FusionClock: Energy-Optimal Clock-Tree Reconfigurations for Energy-Constrained Real-Time Systems

GO TO

menti.com

ENTER THE CODE

5124 8694

0

12th July 2023

Eva Dengler, Phillip Raffeck, Simon Schuster, Peter Wägemann

Friedrich-Alexander-Universität Erlangen-Nürnberg

Supported by the DFG under the grant WA 5186/1-1 (Watwa)

Lehrstuhl für Verteilte Systeme
und Betriebssysteme

FAU Friedrich-Alexander-Universität
Technische Fakultät

DFG

Application Scenarios

Application Scenarios

- embedded real-time systems: worst-case execution time (WCET), worst-case energy consumption (WCEC)

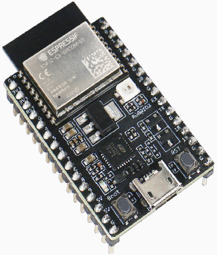
Application Scenarios

- embedded real-time systems: worst-case execution time (WCET), worst-case energy consumption (WCEC)
- devices massively influence timing and energy behaviour

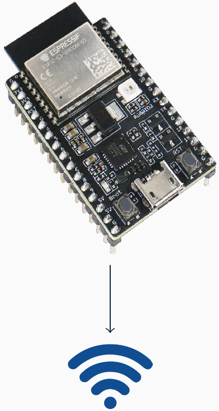
Application Scenarios

- embedded real-time systems: worst-case execution time (WCET), worst-case energy consumption (WCEC)
- devices massively influence timing and energy behaviour
  ⇒ reduce energy consumption for longer battery life

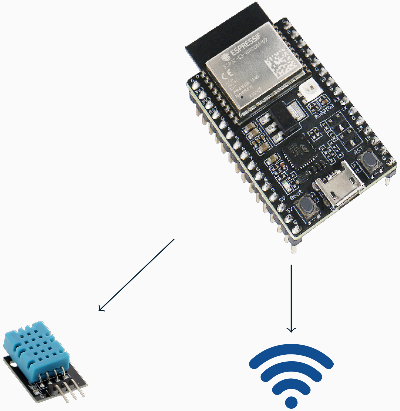
Devices on Embedded Systems



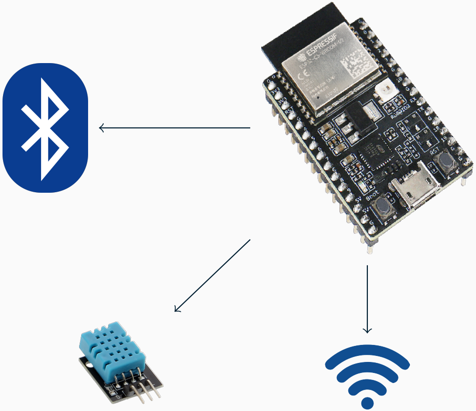
Devices on Embedded Systems



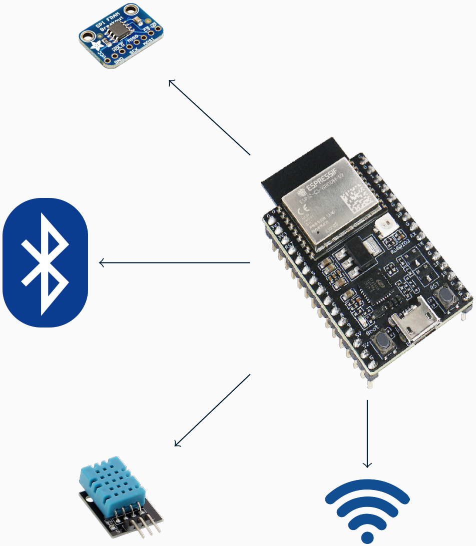
Devices on Embedded Systems



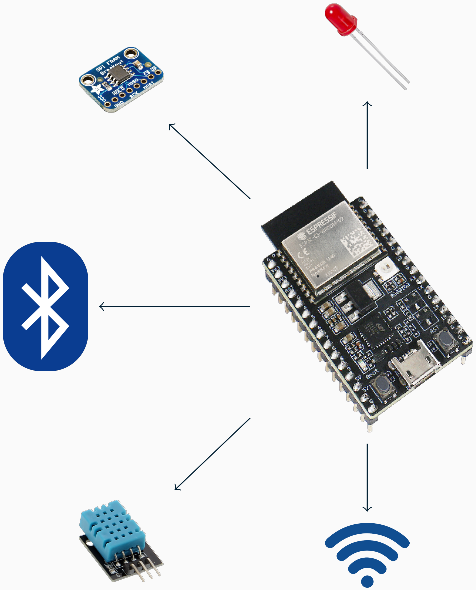
Devices on Embedded Systems



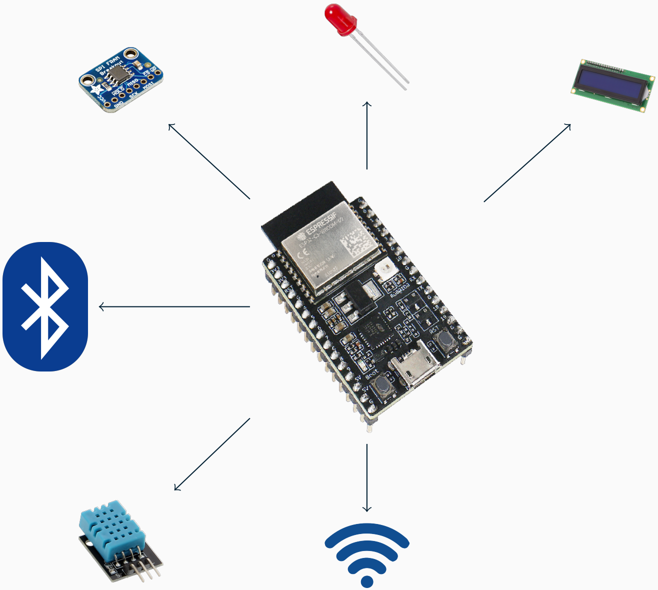
Devices on Embedded Systems



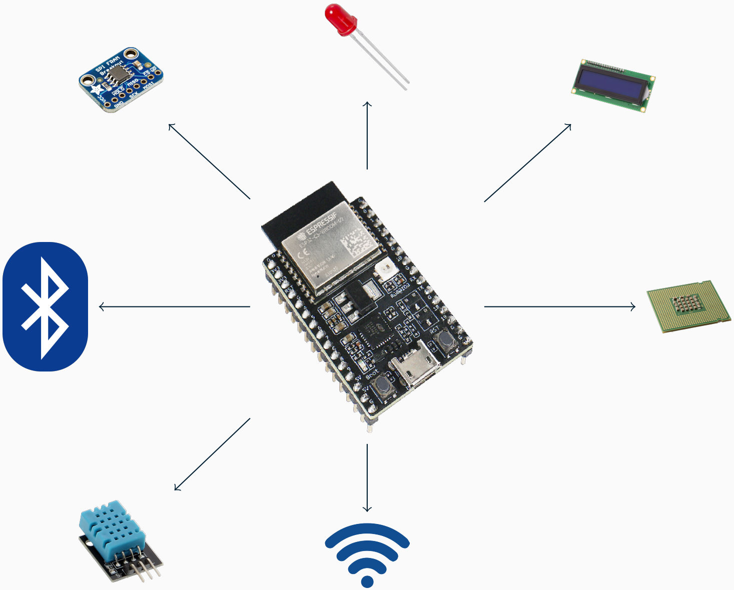
Devices on Embedded Systems



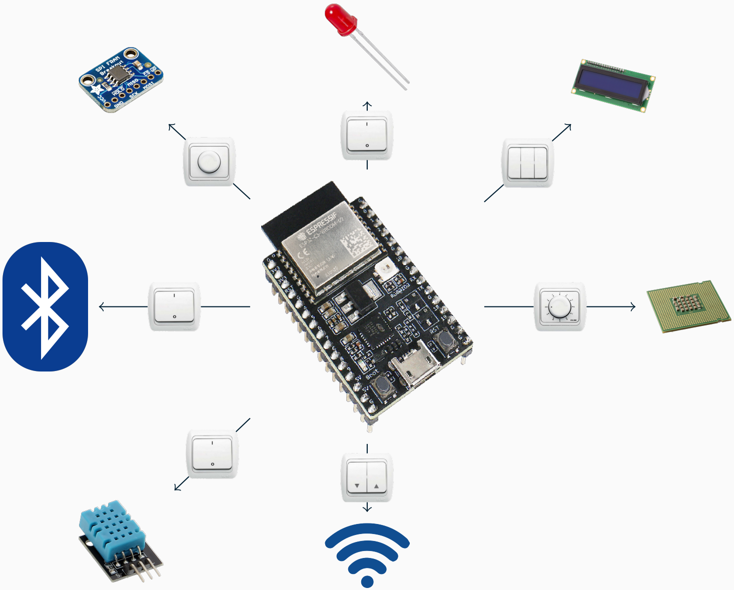
Devices on Embedded Systems



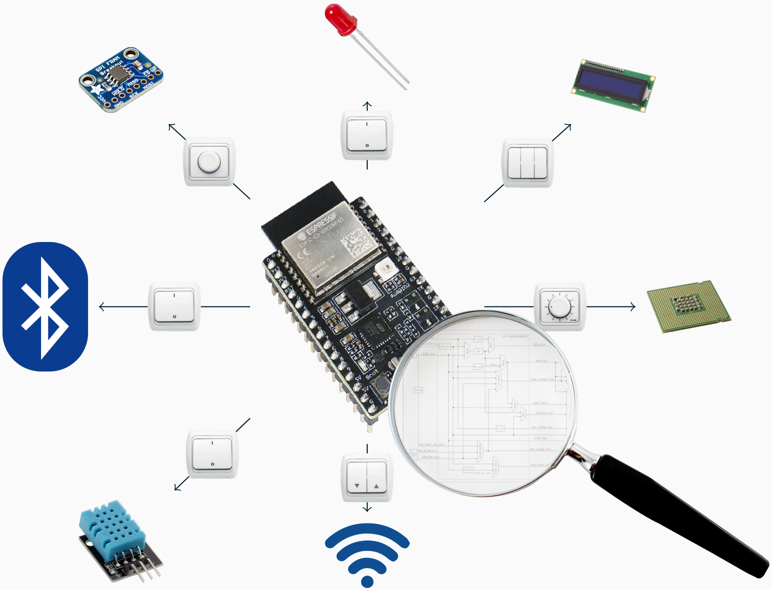
Devices on Embedded Systems



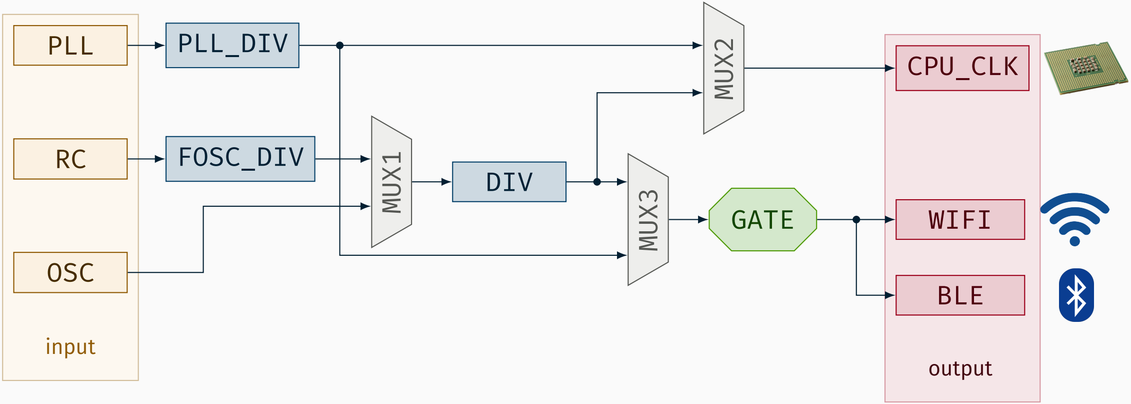
Devices on Embedded Systems



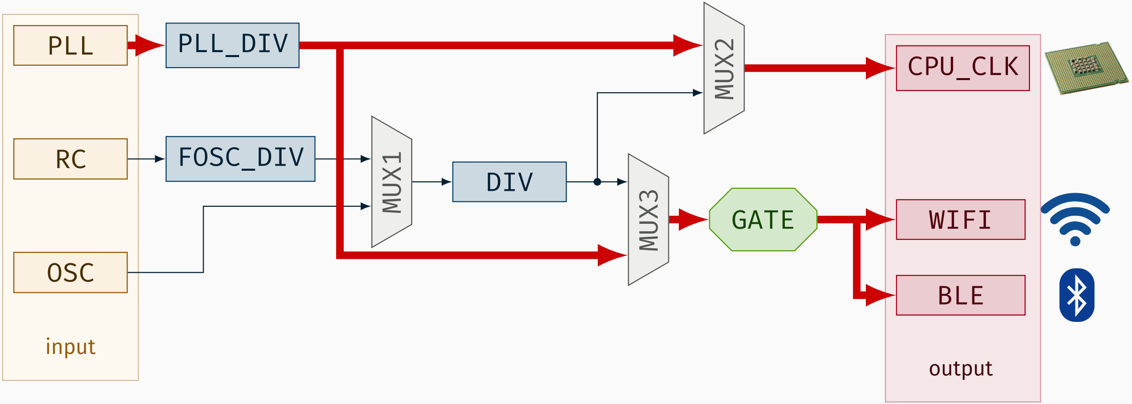
Devices on Embedded Systems

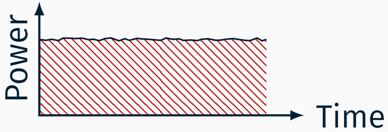


The Clock Tree

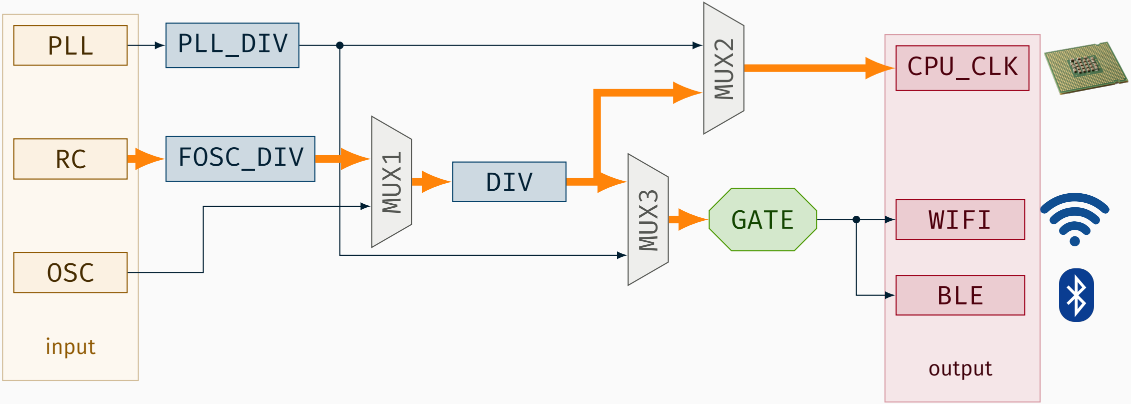


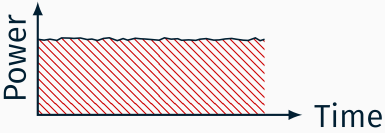
The Clock Tree



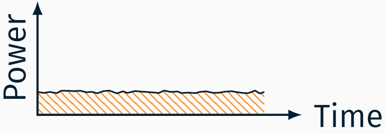


The Clock Tree





v.s.



Problem Description

System Model / Requirements for FusionClock

- single-core platforms

System Model / Requirements for FusionClock

- single-core platforms
  - configurable via **clock tree**

System Model / Requirements for FusionClock

- single-core platforms
  - configurable via **clock tree**
- static and sound model of the system for WCET/WCEC analyses

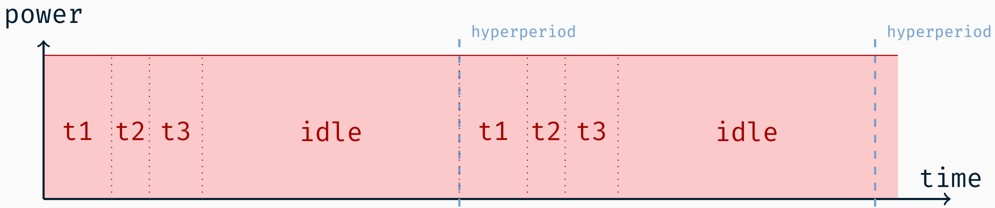
System Model / Requirements for FusionClock

- single-core platforms
  - configurable via **clock tree**
- static and sound model of the system for WCET/WCEC analyses
  - including the **devices** of the system

System Model / Requirements for FusionClock

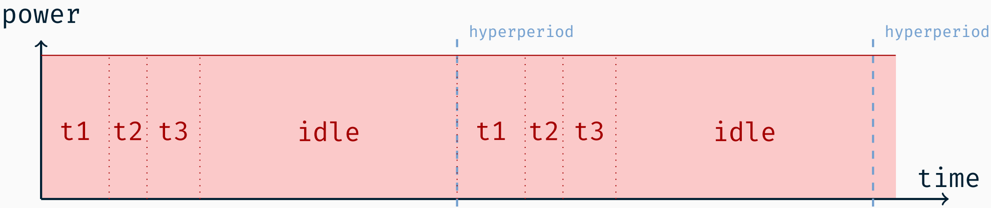
- single-core platforms
  - configurable via **clock tree**
- static and sound model of the system for WCET/WCEC analyses
  - including the **devices** of the system
- strictly periodic, cyclic task model
  - **time-triggered schedule**

Problem Analysis



all-always-on approach

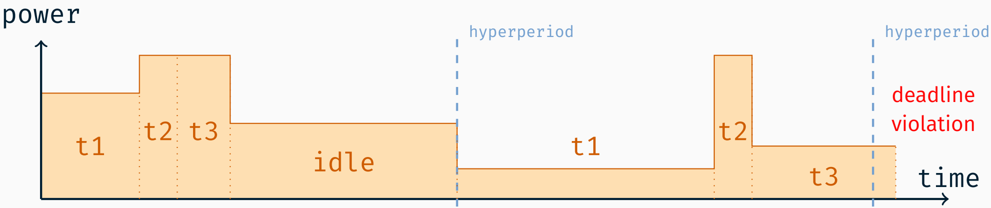
Problem Analysis



all-always-on approach

× minimzation of energy consumption

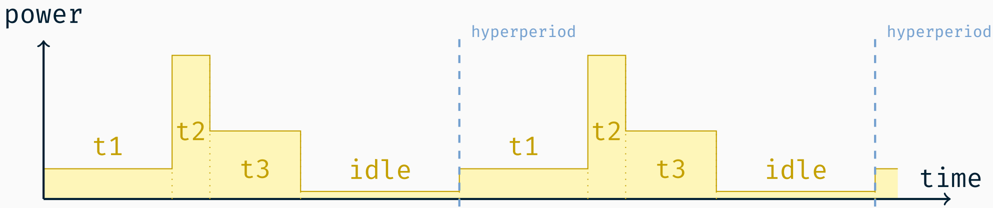
Problem Analysis



feedback-based approach: reconfigurations during execution

- minimzation of energy consumption
- × real-time guarantees

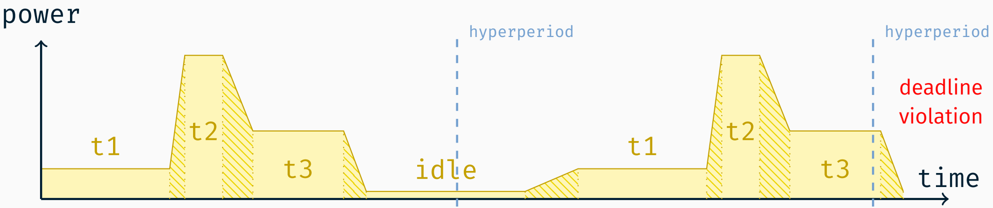
Problem Analysis



static approach: analysis before execution

- minimzation of energy consumption
- real-time guarantees

Problem Analysis



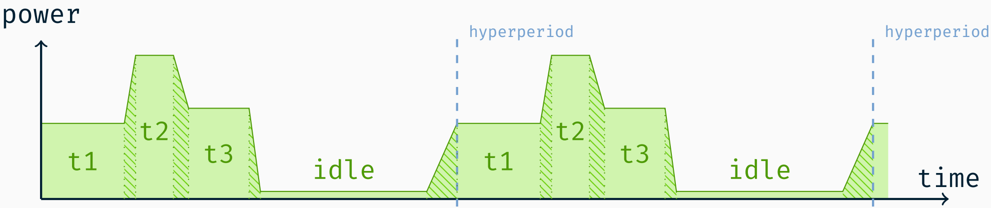
static approach without reconfiguration penalties

- minimzation of energy consumption
- × real-time guarantees
- × consideration of reconfiguration costs

Problems

1. CPU-only approaches...
   - neglect energy consumption of **devices**
   - ignore **dependencies of devices** and **clock-tree configurations**
2. **no guarantees** of feedback-based approaches
3. missing **reconfiguration penalties**

Concept of FusionClock



static approach with reconfiguration penalties

- ✓ minimzation of energy consumption
- ✓ real-time guarantees
- ✓ consideration of reconfiguration costs

The FusionClock Approach

Overview over the FusionClock Approach

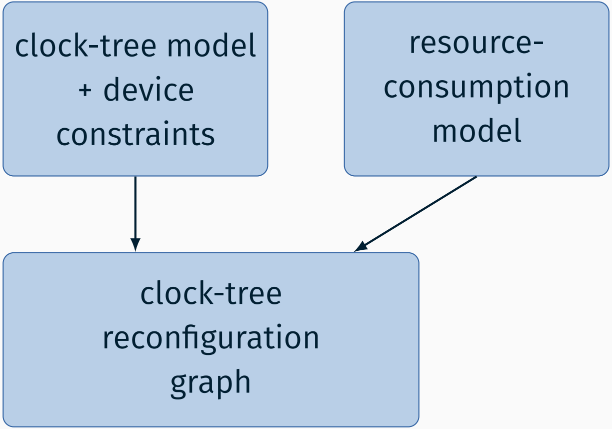
clock-tree model
+ device
constraints

Overview over the FusionClock Approach

clock-tree model + device constraints

resource-consumption model

Overview over the FusionClock Approach

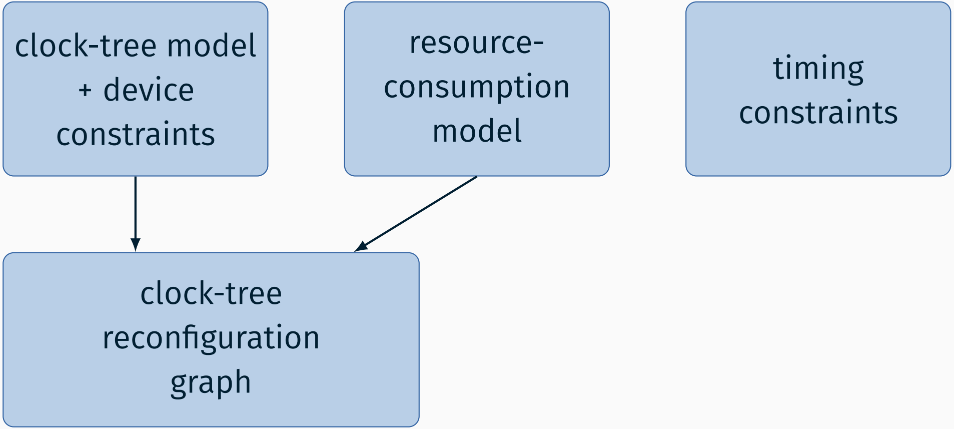


clock-tree model + device constraints

resource-consumption model

clock-tree reconfiguration graph

Overview over the FusionClock Approach

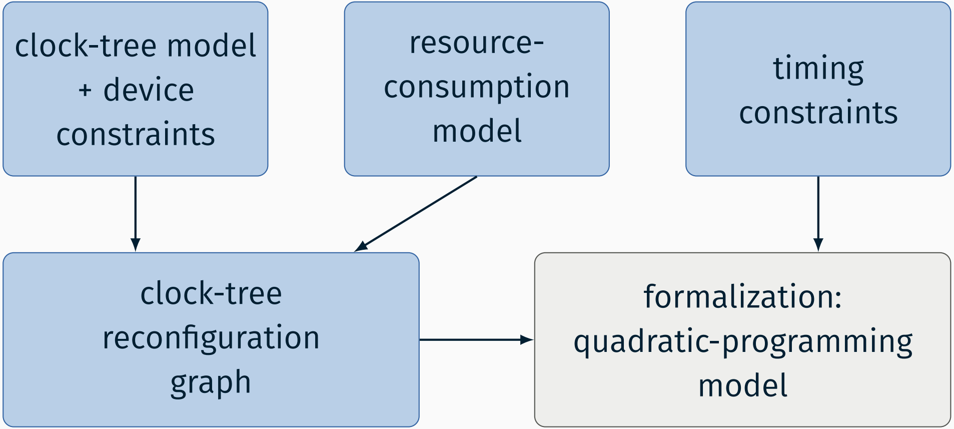

clock-tree model + device constraints

resource-consumption model

timing constraints

clock-tree reconfiguration graph

Overview over the FusionClock Approach


clock-tree model + device constraints

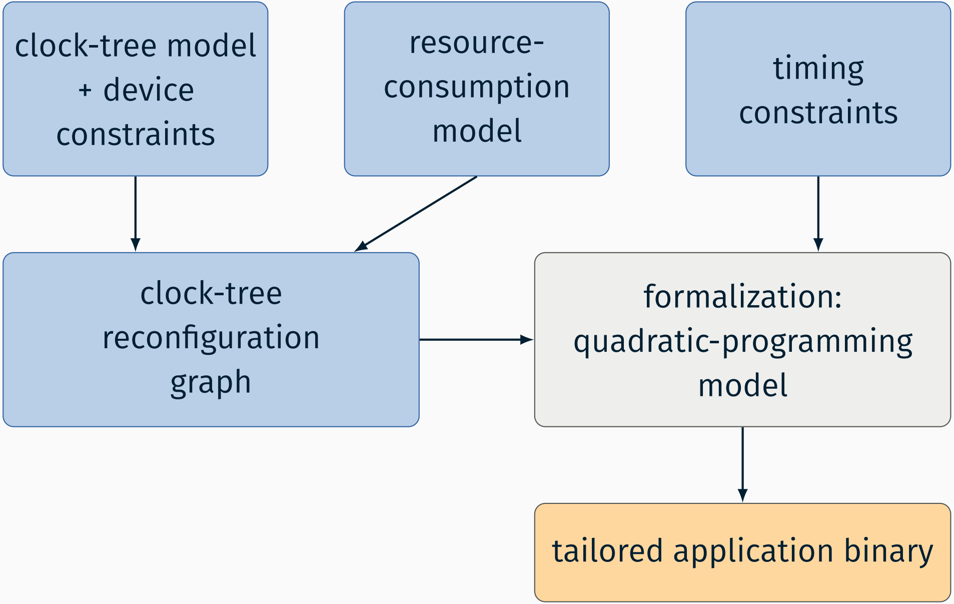
resource-consumption model

timing constraints

clock-tree reconfiguration graph

formalization: quadratic-programming model

Overview over the FusionClock Approach



clock-tree model + device constraints

resource-consumption model

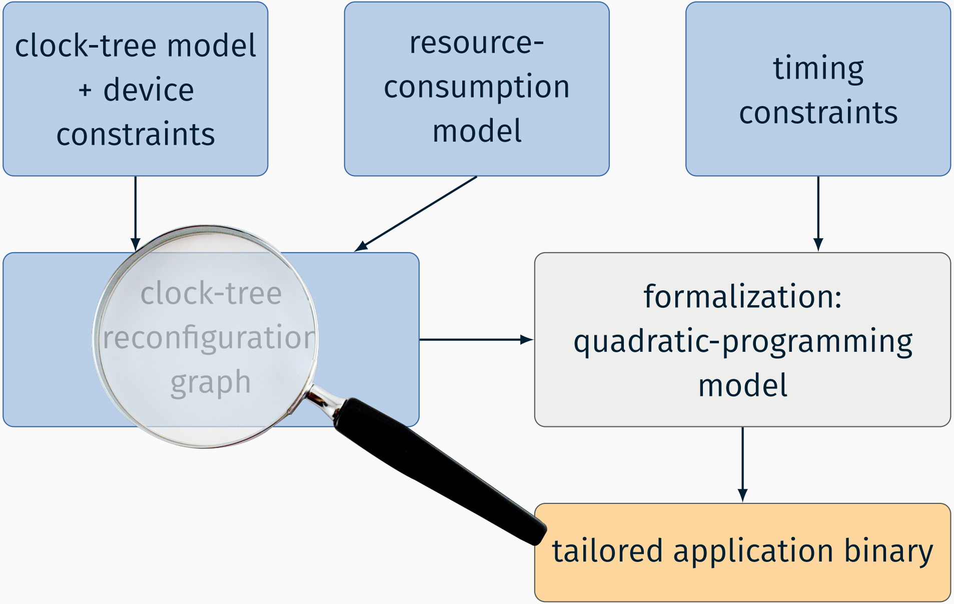
timing constraints

clock-tree reconfiguration graph

formalization: quadratic-programming model

tailored application binary

Overview over the FusionClock Approach



clock-tree model + device constraints

resource-consumption model

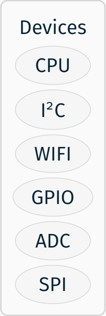
timing constraints

clock-tree reconfiguration graph

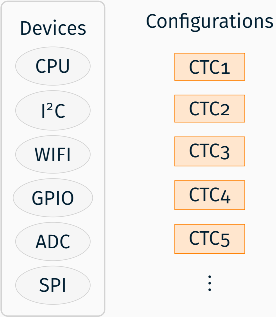
formalization: quadratic-programming model

tailored application binary

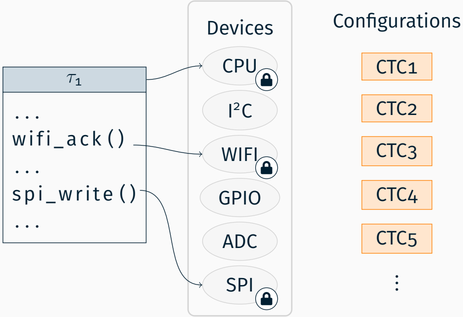
Clock-Tree Reconfiguration Graph



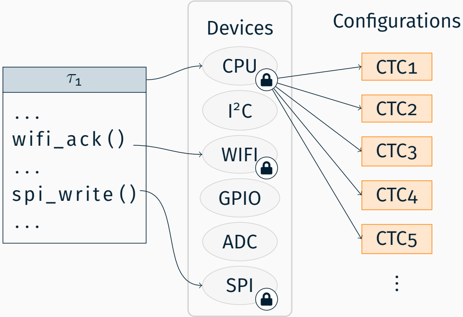
Clock-Tree Reconfiguration Graph



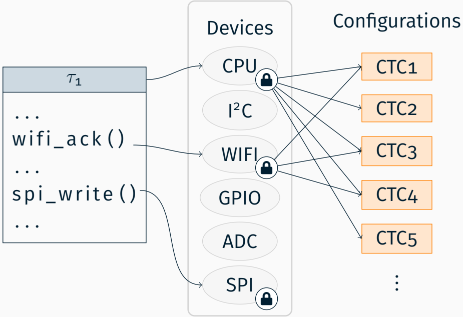
Clock-Tree Reconfiguration Graph



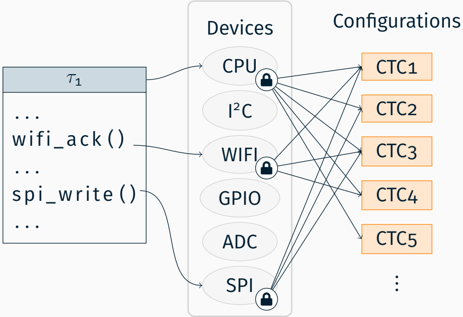
Clock-Tree Reconfiguration Graph



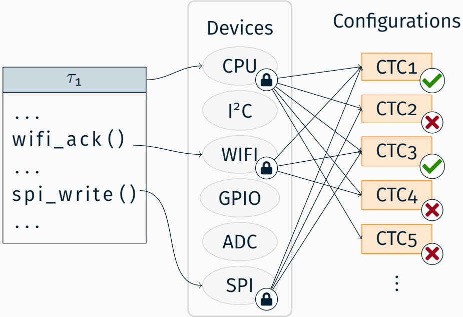
Clock-Tree Reconfiguration Graph



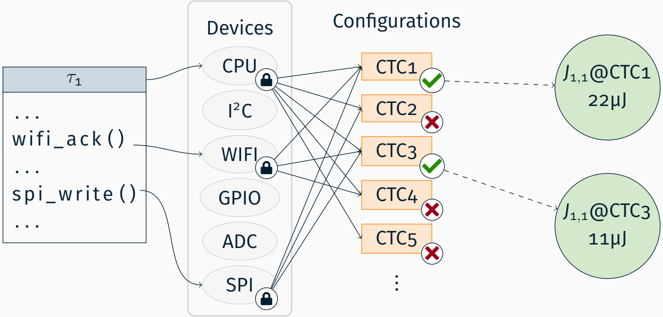
Clock-Tree Reconfiguration Graph



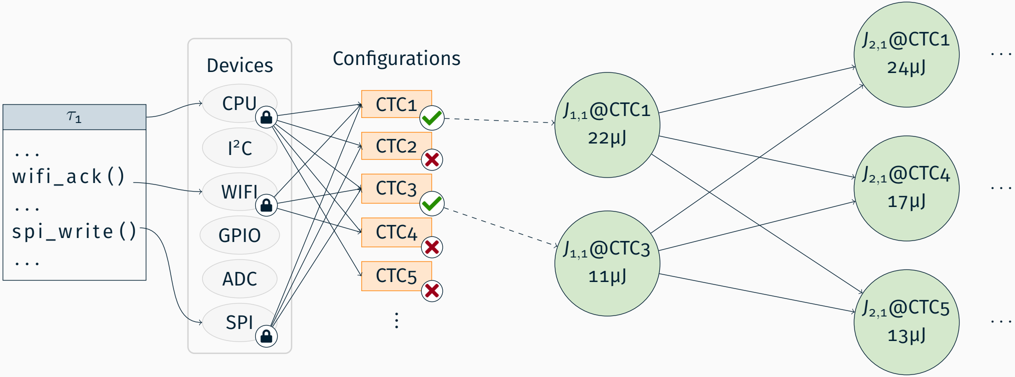
Clock-Tree Reconfiguration Graph



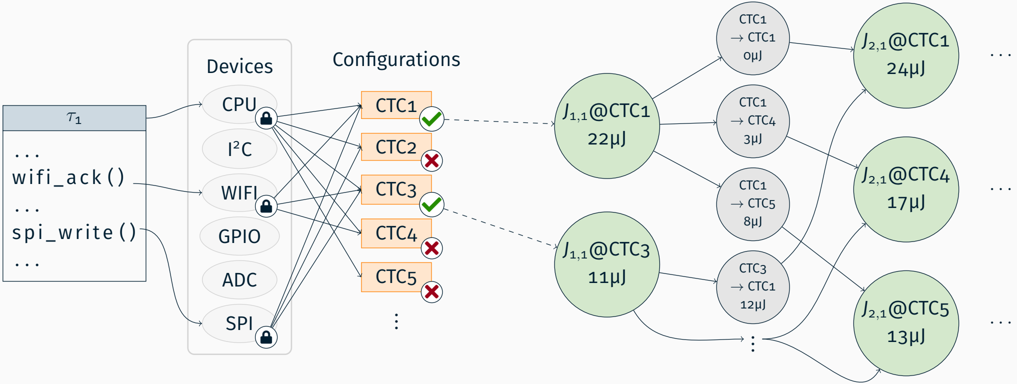
Clock-Tree Reconfiguration Graph



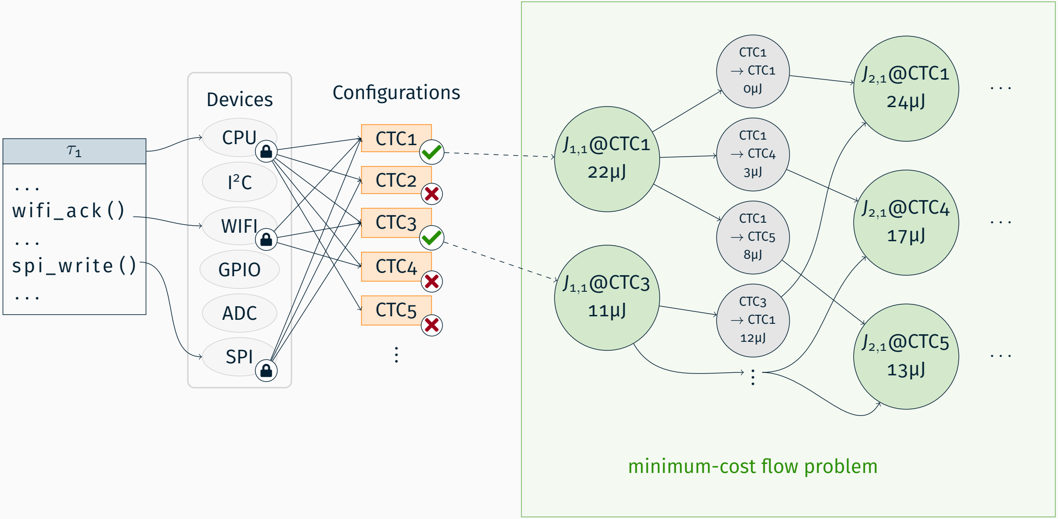
Clock-Tree Reconfiguration Graph



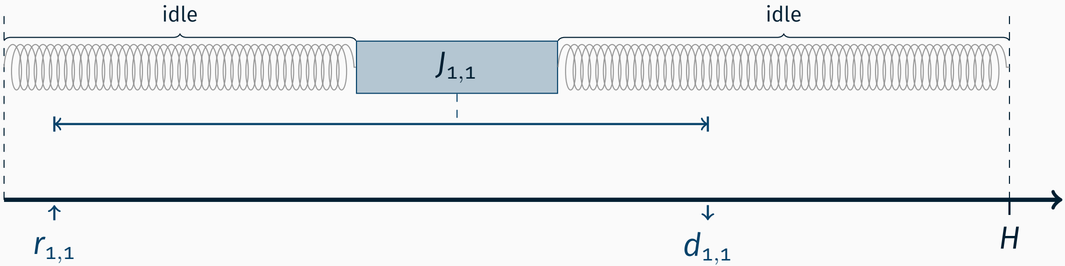
Clock-Tree Reconfiguration Graph



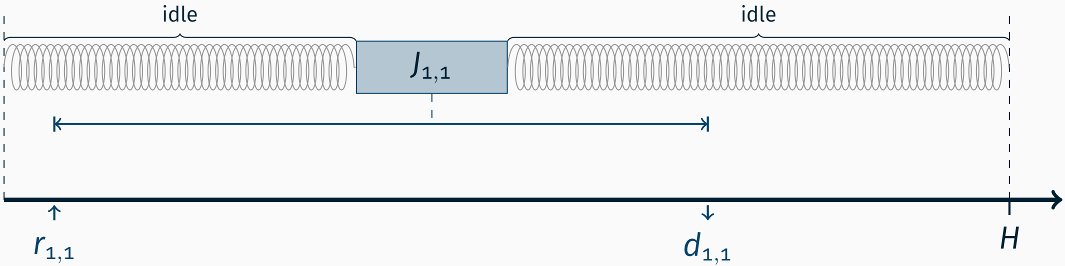
Clock-Tree Reconfiguration Graph



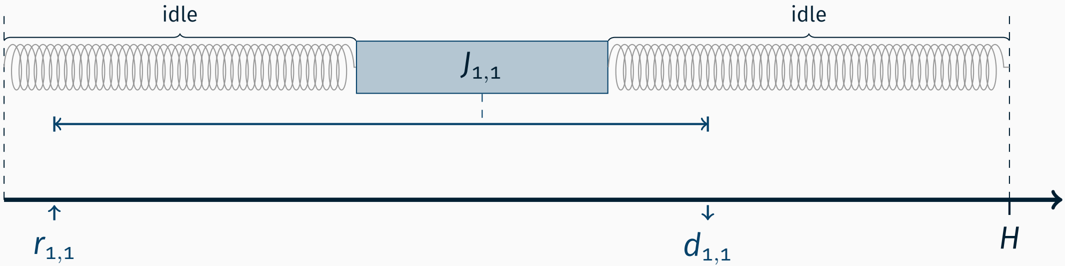
Incorporation of a Time-Triggered Schedule



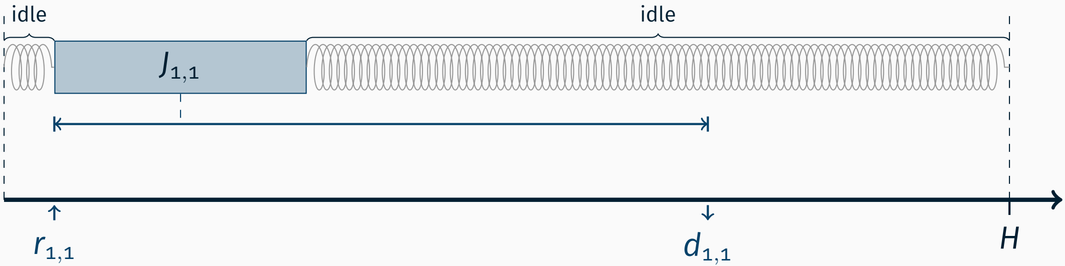
Incorporation of a Time-Triggered Schedule



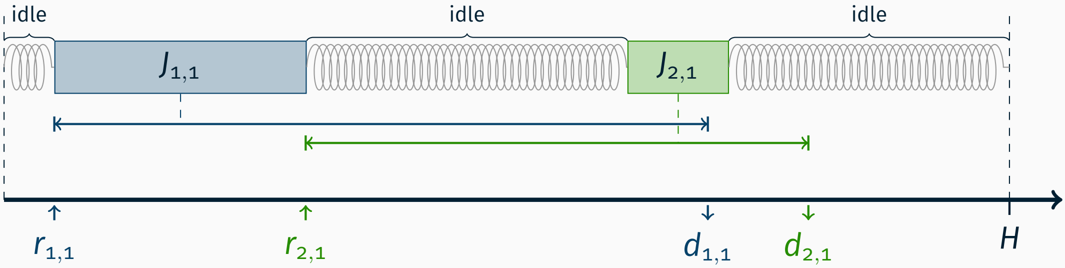
Incorporation of a Time-Triggered Schedule



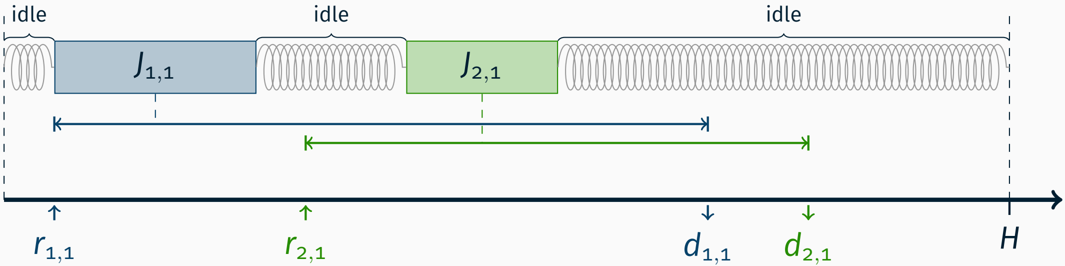
Incorporation of a Time-Triggered Schedule



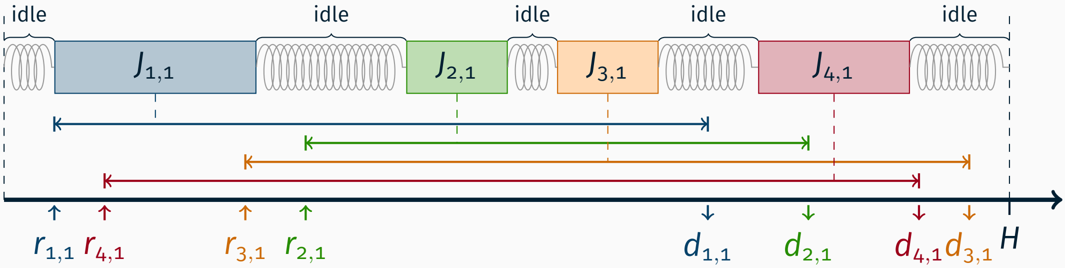
Incorporation of a Time-Triggered Schedule



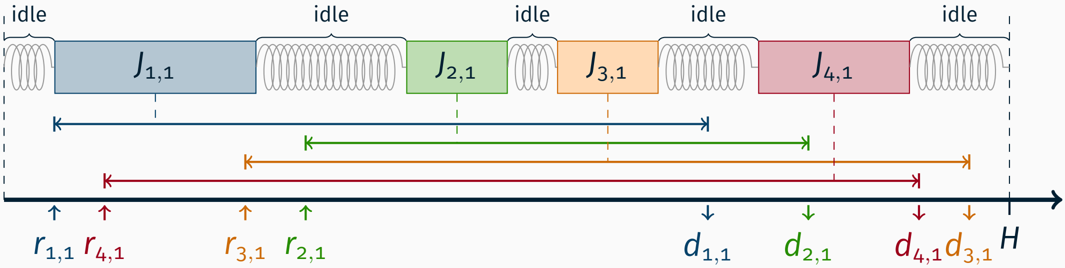
Incorporation of a Time-Triggered Schedule



Incorporation of a Time-Triggered Schedule



Incorporation of a Time-Triggered Schedule



Distributing the slack:

- **reconfiguration penalties**
- **idling**: start times, durations, and configurations

Formalization

min **energy costs** of jobs and idling options
+ **energy penalty** for reconfiguration

w.r.t.
constraints in the **clock-tree reconfiguration graph**
all times **sum up** to **hyperperiod**
each job **starts** at or after its **release time**
each job **finishes** before or at its **deadline**

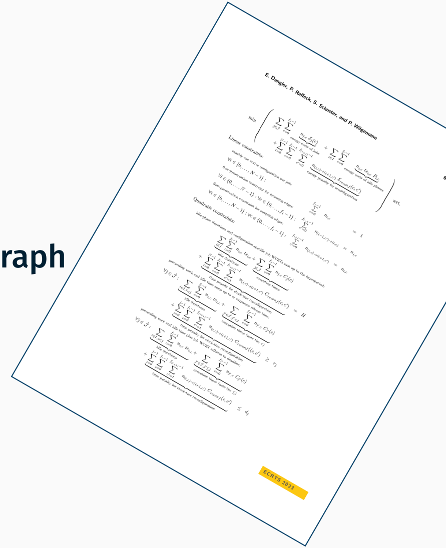
Formalization

min **energy costs** of jobs and idling options

+ **energy penalty** for reconfiguration

w.r.t.

- constraints in the **clock-tree reconfiguration graph**
- all times **sum up** to **hyperperiod**
- each job **starts** at or after its **release time**
- each job **finishes** before or at its **deadline**



E. Dengler, P. Raffeck, S. Schuster, and P. Wägemann

$$\min \left(\sum_{j\in\mathcal{J}} \sum_{c=0}^{f_j-1} \underbrace{n_{j,c}\,\mathcal{E}_j(c)}_{\text{energy costs of jobs}} + \sum_{i\in\mathcal{I}} \sum_{c=0}^{f_i-1} \underbrace{n_{i,c}\,ts_{i,c}\,P_{i,c}}_{\text{energy costs of idle phases}} + \sum_{i=0}^{N-1} \sum_{c=0}^{f_i-1} \sum_{c'=0}^{f_{(i+1)}-1} \underbrace{n_{(i,c)\to(i+1,c')}\,\mathcal{E}_{reconf}(c,c')}_{\text{energy penalty for reconfiguration}} \right) \text{wrt.}$$

Linear constraints:

exactly one active configuration per job:

$$\forall i \in \{0,\dots,N-1\}: \quad \sum_{c=0}^{f_i-1} n_{i,c} = 1$$

flow-preservation constraint for incoming edges:

$$\forall i \in \{0,\dots,N-1\}: \forall c \in \{0,\dots,f_i-1\}: \quad \sum_{c'=0}^{f_{i-1}-1} n_{(i-1,c')\to(i,c)} = n_{i,c}$$

flow-preservation constraint for outgoing edges:

$$\forall i \in \{0,\dots,N-1\}: \forall c \in \{0,\dots,f_i-1\}: \quad \sum_{c'=0}^{f_{i+1}-1} n_{(i,c)\to(i+1,c')} = n_{i,c}$$

Quadratic constraints:

idle-phase durations and configuration-specific job WCETs sum up to the hyperperiod:

$$\underbrace{\sum_{i\in\mathcal{I}} \sum_{c=0}^{f_i-1} n_{i,c}\,ts_{i,c}}_{\text{idle durations}} + \underbrace{\sum_{j\in\mathcal{J}} \sum_{c=0}^{f_j-1} n_{j,c}\,C_j(c)}_{\text{execution times}} + \underbrace{\sum_{i=0}^{N-1} \sum_{c=0}^{f_i-1} \sum_{c'=1}^{f_{(i+1)}-1} n_{(i,c)\to(i+1,c')}\,C_{reconf}(c,c')}_{\text{time penalty for clock-tree reconfiguration}} = H$$

preceding work and idle time sums up to or surpasses release time:

$$\forall j \in \mathcal{J}: \quad \underbrace{\sum_{i\in\mathcal{I},i<j} \sum_{c=0}^{f_i-1} n_{i,c}\,ts_{i,c}}_{\text{idle durations}} + \underbrace{\sum_{j'\in\mathcal{J},j'<j} \sum_{c=0}^{f_{j'}-1} n_{j',c}\,C_{j'}(c)}_{\text{execution times (note the <)}} + \underbrace{\sum_{i=0}^{j-1} \sum_{c=0}^{f_i-1} \sum_{c'=1}^{f_{(i+1)}-1} n_{(i,c)\to(i+1,c')}\,C_{reconf}((c,c')}_{\text{time penalty for clock-tree reconfiguration}} \geq r_j$$

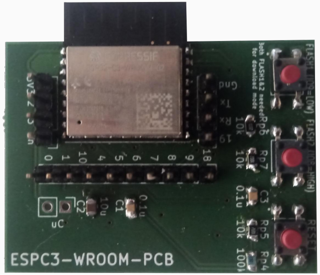
preceding work and idle time plus job WCET adheres to deadline:

$$\forall j \in \mathcal{J}: \quad \underbrace{\sum_{i\in\mathcal{I},i<j} \sum_{c=0}^{f_i-1} n_{i,c}\,ts_{i,c}}_{\text{idle durations}} + \underbrace{\sum_{j'\in\mathcal{J},j'\leq j} \sum_{c=0}^{f_{j'}-1} n_{j',c}\,C_{j'}(c)}_{\text{execution times (note the}\leq)} + \underbrace{\sum_{i=0}^{j-1} \sum_{c=0}^{f_i-1} \sum_{c'=1}^{f_{(i+1)}-1} n_{(i,c)\to(i+1,c')}\,C_{reconf}(c,c')}_{\text{time penalty for clock-tree reconfiguration}} \leq d_j$$

ECRTS 2023

Evaluation

Evaluation Hardware



Evaluation: Break-Even Point Analysis

```
while(true) {
    fibonacci_calculation();
    idle(until=hyperperiod.end);
}
```

Evaluation: Break-Even Point Analysis

```
while(true) {
    fibonacci_calculation();
    idle(until=hyperperiod.end);
}
```

- 5 CTCs

Evaluation: Break-Even Point Analysis

```
while(true) {
    fibonacci_calculation();
    idle(until=hyperperiod.end);
}
```

- 5 CTCs: which one is used for the compute task?

Evaluation: Break-Even Point Analysis

```
while(true) {
    fibonacci_calculation();
    idle(until=hyperperiod.end);
}
```

- 5 CTCs: which one is used for the compute task?
- 3 idle options

Evaluation: Break-Even Point Analysis

```
while(true) {
    fibonacci_calculation();
    idle(until=hyperperiod.end);
}
```

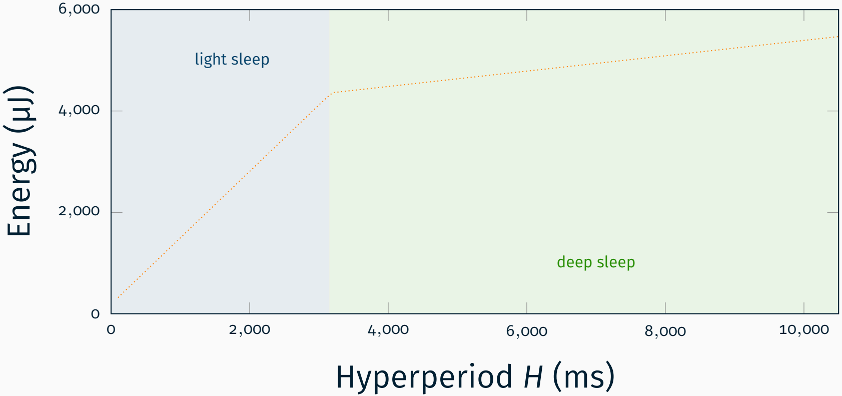
- 5 CTCs: which one is used for the compute task?
- 3 idle options: when do the energy savings outweigh the reconfiguration penalties?

Evaluation: Break-Even Point Analysis

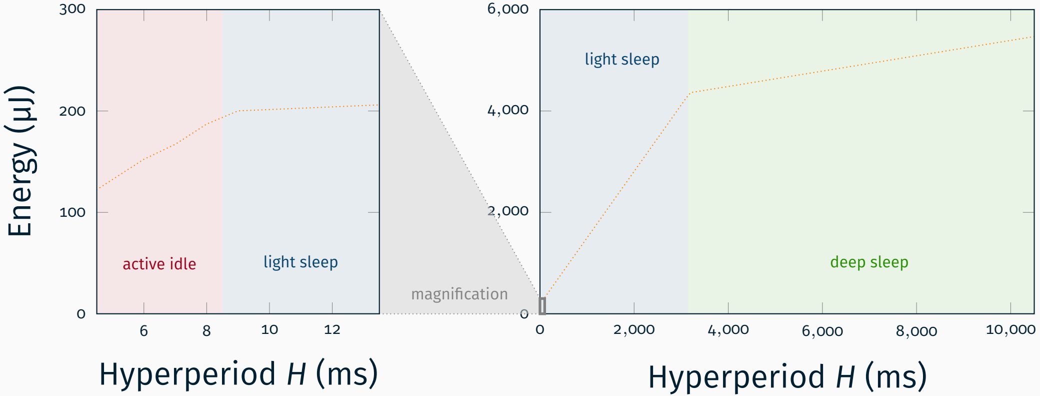
```
while(true) {
    fibonacci_calculation();
    idle(until=hyperperiod.end);
}
```

- 5 CTCs: which one is used for the compute task?
- 3 idle options: when do the energy savings outweigh the reconfiguration penalties?
- how do actual measurements compare to the predicted energy consumptions?

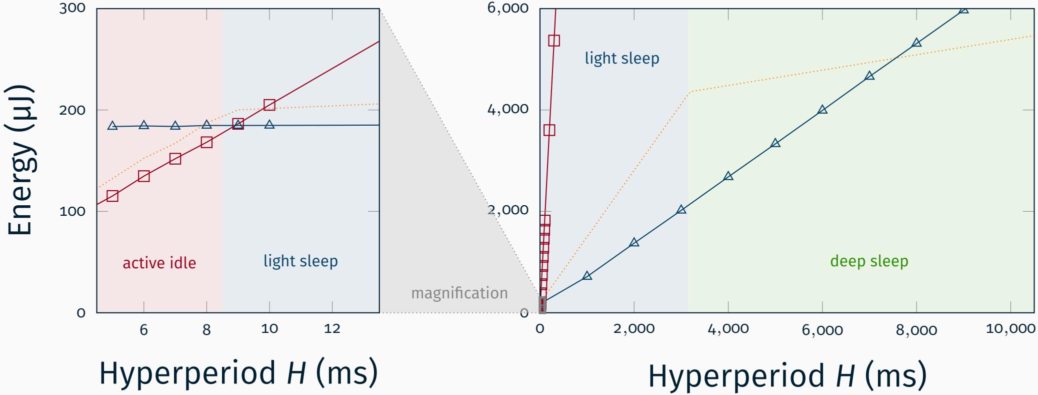
Evaluation: Break-Even Point Analysis



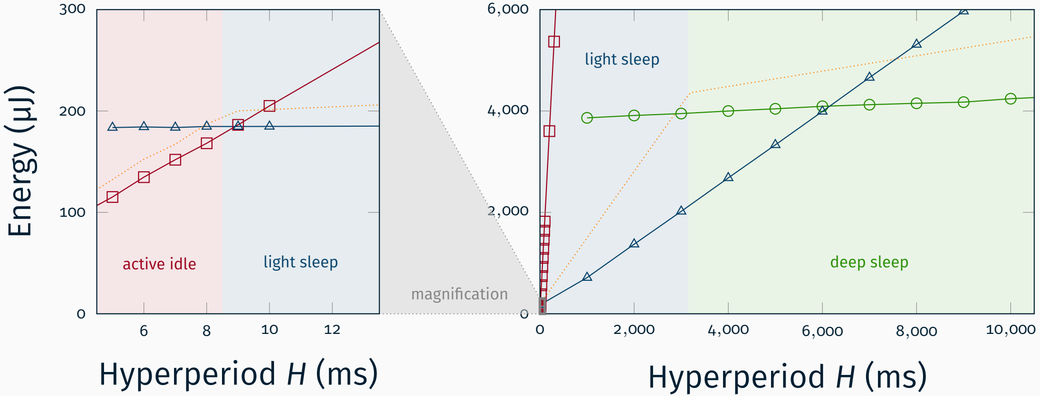
Evaluation: Break-Even Point Analysis



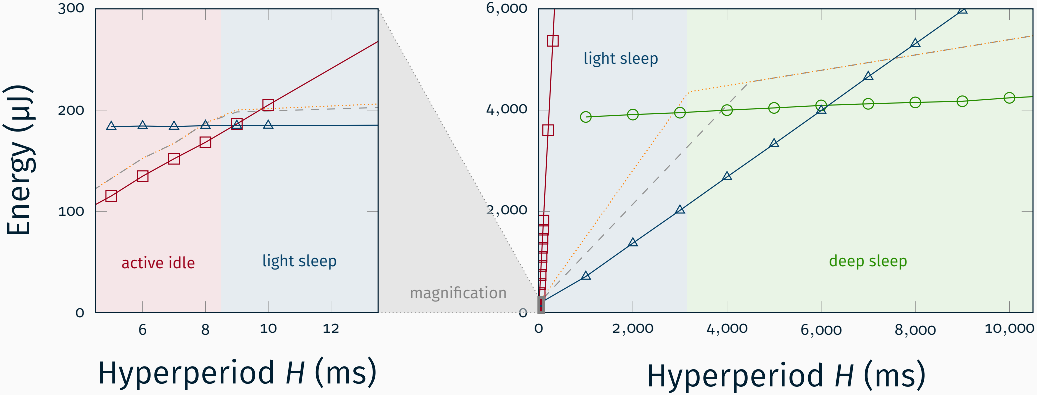
Evaluation: Break-Even Point Analysis



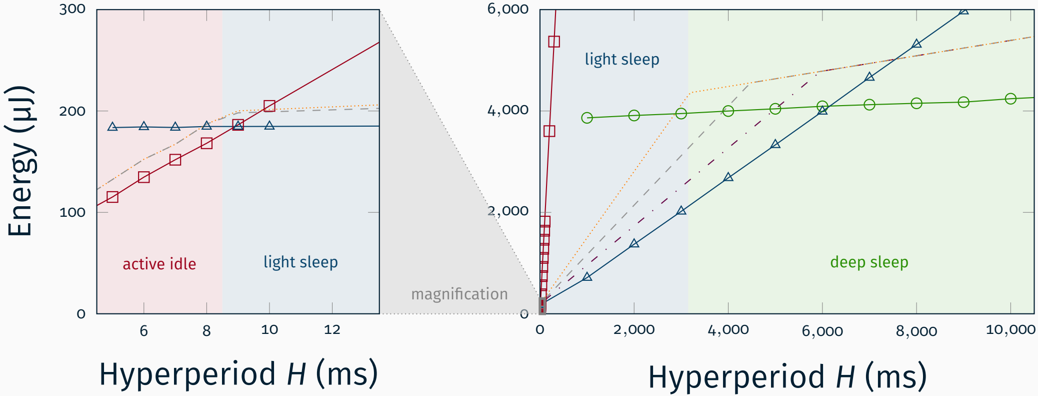
Evaluation: Break-Even Point Analysis



Evaluation: Break-Even Point Analysis



Evaluation: Break-Even Point Analysis



Evaluation: Taskset Generation

Does FusionClock ...

- ... determine a reliable upper bound?
- ... minimze energy consumption in comparison to device-unselective approaches?

Evaluation: Taskset Generation

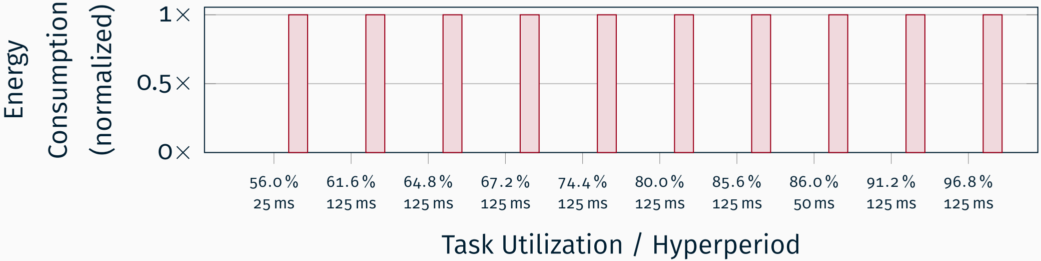
Does FusionClock …

- … determine a reliable upper bound?
- … minimze energy consumption in comparison to device-unselective approaches?

Evaluation with generated tasksets:

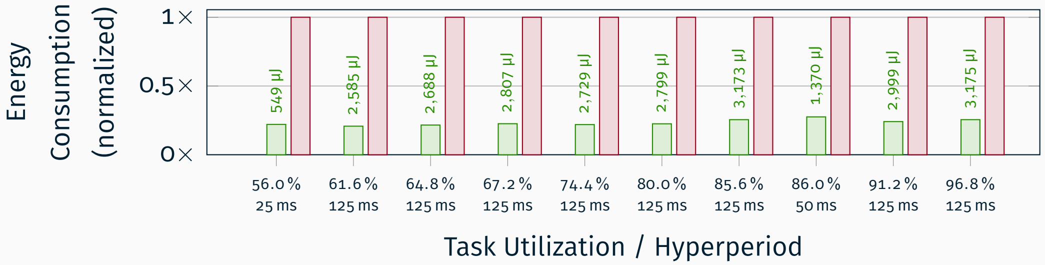
- simulate device usage: sense, compute, actuate
- 5 active modes
- 2 idle modes: light sleep, deep sleep
- 9 to 18 tasks

Evaluation: Taskset Generation



binary without clock-tree reconfigurations

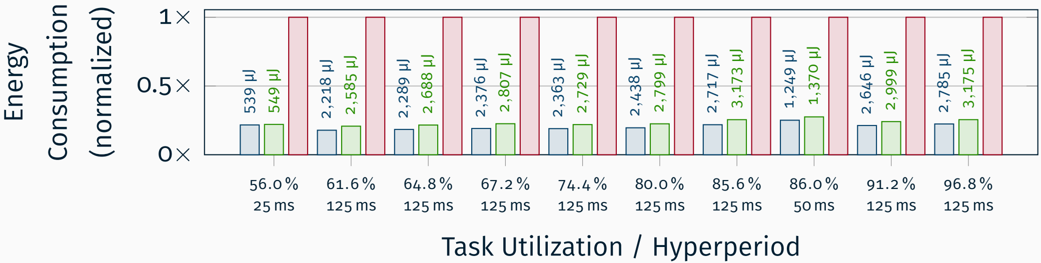
Evaluation: Taskset Generation



predicted energy consumption

binary without clock-tree reconfigurations

Evaluation: Taskset Generation

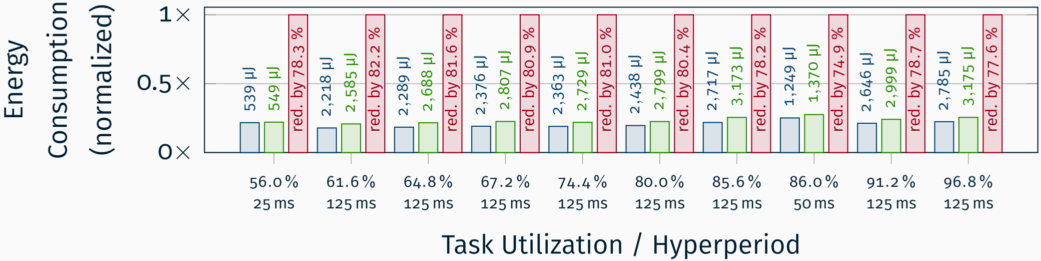


tailored application binary

predicted energy consumption

binary without clock-tree reconfigurations

Evaluation: Taskset Generation



tailored application binary

predicted energy consumption

binary without clock-tree reconfigurations

Problems solved by FusionClock

1. CPU-only approaches...
   - neglect energy consumption of devices
   - ignore dependencies of devices and clock-tree configurations
2. no guarantees of feedback-based approaches
3. missing reconfiguration penalties

Problems solved by FusionClock

1. CPU-only approaches...
   - neglect energy consumption of devices
   - ignore dependencies of devices and clock-tree configurations

   ✓ **device-aware model, making use of the system's clock tree**
2. no guarantees of feedback-based approaches
3. missing reconfiguration penalties

Problems solved by FusionClock

1. CPU-only approaches...
    - neglect energy consumption of devices
    - ignore dependencies of devices and clock-tree configurations

    ✓ **device-aware model, making use of the system’s clock tree**
2. no guarantees of feedback-based approaches

    ✓ **resource consumption guarantees due to static approach**
3. missing reconfiguration penalties

Problems solved by FusionClock

1. CPU-only approaches...
   - neglect energy consumption of devices
   - ignore dependencies of devices and clock-tree configurations

   ✓ **device-aware model, making use of the system's clock tree**
2. no guarantees of feedback-based approaches

   ✓ **resource consumption guarantees due to static approach**
3. missing reconfiguration penalties

   ✓ **inclusion of clock-tree reconfiguration costs in optimization**

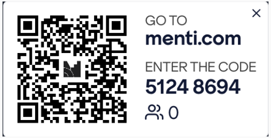
Questions?

Source Code and Artifact Evaluation of FusionClock

`https://gitlab.cs.fau.de/fusionclock`

E. Dengler, P. Raffeck, S. Schuster, and P. Wägemann. **FusionClock: WCEC-Optimal Clock-Tree Reconfigurations (Artifact).** *Dagstuhl Artifacts Series*, 9(1):2:1–2:3, 2023.

ECRTS * Artifact * AE * Evaluated *
Consistent * Complete * Well Documented * Easy to Reuse *



GO TO
menti.com
ENTER THE CODE
5124 8694
0