### Introduction

**WCET Analysis**
- Timing-critical applications
- Strict deadlines on the response times
- WCET bound of a program
- Upper bound on the execution times
- Depends on behavior at microarch. level
- Obtained by WCET analysis
- Modern processors are complex
- Designed for average-case performance
- Too many behaviors at microarch. level
- Exhaustive simulation not possible
- Need for approximation
- Hide some microarchitectural details

**Multi-Core Processors**
- Resources shared between the cores
- Data buses
- Caches
- Advantages
  - Reduced weight
  - Reduced energy consumption
  - Reduced production costs
- Shared resource interference
  - Performance can drop
- Challenge for WCET analysis
  - Consider all access interleavings
  - Need for approximation

**Existing Approaches**
- Limited to particular...
- Classes of processors
- Ways of approximation
- Formalisms and algorithms differ
- Yet, they follow a common methodology
- Coarse approximation as baseline
- Exclude some spurious behavior

**Our Contribution**
- A meta approach
- Formalizes common methodology

---

### Meta Approach

- For simplicity
  - One program per core
  - Each program is run at most once
- Concrete execution behavior of given system
  \[ \text{Traces} \subseteq \text{Universe} \]
- WCET of program on core \( C \)
  \[ WCET_C = \max_{t \in \text{Traces}} et_C(t) \]
- Approximation by abstract traces
  - Abstract model \( (\text{Traces}, \gamma_{\text{trunc}}) \)
  - Overapproximates concrete behavior
    \[ \bigcup_{t \in \text{Traces}} \gamma_{\text{trace}}(\hat{t}) \supseteq \text{Traces} \]
- Provides WCET bound
  \[ \max_{t \in \text{Traces}} \text{UB}et_C(\hat{t}) \geq WCET_C \]
- Infeasible abstract traces
  \[ \text{Infeas} = \{ \hat{t} | \hat{t} \in \text{Traces} \land \gamma_{\text{trace}}(\hat{t}) \cap \text{Traces} = \emptyset \} \]
  - Only describe spurious behavior
  - May dominate WCET bound calculation
- System properties
  \[ \text{Prop} = \{ P_1, \ldots, P_{\#\text{Prop}} \} \]
  - Hold for each system behavior
    \[ \forall t \in \text{Traces} : \forall P_i \in \text{Prop} : P_i(t) \]
- Lifting properties to abstract traces
- Soundness criterion for lifted version \( \hat{P}_i \) of \( P_i \)
  \[ \exists t \in \gamma_{\text{trace}}(\hat{t}) : P_i(t) \Rightarrow \hat{P}_i(\hat{t}) \]
- Detecting infeasible abstract trace \( \hat{t} \)
  \[ \exists P_i \in \text{Prop} : \neg \hat{P}_i(\hat{t}) \Rightarrow \hat{t} \in \text{Infeas} \]
- Remove some infeasible abstract traces
  \[ \text{LessTraces} = \{ \hat{t} | \hat{t} \in \text{Traces} \land \forall P_i \in \text{Prop} : \hat{P}_i(\hat{t}) \} \]
- Potentially improved WCET bound
  \[ \max_{t \in \text{Traces}} \text{UB}et_C(\hat{t}) \geq \max_{t \in \text{LessTraces}} \text{UB}et_C(\hat{t}) \geq WCET_C \]

---

### Property Lifting Example

- Round-robin bus arbitration
  - Before a requested bus access is granted, at most one bus access per concurrent core is granted.
  \[ \leq \#\text{Cores} - 1 \]
  \[ \text{Concurrent accesses granted} \]

- Blocked cycles for a single bus access
  - No access takes longer than \( l_a \) cycles
  \[ \leq \#\text{Cores} - 1 \]

- A round-robin property
  - Shall hold for all \( t \in \text{Traces} \)
    \[ P_{rr}(t) \iff [\# \text{blockedCycles}_C(t)] \leq \#\text{accesses}_C(t) \cdot (\#\text{Cores} - 1) \cdot l_a \]
  - Lifting \( P_{rr} \) to abstract traces
    \[ \exists t \in \gamma_{\text{trace}}(\hat{t}) : P_{rr}(t) \]
    \[ \exists t \in \gamma_{\text{trace}}(\hat{t}) : \# \text{blockedCycles}_C(t) \leq \#\text{accesses}_C(t) \cdot (\#\text{Cores} - 1) \cdot l_a \]
    \[ \Rightarrow \text{UB} \# \text{blockedCycles}_C(\hat{t}) \leq \text{UB} \#\text{accesses}_C(\hat{t}) \cdot (\#\text{Cores} - 1) \cdot l_a \]
    \[ \Rightarrow: P_{rr}(\hat{t}) \]

### Acknowledgement

AVACS

http://www.uni-saarland.de  http://www.avacs.org