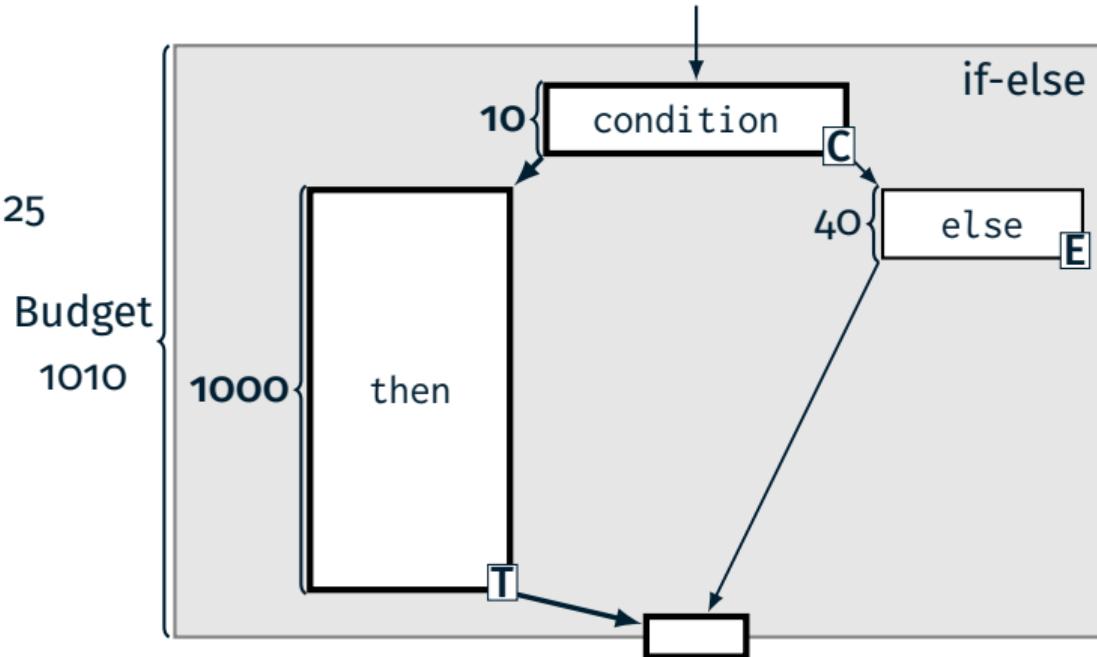
Generate if-else with budget 1010 and overweighting factor $\mathcal{F} = 25$

Budget
1010



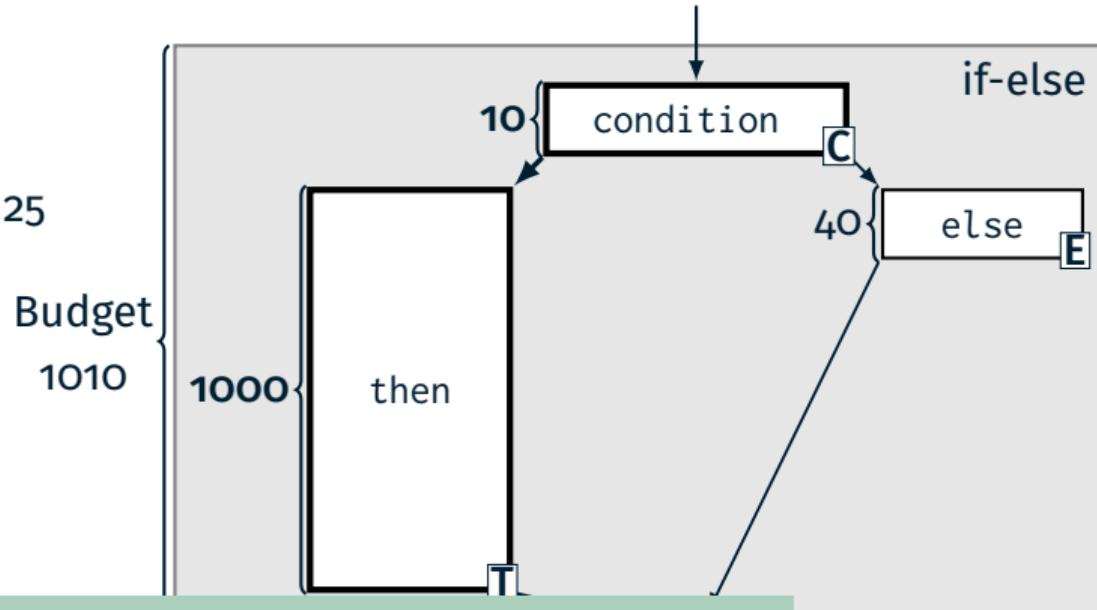
Path Budget: Overweighting Branches (2)

Generate if-else with budget 1010 and overweighting factor $\mathcal{F} = 25$



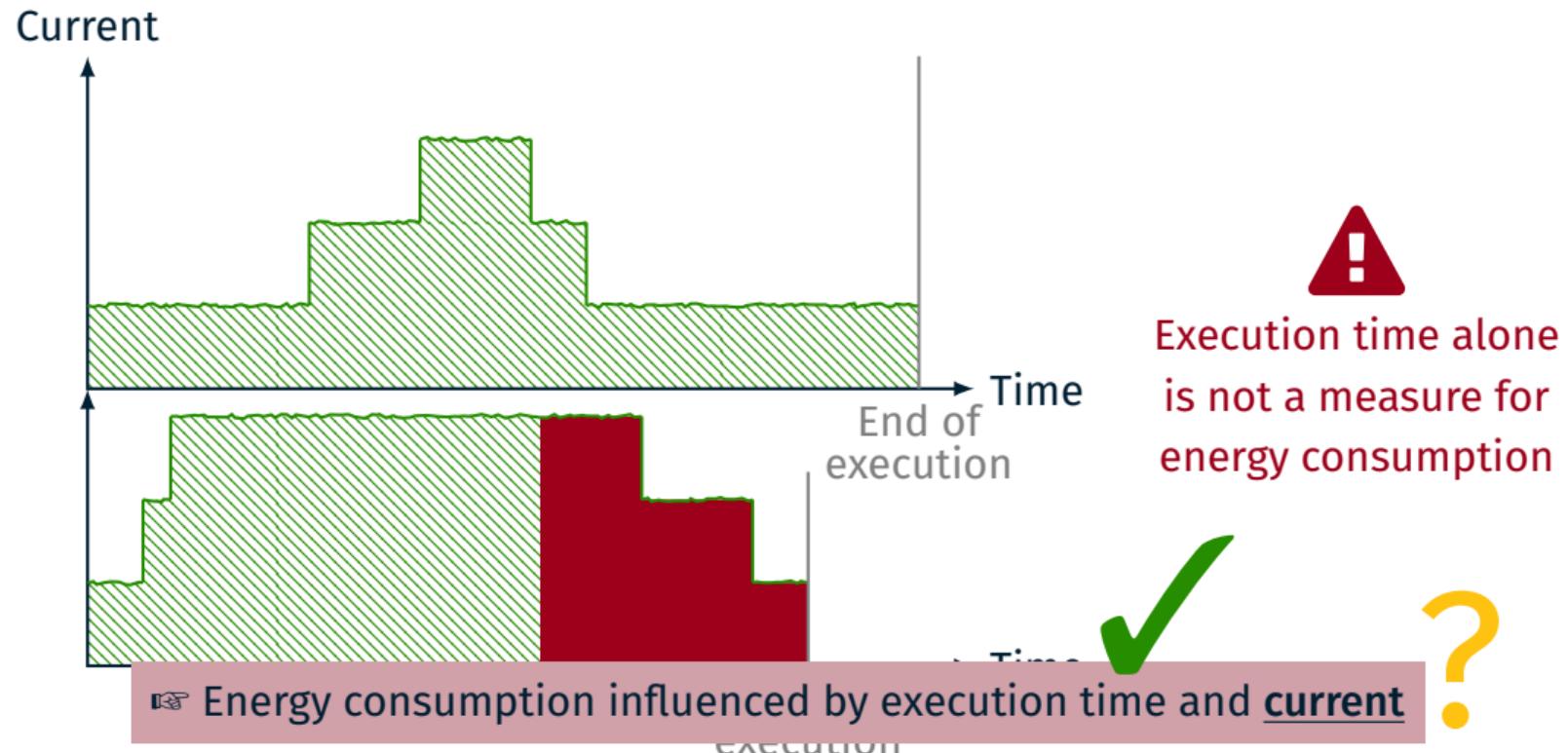
Path Budget: Overweighting Branches (2)

Generate if-else with budget 1010 and overweighting factor $\mathcal{F} = 25$



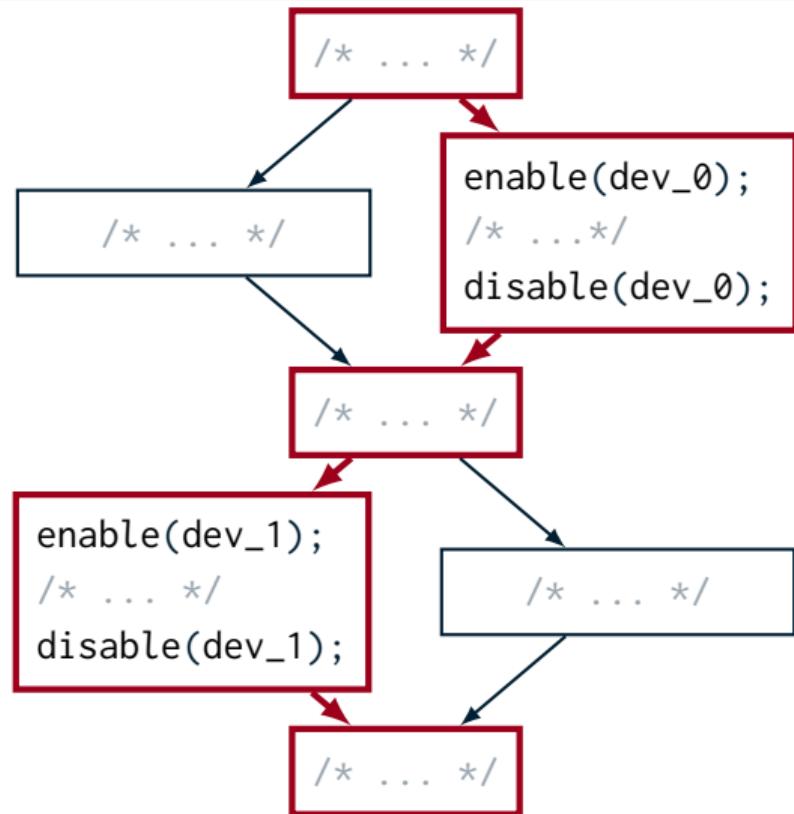
✓ Longest execution path known by construction

Benchmarks with Known (Worst-Case) Energy Consumption



Generating the Path With Maximum Energy Consumption

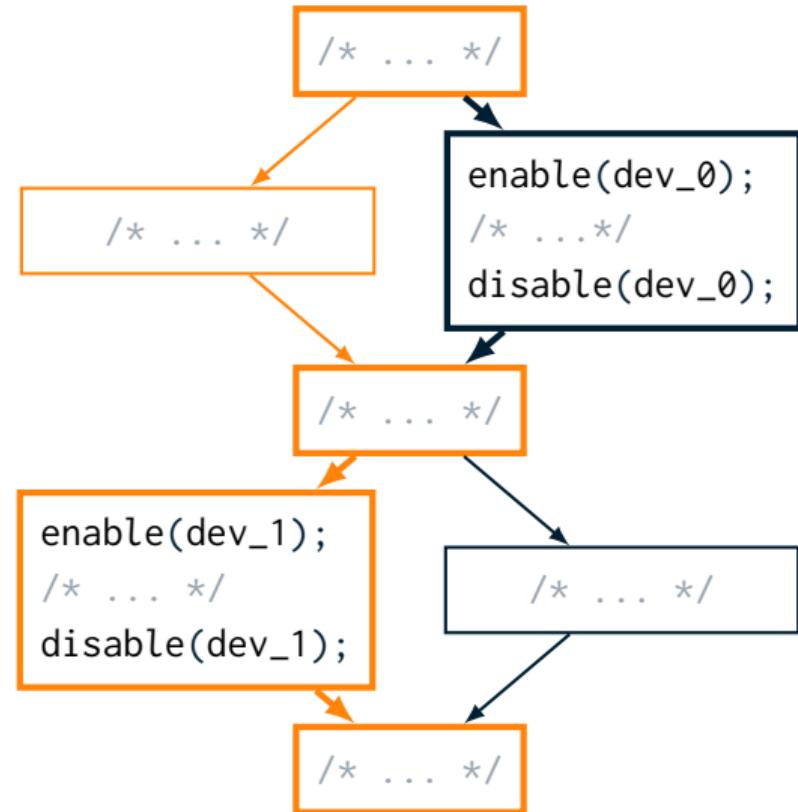
Approach: Generate device interactions
along worst-case path



Generating the Path With Maximum Energy Consumption

Approach: Generate device interactions along worst-case path

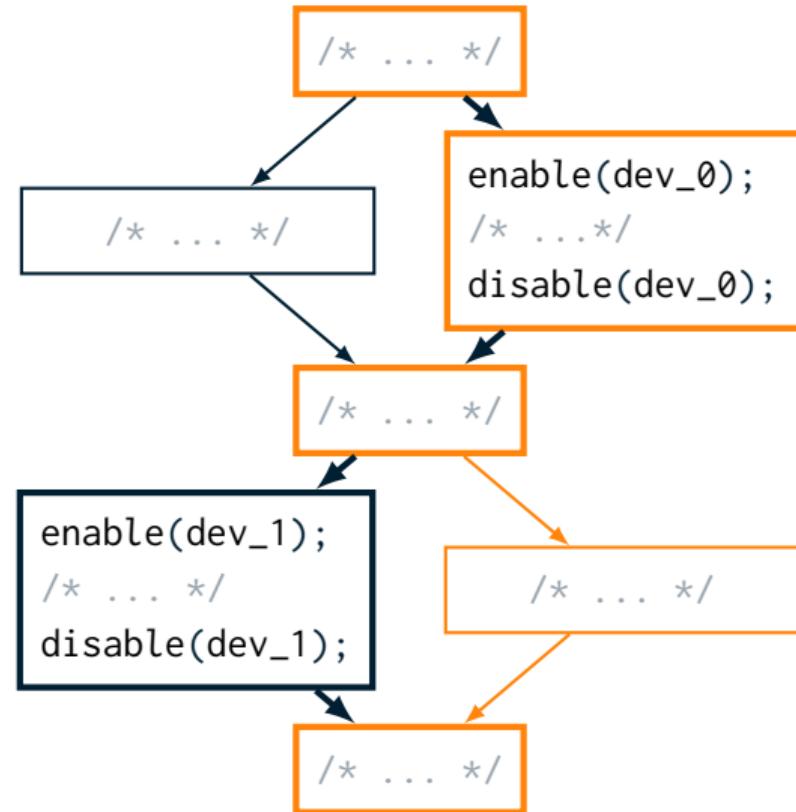
- Worst-case path shares parts with non-worst-case paths
- Device interactions also along non-worst-case paths



Generating the Path With Maximum Energy Consumption

Approach: Generate device interactions along worst-case path

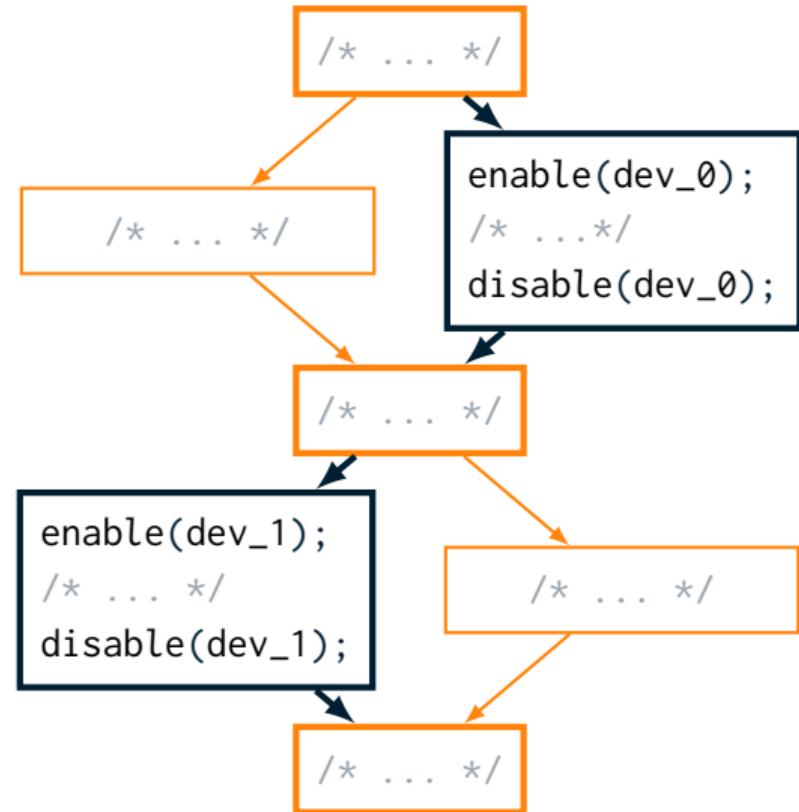
- Worst-case path shares parts with non-worst-case paths
- Device interactions also along non-worst-case paths



Generating the Path With Maximum Energy Consumption

Approach: Generate device interactions along worst-case path

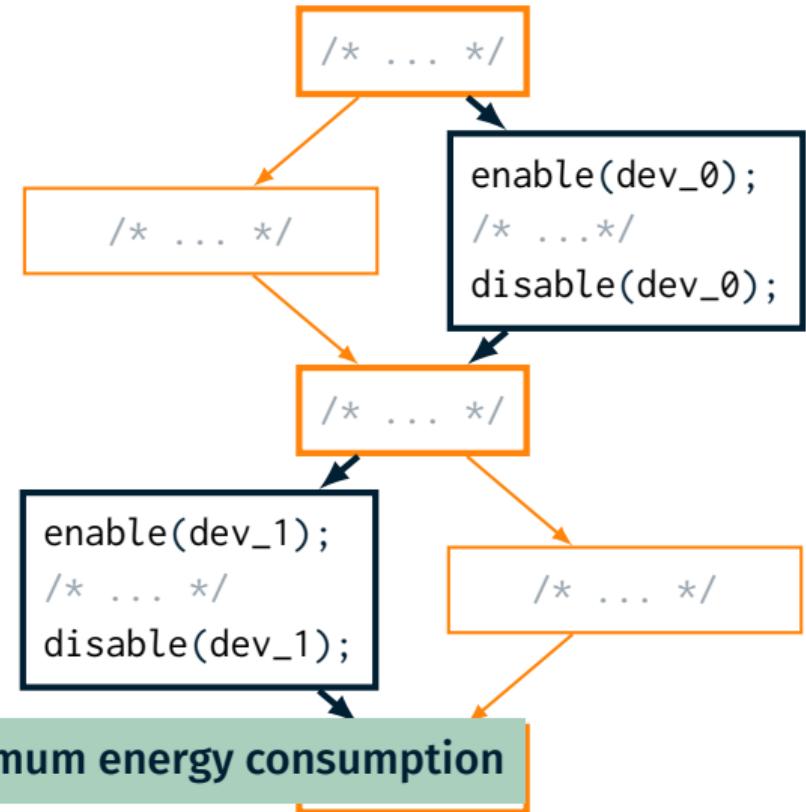
- Worst-case path shares parts with non-worst-case paths
- Device interactions also along non-worst-case paths



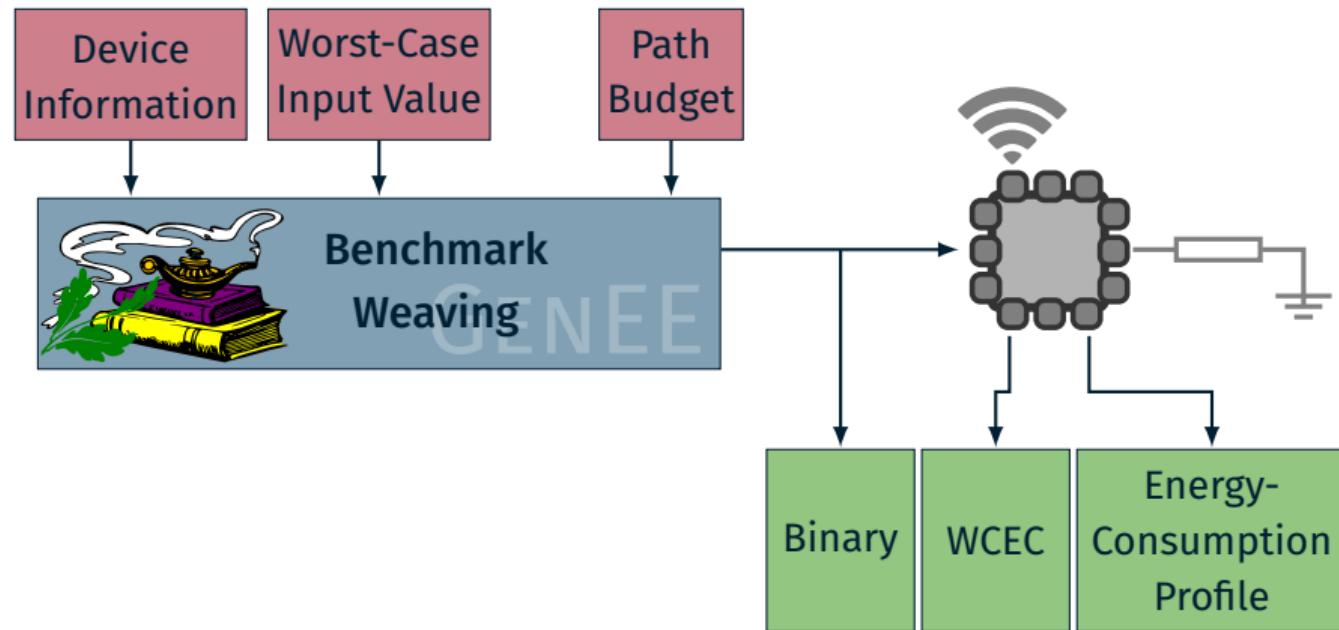
Generating the Path With Maximum Energy Consumption

Approach: Generate device interactions along worst-case path

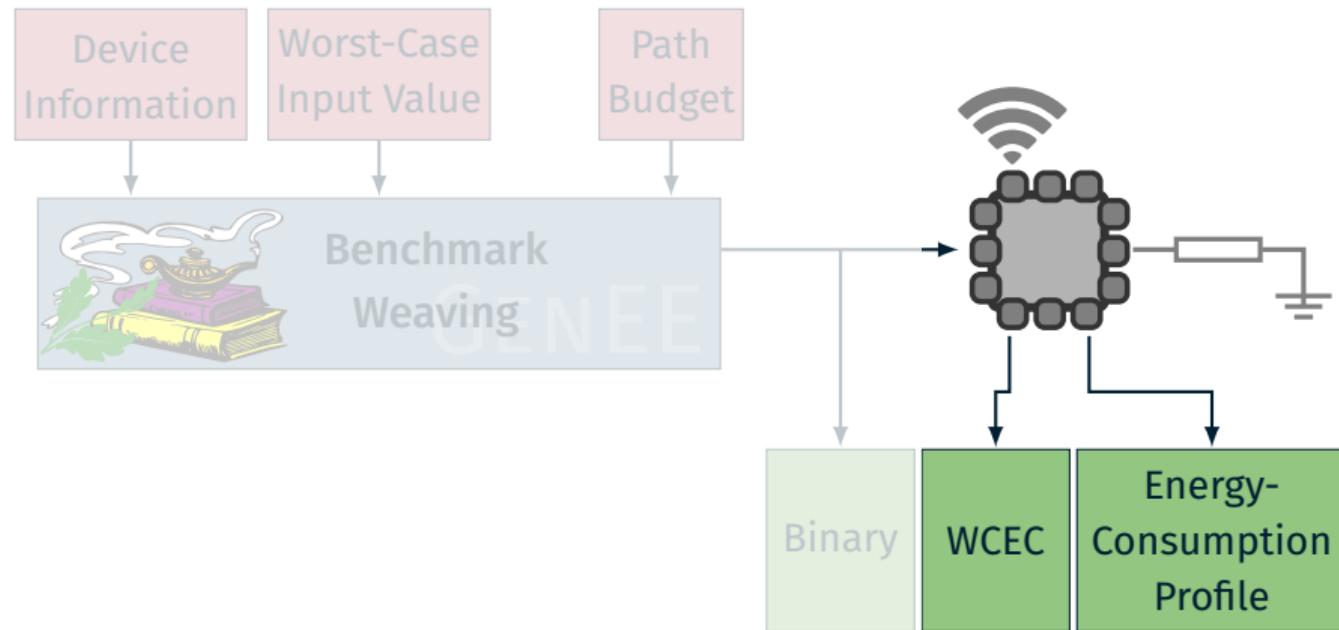
- Worst-case path shares parts with non-worst-case paths
- Device interactions also along non-worst-case paths



GENEE: Overview

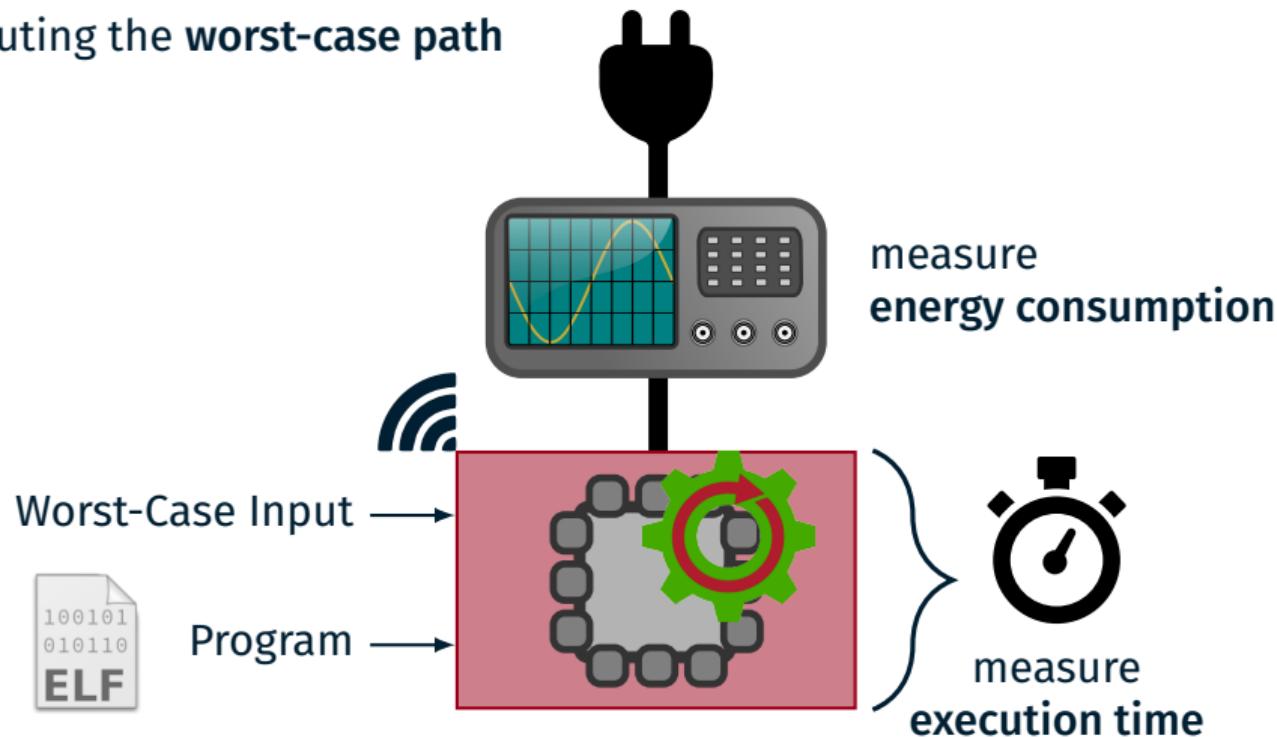


GENEE: Overview



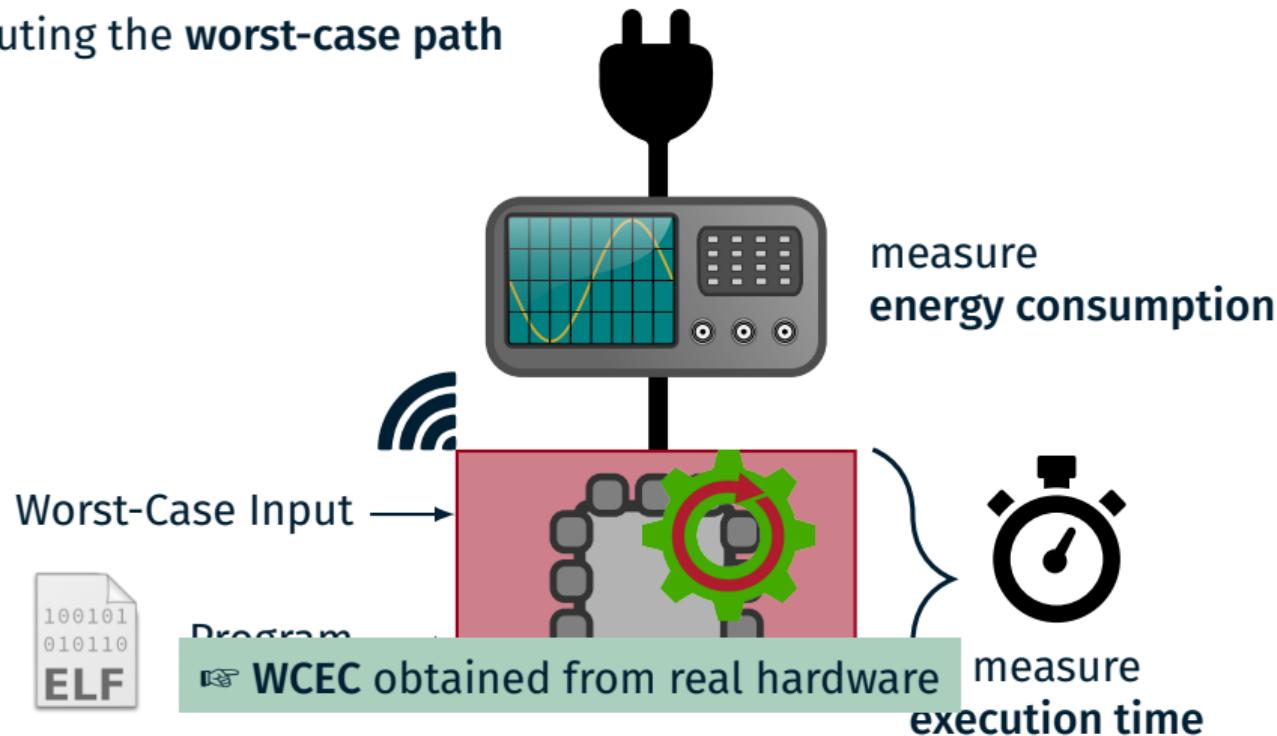
Determining (Worst-Case) Parameters

Approach: Measure execution time and energy consumed while executing the worst-case path



Determining (Worst-Case) Parameters

Approach: Measure execution time and energy consumed while executing the worst-case path



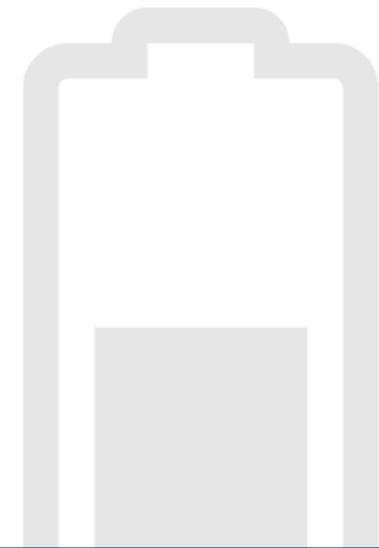
Outline

Motivation

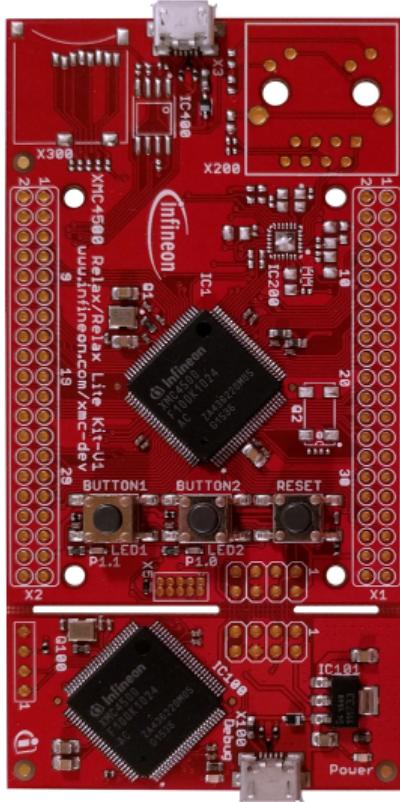
The GENEE Benchmark Generator

Evaluation

Conclusion



Evaluation: Setup



ARM Cortex-M4

- 3-stage pipeline
- Frequency: 30 MHz
- 4 KB instruction cache

Four External Devices

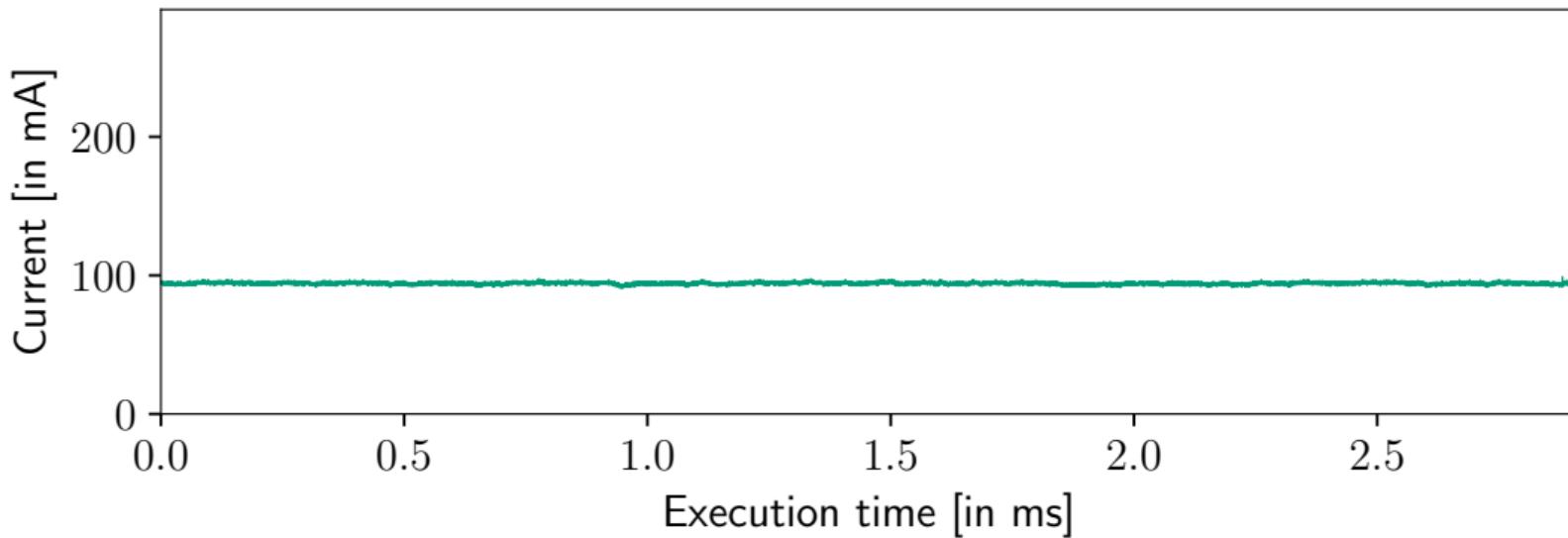
- $56\Omega \Rightarrow 59\text{mA } @3.3\text{V}$

Energy Measurement

- Tektronix MSO 4034 @2.5GS/s
- Instrument shunt 0.51Ω

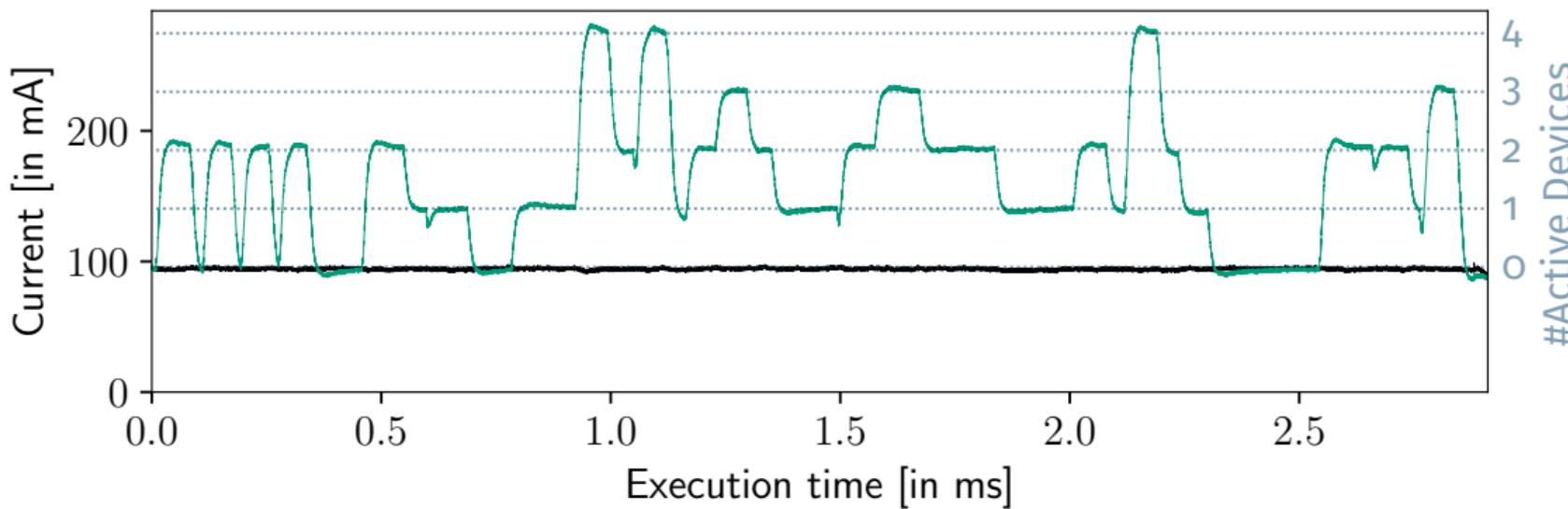
Assessing the Effects of Devices

- **Approach:** Trace current over time of **worst-case path**
- > 7 Million samples (per Channel)
- Without devices: 95 mA (constant)



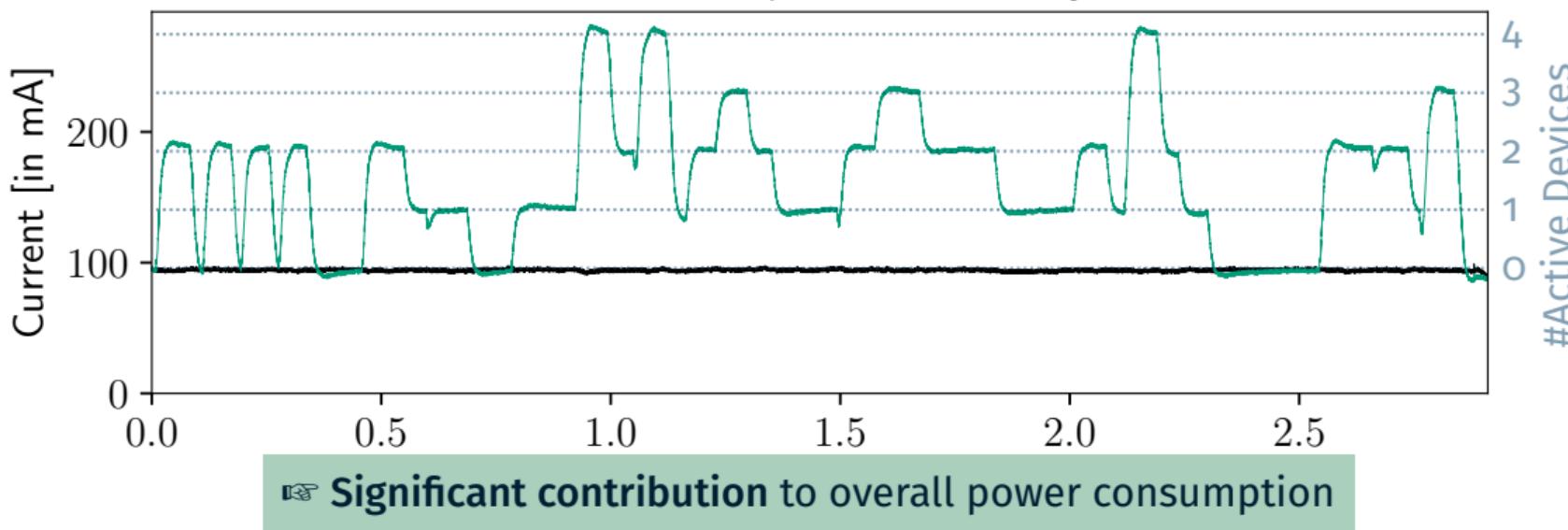
Assessing the Effects of Devices

- **Approach:** Trace current over time of **worst-case path**
- > 7 Million samples (per Channel)
- Without devices: 95 mA (constant), with devices: **Up to 280 mA**



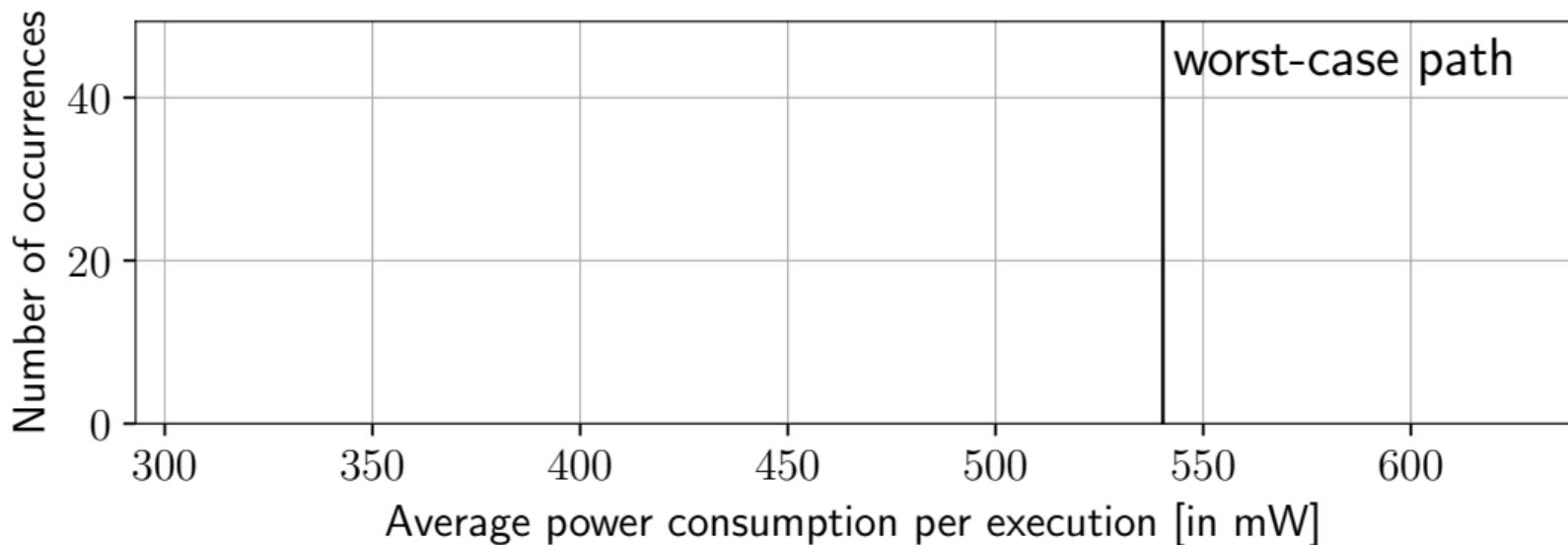
Assessing the Effects of Devices

- **Approach:** Trace current over time of **worst-case path**
- > 7 Million samples (per Channel)
- Without devices: 95 mA (constant), with devices: **Up to 280 mA**



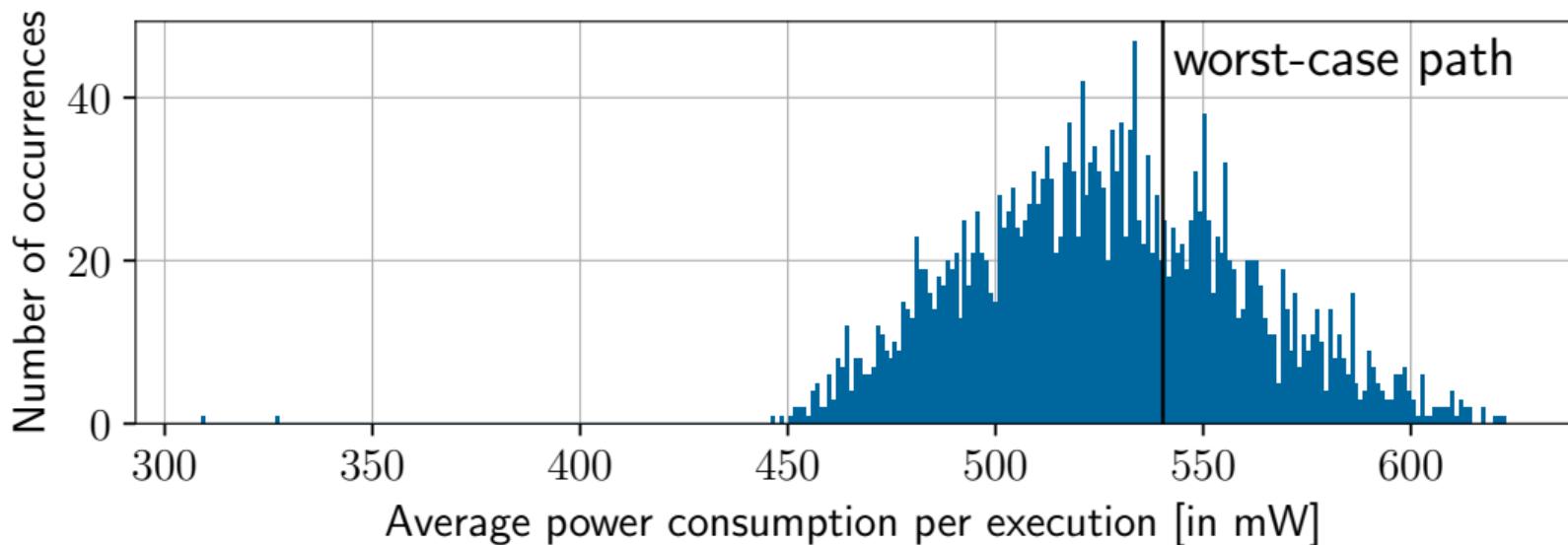
Assessing the Behavior of Non-Worst-Case Paths

- **Approach:** Average power consumption for 2500 different input values
- Average power consumption = $\frac{\text{Energy consumption}}{\text{Execution time}}$



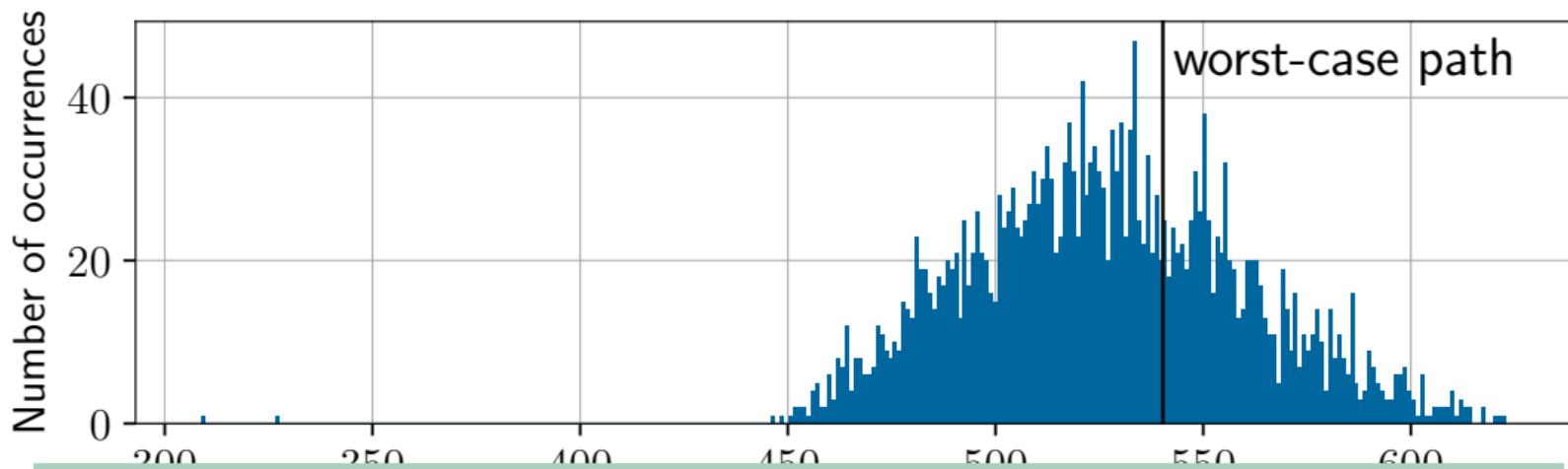
Assessing the Behavior of Non-Worst-Case Paths

- Approach: Average power consumption for 2500 different input values
- Average power consumption = $\frac{\text{Energy consumption}}{\text{Execution time}}$



Assessing the Behavior of Non-Worst-Case Paths

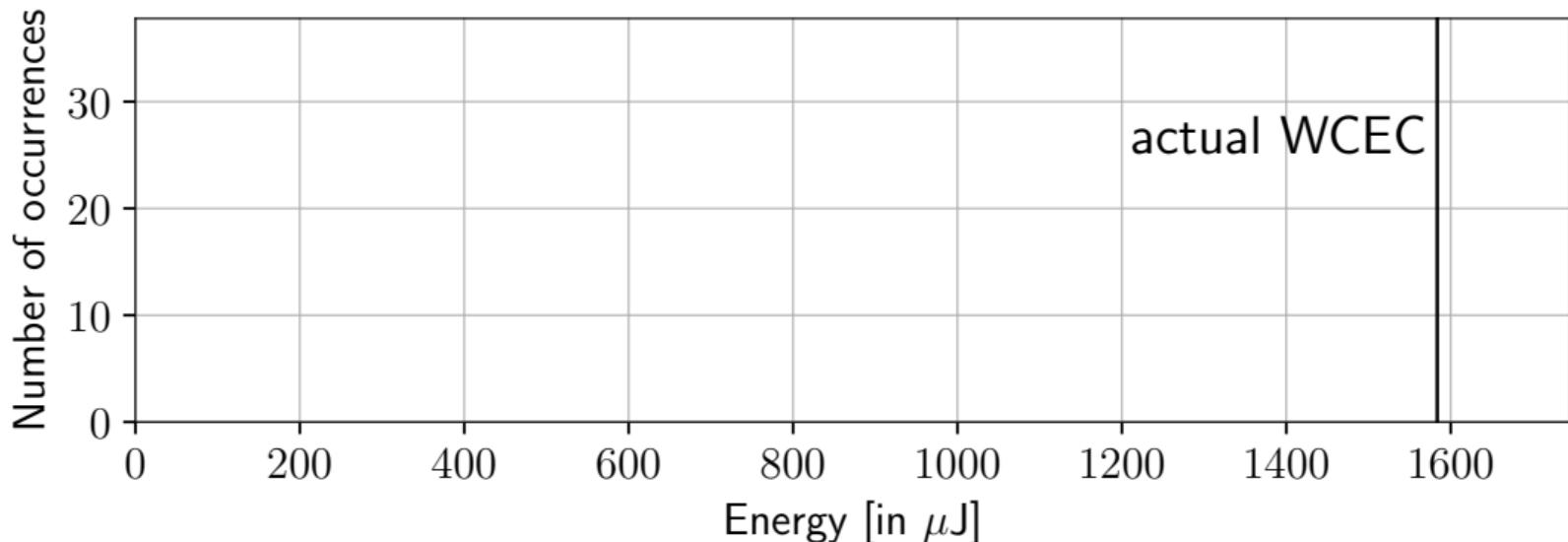
- Approach: Average power consumption for 2500 different input values
- Average power consumption = $\frac{\text{Energy consumption}}{\text{Execution time}}$



☞ Complex benchmarks with device activations along non-worst-case paths

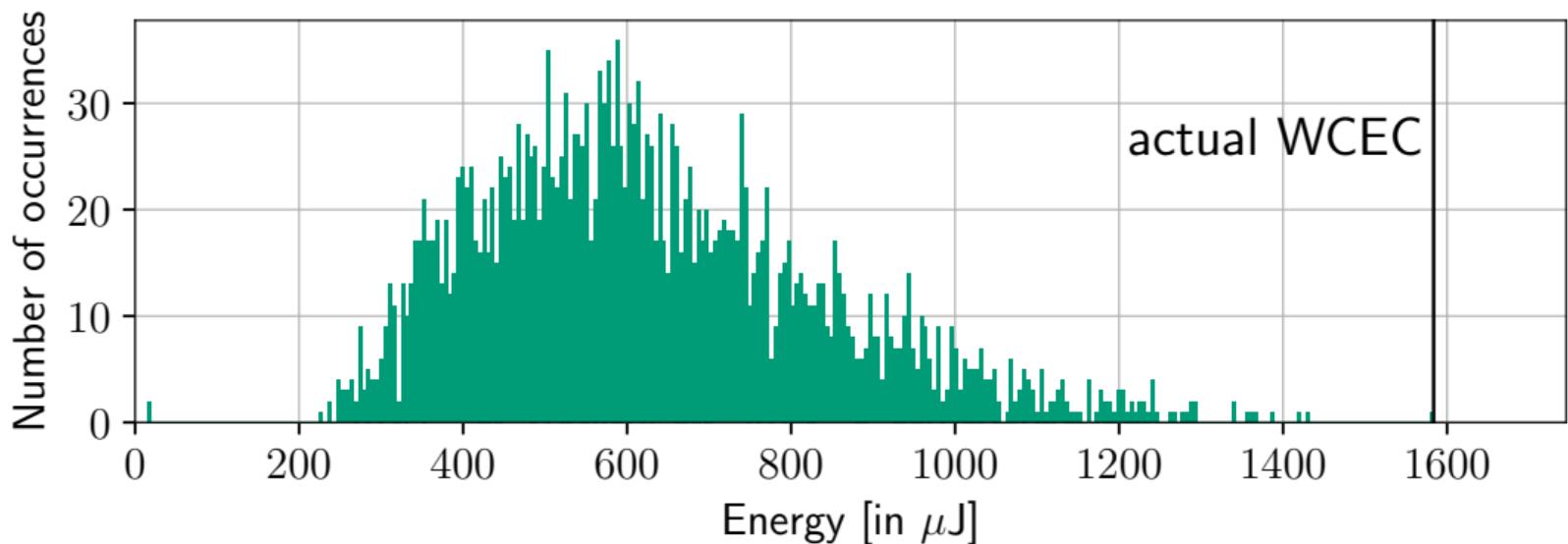
Validating the Generated WCEC

- Approach: Compare WCEC against consumptions for different input values



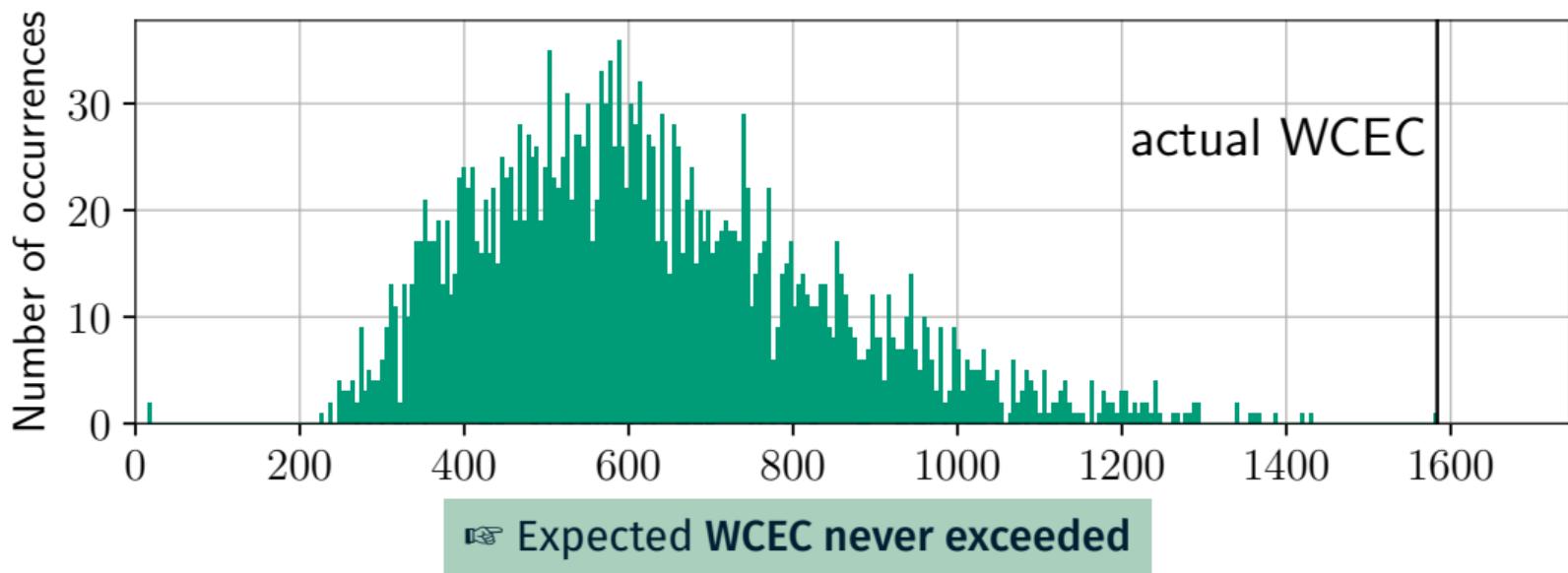
Validating the Generated WCEC

- Approach: Compare WCEC against consumptions for different input values



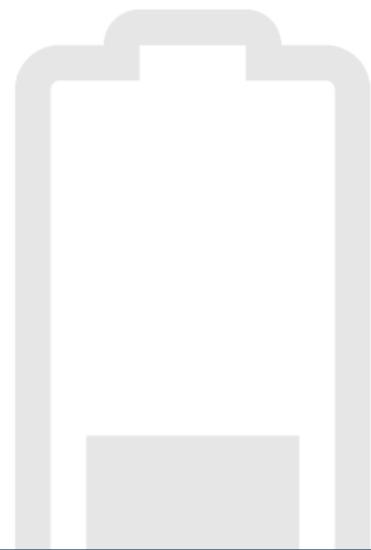
Validating the Generated WCEC

- Approach: Compare WCEC against consumptions for different input values



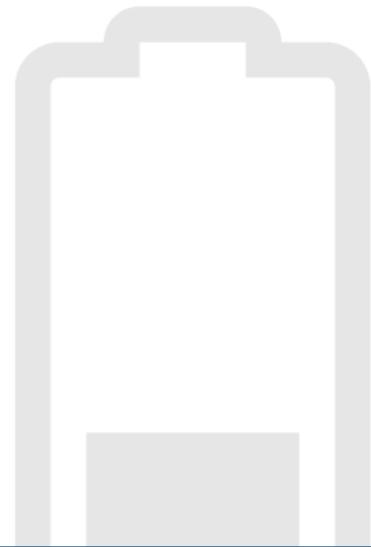
Conclusion

- *Motivation: Missing baselines for existing benchmarks*



Conclusion

- *Motivation: Missing baselines for existing benchmarks*
- Generate benchmarks with known **worst-case paths**
- Extract WCEC via **concrete execution** on target platform

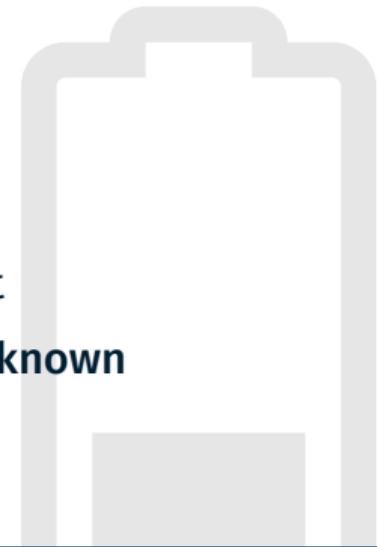


Conclusion

- **Motivation: Missing baselines for existing benchmarks**
- Generate benchmarks with known **worst-case paths**
- Extract WCEC via **concrete execution** on target platform



- ✓ Complex benchmarks
- ✓ Execution path depends on input
- ✓ **Worst-case energy consumption known**



Questions?



The source code is available at:
<https://gitlab.cs.fau.de/gene>

