Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

EXCELENCIA
SEVERO
OCHOA

Towards Limiting the Impact of Timing Anomalies in Complex Real-Time Processors

Pedro Benedicte, Jaume Abella,
Carles Hernandez, Enrico Mezzetti,
Francisco J. Cazorla

**Asia and South Pacific
Design Automation Conference**

January 22nd
Tokyo, Japan

Outline

- Introduction to
  - Real-time systems
  - Timing analysis techniques
  - Timing Anomalies (TA)
- TA in STA
  - Studied in the literature
- TA in MBTA
  - **Not yet defined in the literature**
  - **Our work is the first describing TA for MBTA**
- TA in MBPTA
  - Taxonomy
  - Approach to handle TA
  - Case study: multicore processor
  - Experimentation to show the impact of TA

Critical Real Time Systems

- Used in industries like:
  - Avionics
  - Automotive
  - Railway
- Require:
  - Functional correctness
  - Timing correctness
- Validation and verification is needed for both:
  - Avionics: DO178B/C
  - Automotive: ISO26262
- We focus on Timing Correctness

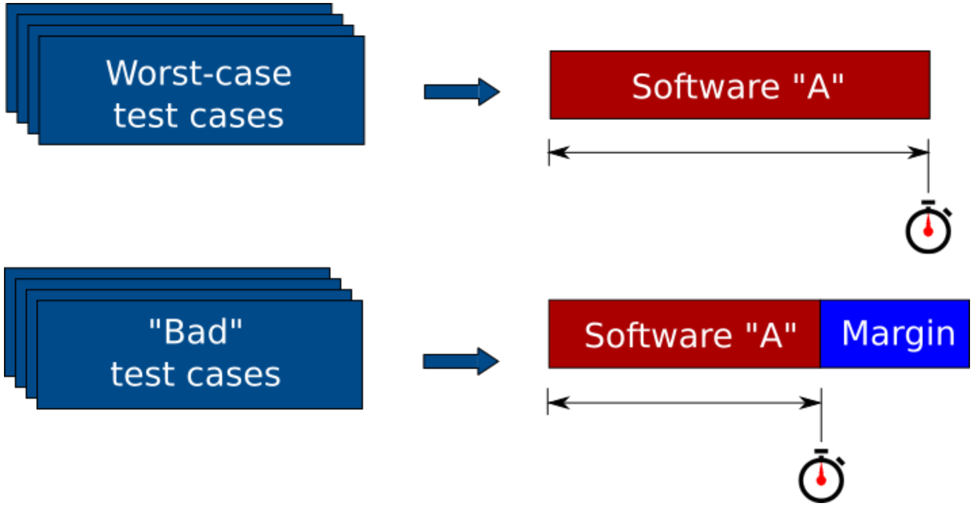
Static Timing Analysis (STA)

- Develop an **abstract model** of the system
  - For instance mathematical model
  - Analyse the model
- Sometimes **not feasible** to make an accurate model
  - Manufacturers do not disclose the implementation
  - Manufacturers sometimes do not know (in terms of timing)

- **Model all possible inputs** and hardware **states**
- Since this is highly challenging due to the high number of possible states, abstraction is used
  - States that do not lead to the worst-case behaviour are discarded
  - **Pessimism is increased** but **complexity is reduced**

Measurement-Based Timing Analysis (MBTA)

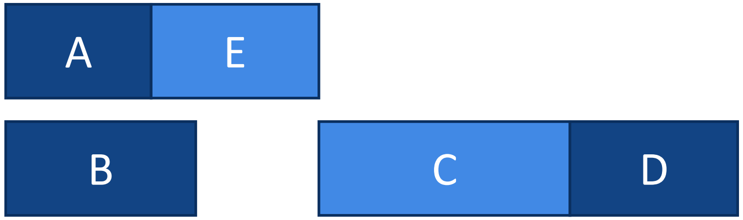
- Most used technique in the industry
- Analysis Time (AT)
  - Execute a number of analysis (stressing) tests/runs
  - Get measurements (under stressful conditions)
  - WCET = HWM + margin (to cover the unknowns)
- Representativeness challenge
  - Do the tests cover the worst conditions that can arise at operation time?



Timing Anomalies (TA)

- Definition:
  - Local worse-case does not lead to the global worst-case
- Example:
  - A-E, C-B-D are data dependent
  - C & E use the same resource

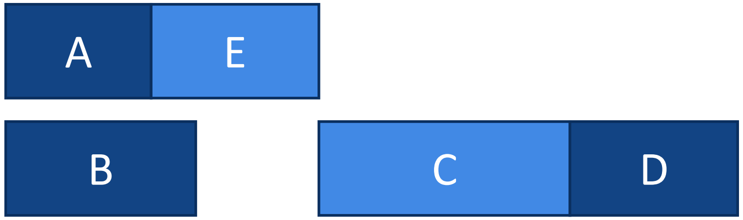
Case 1



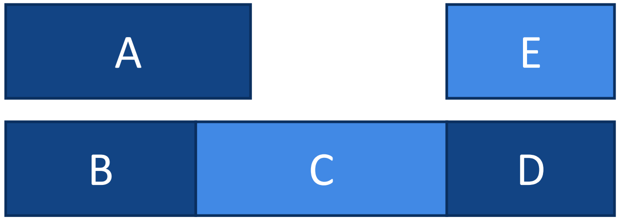
Timing Anomalies (TA)

- Definition:
  - Local worse-case does not lead to the global worst-case
- Example:
  - A-E, C-B-D are data dependent
  - C & E use the same resource

Case 1

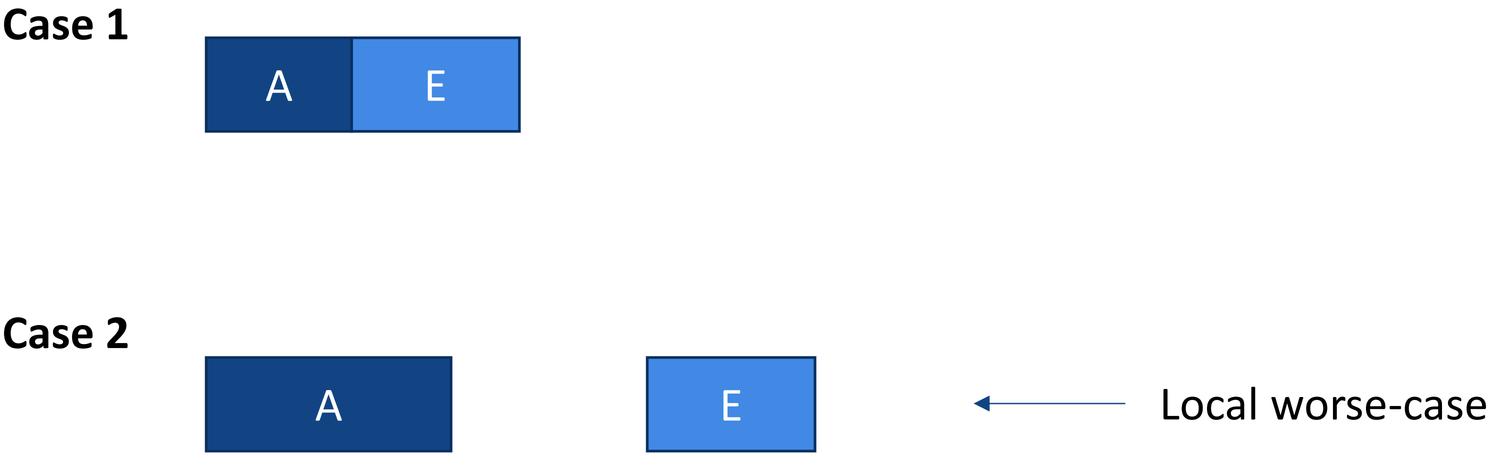


Case 2



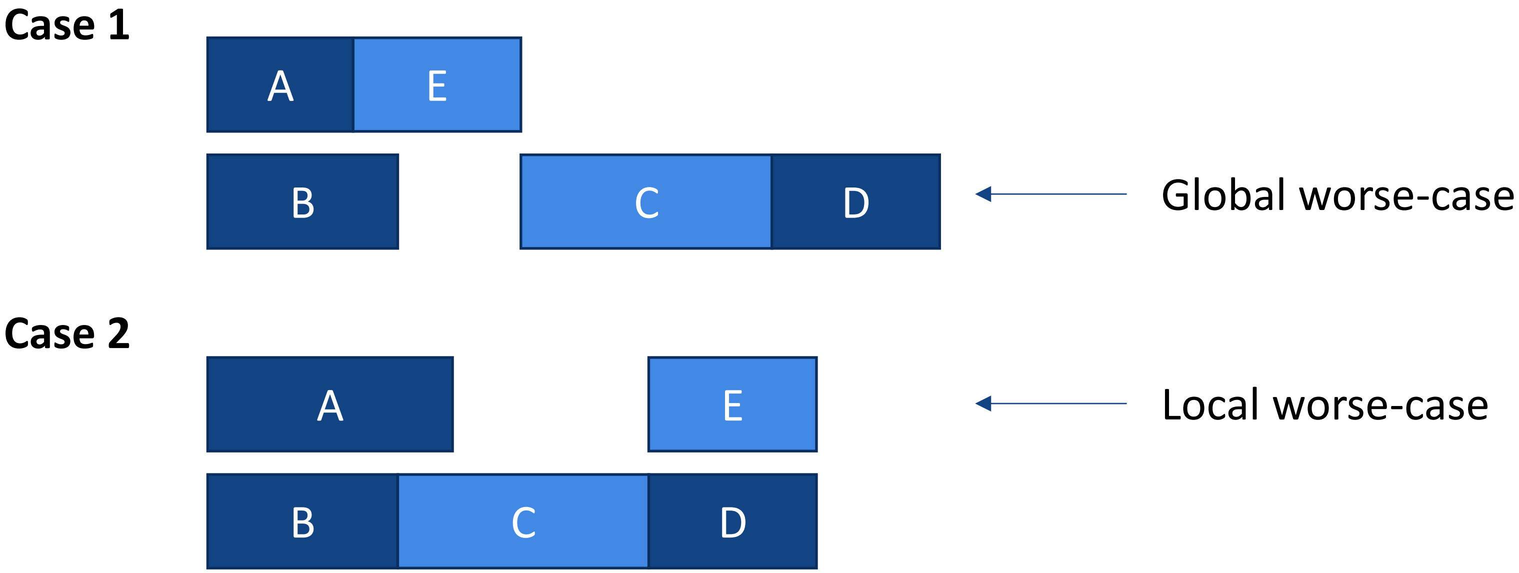
Timing Anomalies (TA)

- Definition:
  - Local worse-case does not lead to the global worst-case
- Example:
  - A-E, C-B-D are data dependent
  - C & E use the same resource



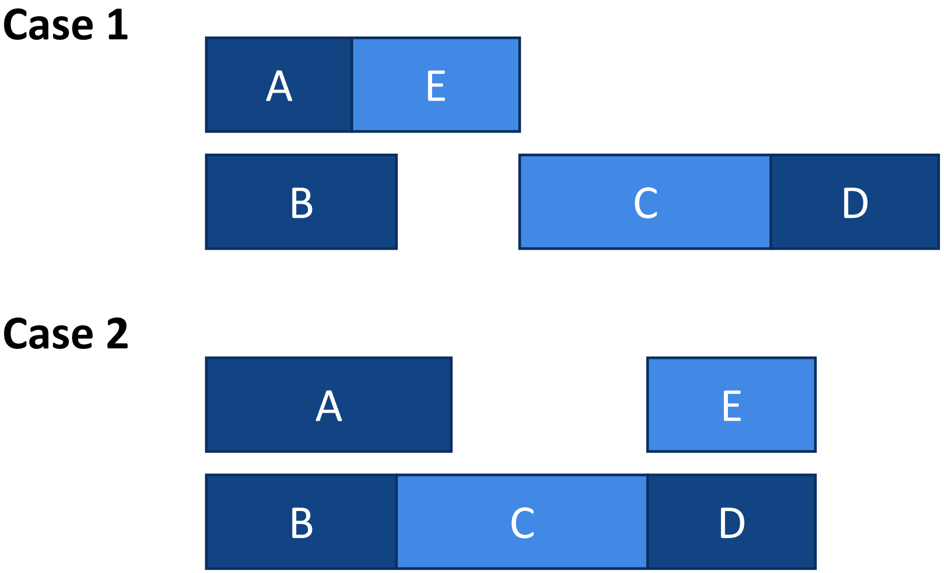
Timing Anomalies (TA)

- Definition:
  - Local worse-case does not lead to the global worst-case
- Example:
  - A-E, C-B-D are data dependent
  - C & E use the same resource



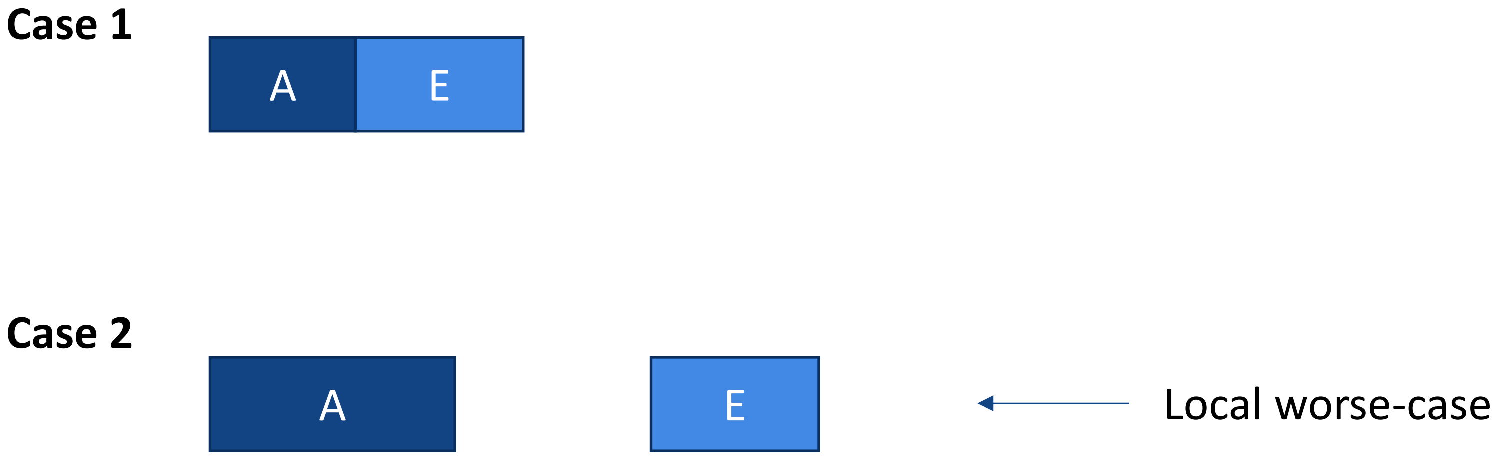
SoA: Timing anomalies in STA

- Timing anomalies jeopardize STA
- States that will not lead to the worst-case behaviour (**assuming no TA**) are discarded
- Assumption: timing can be analysed at the level of single execution blocks



SoA: Timing anomalies in STA

- Timing anomalies jeopardize STA
- States that will not lead to the worst-case behaviour (**assuming no TA**) are discarded
- Assumption: timing can be analysed at the level of single execution blocks



Case 1

A E

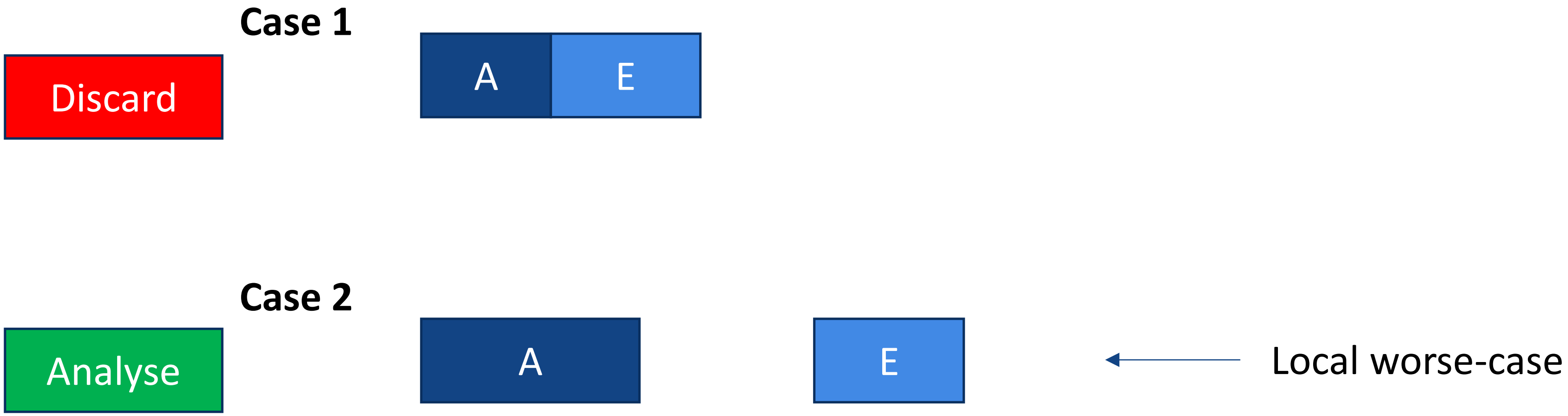
Case 2

A E

Local worse-case

SoA: Timing anomalies in STA

- Timing anomalies jeopardize STA
- States that will not lead to the worst-case behaviour (**assuming no TA**) are discarded
- Assumption: timing can be analysed at the level of single execution blocks



Case 1

Discard

A E

Case 2

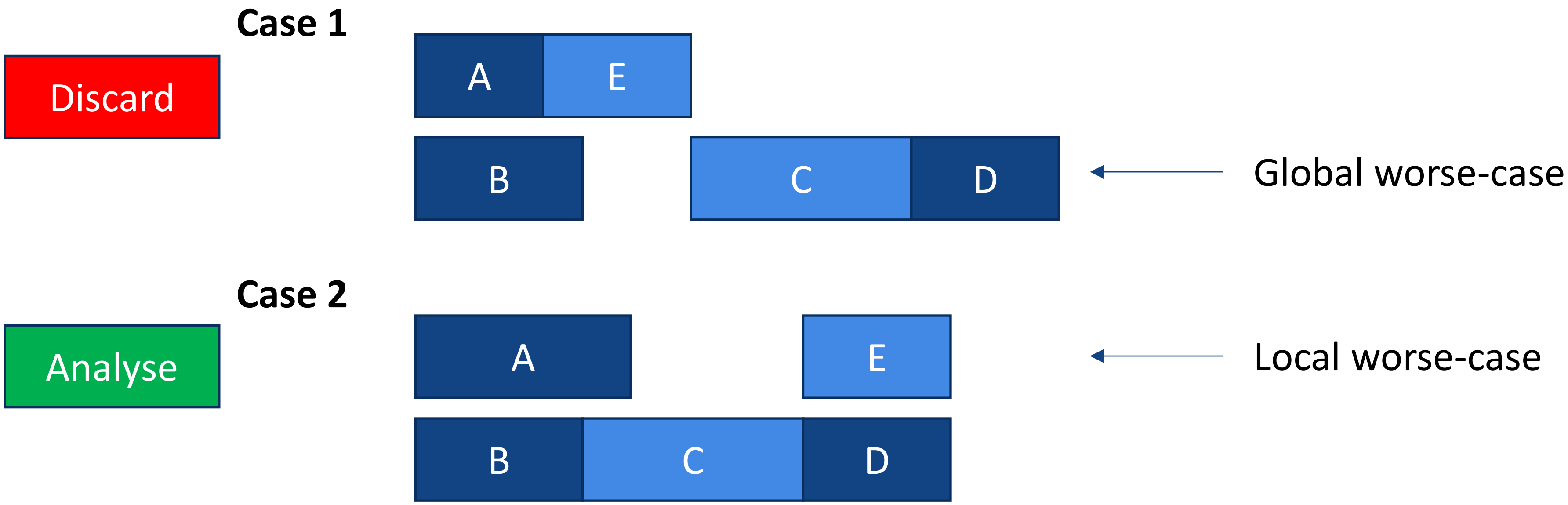
Analyse

A E

Local worse-case

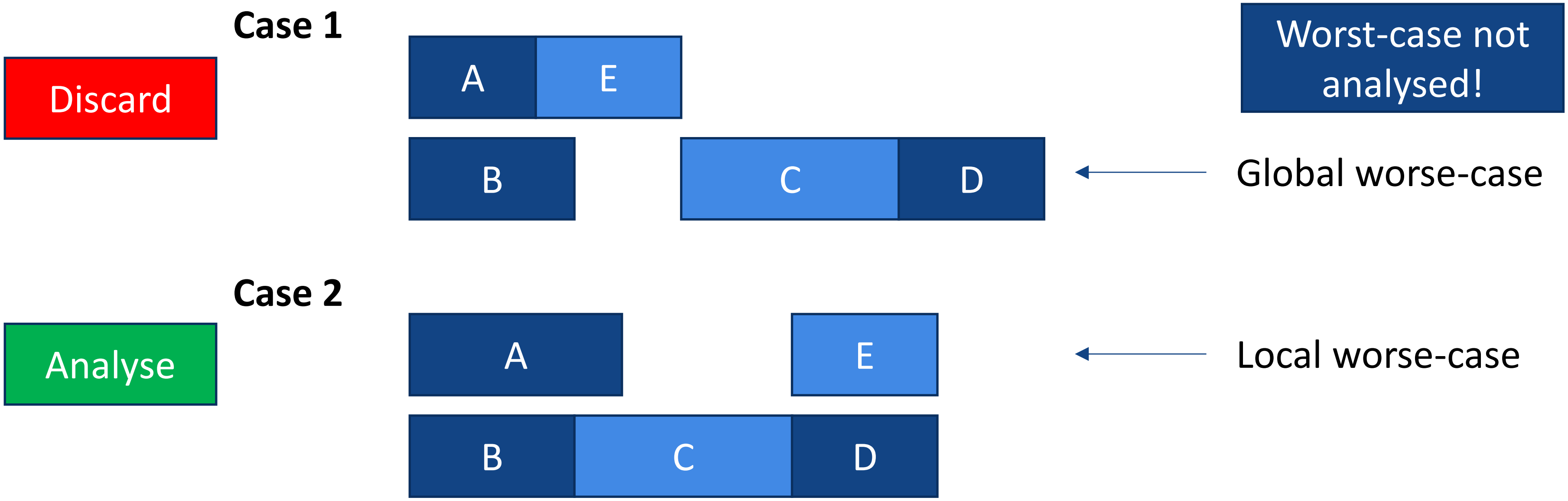
SoA: Timing anomalies in STA

- Timing anomalies jeopardize STA
- States that will not lead to the worst-case behaviour (**assuming no TA**) are discarded
- Assumption: timing can be analysed at the level of single execution blocks



SoA: Timing anomalies in STA

- Timing anomalies jeopardize STA
- States that will not lead to the worst-case behaviour (**assuming no TA**) are discarded
- Assumption: timing can be analysed at the level of single execution blocks



Case 1

Discard

Worst-case not analysed!

Global worse-case

Case 2

Analyse

Local worse-case

SoA: Timing anomalies in STA

- Assess whether there are TA or not
- If there are TA, do not discard locally-good scenarios

- Taxonomy of TA in STA:
  - Bounded
    - Add constant, pessimistic time to the WCET
  - Unbounded
    - Cannot be quantified because of domino effect
    - Would require to analyze a huge number of possibilities → unfeasible

Contribution 1:
Timing anomalies in MBTA

Asia and South Pacific
Design Automation Conference

January 22nd
Tokyo, Japan

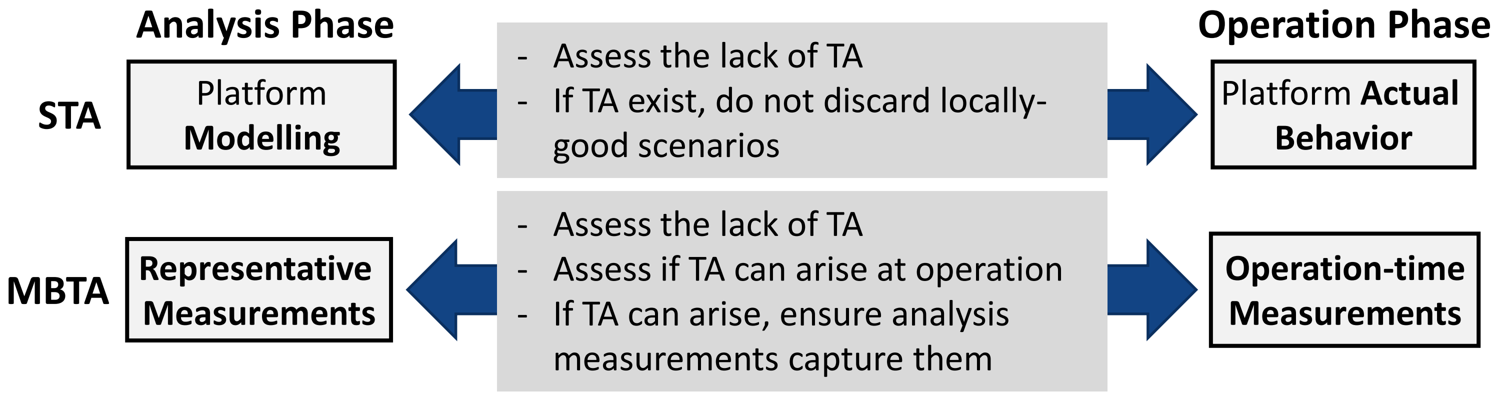
Contribution 1: Timing anomalies in MBTA

- TA do not break any MBTA assumptions
  - If Analysis is representative of Deployment, timing estimates hold

- **Challenge:** know whether **TA have been triggered or not**
  - Usually **requires low level control** over hardware

- TA in MBTA require the user to assess:
  1. Whether there are TA or not
  2. Whether they can occur at OT
  3. Whether they are captured at AT or not
  4. Whether the impact observed at AT upperbounds the impact in OT

Contribution 1: Timing anomalies in MBTA



Analysis Phase

Operation Phase

STA

Platform **Modelling**

- Assess the lack of TA
- If TA exist, do not discard locally-good scenarios

Platform **Actual Behavior**

MBTA

Representative Measurements

- Assess the lack of TA
- Assess if TA can arise at operation
- If TA can arise, ensure analysis measurements capture them

Operation-time Measurements

- Compared to TA in STA:
  - No need to model timing anomalies
  - No need to analyze the potential states
  - TA need to be handled in MBTA too, but in a different way
- However:
  - **Difficult to obtain representative tests** that capture them at AT
    - Not all aspects of hardware can be controlled
  - If TA are not captured at AT, timing estimates become unreliable
- **TA can be handled with MBPTA**

Towards Limiting the Impact of Timing Anomalies
in Complex Real-Time Processors

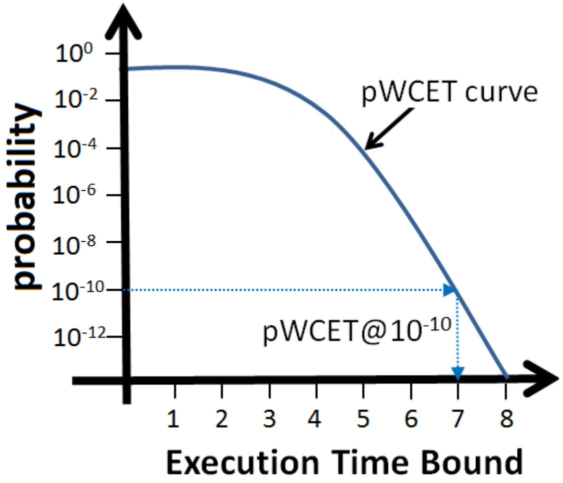
Contribution 2:
handling TA for MBPTA

Asia and South Pacific
Design Automation Conference

January 22nd
Tokyo, Japan

Introduction to MBPTA

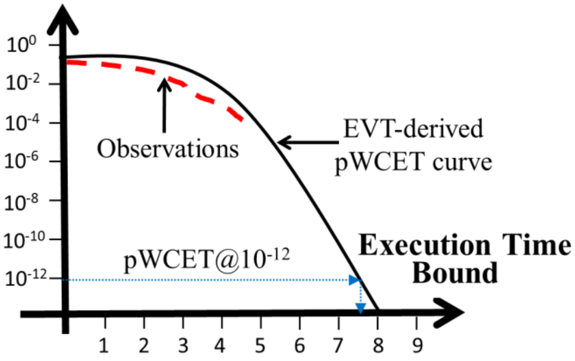
- MBPTA
  - Probabilistic variant of MBTA
- Probabilistic?
  - Even for the most critical systems, safety measures are set to deal with a non-null fault probability.
    - Safety measures ensure that faults do not become failures.
  - Residual risk must be proven low enough
- MBPTA delivers a probabilistic WCET (pWCET) upper-bounding the residual risk of a timing fault.



Introduction to MBPTA

- MBPTA changes platform behaviour so that:
  - Sources of time variation (jitter) handled by the platform not the user
  - How? Changing the behaviour of individual ‘events’
    - Randomizing
      - The impact of the different sources of jitter is probabilistically captured
    - Upper-bounding
      - Each measurement captures the worst-case impact of this type of event
- Use Extreme Value Theory (EVT) for tail projection:
  - Different runs can cover different sources of jitter
  - EVT computes the probability that they happen together

MBPTA = Randomization + EVT



Timing anomalies in MBPTA: observations

- TA in time-deterministic processors can happen systematically
  - Not a problem for STA, only need to know if it can happen
  - In MBTA the frequency of the timing anomaly is a problem, since it can affect the reliability of the WCET

- Time randomized processors + MBPTA: break systematic behavior
  - The **occurrence** of some timing anomalies **becomes probabilistic**

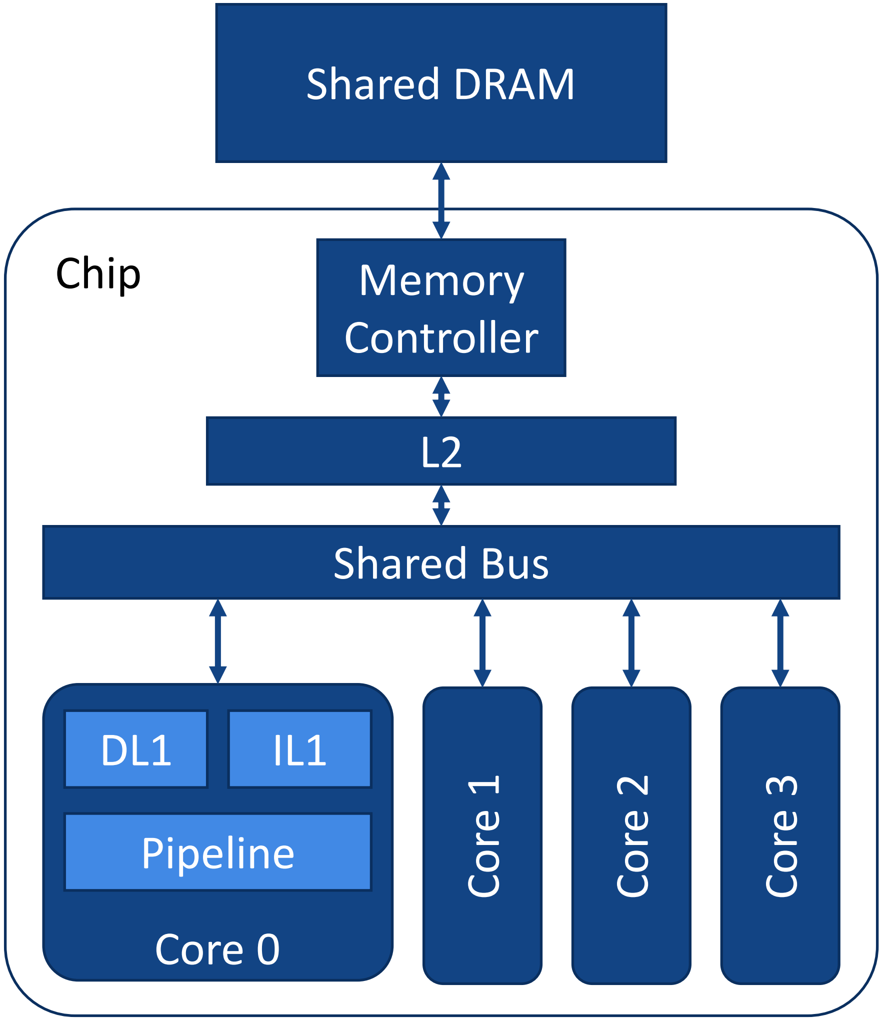
- Constant execution time events cannot trigger timing anomalies
  - **Same behavior at AT and OT** regardless of the state

Taxonomy of timing anomalies in MBPTA

- Types of timing anomalies:
  - No timing anomalies
    - Fixed-latency events
  - Probabilistically-controlled timing anomalies
    - If the probability of an event is the same at AT and OT, the behavior is probabilistically bounded
  - Potentially uncontrolled timing anomalies
    - Different probability of events at AT and OT
    - Non-probabilistic variable latency events

Case study: multicore processor

- Enhanced LEON3
  - Implemented in FPGA
  - Commercially available
- Sources of jitter can cause timing anomalies
- Sources of jitter in LEON3:
  - FDIV/FSQRT operations
  - Cache memories
  - Randomly arbitrated resources

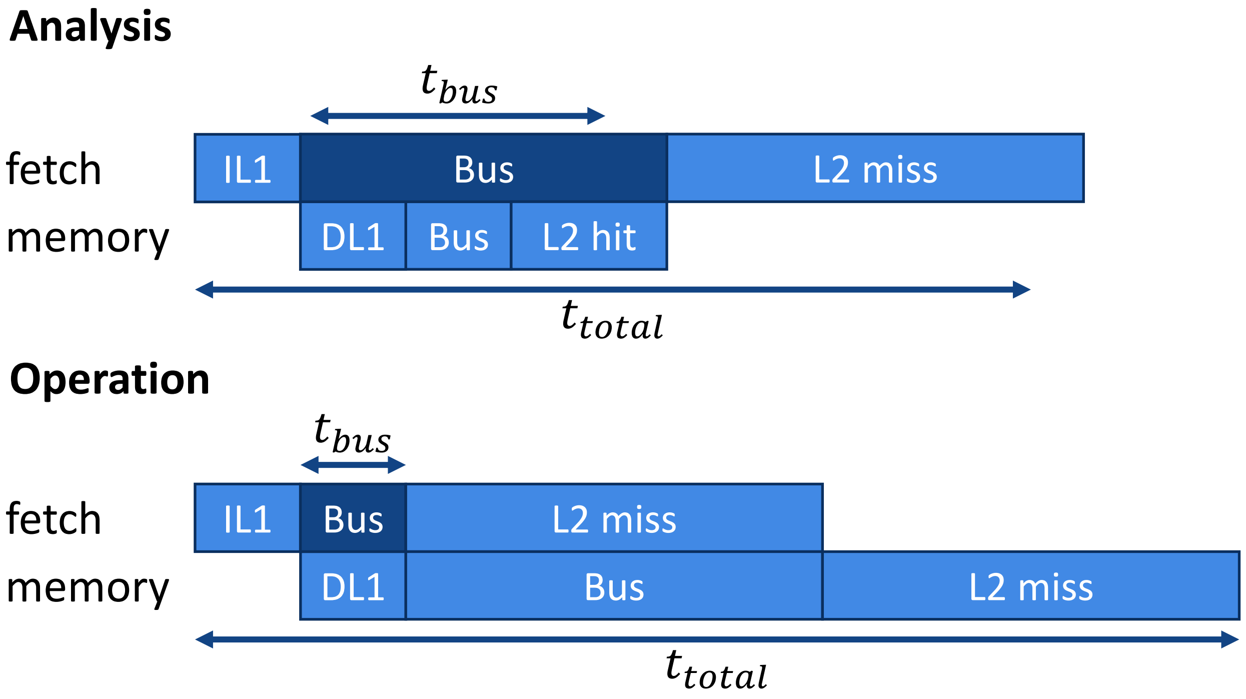


Variable latency units

- FDIV and FSQRT
- Upperbounding Variable-Latency Units
  - Force all operations to take the longest possible time
- Minimum impact on average performance
- No jitter, no timing anomalies

Cache memories: Priority inversion

- Variable latency depending on hit or miss
- Bus arbiter grants access to L2
  - At analysis, worst-case is enforced. Wait max possible time
  - At operation, worst-case is not enforced. Wait required time



Priority inversion in MBPTA

- Possible scenarios depending on AT and OT
  a) Priority inversion happens systematically at AT; or with the same/higher probability at AT than OT
  b) Priority inversion does not happen at AT; or happens with lower probability at AT than OT
- Scenario a) is covered by MBPTA
  - Execution time measurements at analysis account for same or worse conditions than operation
- Scenario b) is not covered by MBPTA
  - The potential impact on timing that these events can have needs to be quantified

Quantifying the impact of TA

- Case b) can be quantified using probabilistic properties
- Random caches used in enhanced LEON3
  - Random placement and random replacement
  - Evictions happen with a given probability
- Probability of a miss evicting the later referenced cache line:
  - $P_{evict} = \frac{1}{S_{L2} \cdot W_{L2}}$
- Δ upperbounds of the impact of TA
  - $\Delta \leq \frac{1}{S_{L2} \cdot W_{L2}} \cdot L2MISS_{count} \cdot L2MISS_{latency}$
- L2: 512KB, 4-way, 32B/line
- $P_{evict} = \frac{1}{S_{L2} \cdot W_{L2}} = \frac{1}{4096 \cdot 1} \approx 0.000244$

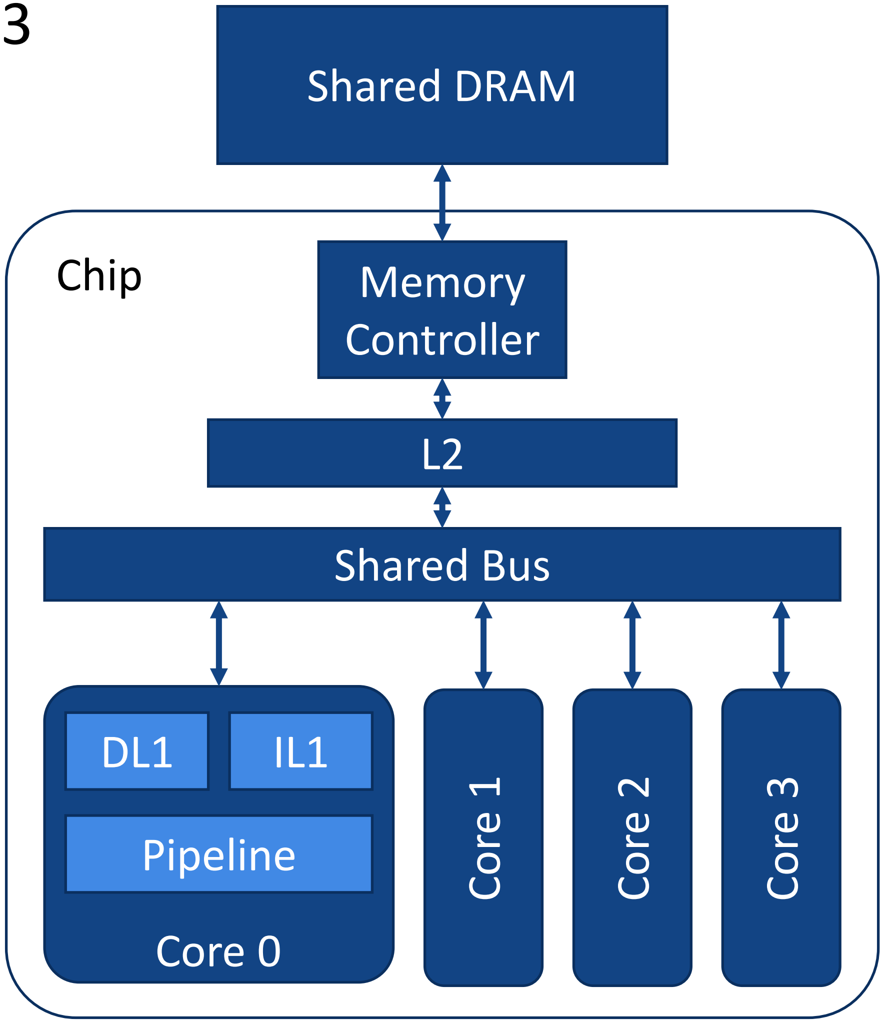
Other possible sources of TA

- Initial cache state
  - In analysis empty cache is enforced
    - More misses
  - In execution, some useful lines can be in the cache
    - Hits that can result in a TA

- Arbitration effects
  - For example: buses and memory controller
  - They determine the order of the petitions
    - Can result in a TA

Experimental setup

- FPGA implementation of LEON3
- 4 core processor
- DL1: 16KB 4-way
- IL1: 16KB 4-way
- L2: 512KB 4-way
- EEMBC auto benchmark
- 1000 runs per bench
- 10^{-12} exceedance threshold



Experimentation

- Impossible to determine if TA occur and prevent/enforce them in a real processor
- Quantitative assessment of the potential impact of TA
- We increase execution time with the upper-bound of TA
  - $Number\ of\ TA = \frac{L2\ misses}{S_{L2} \cdot W_{L2}}$
  - $Time\ increase = Number\ of\ TA \cdot 28\ cycles$

Experimental results

- Ratio of pWCET with and without TA

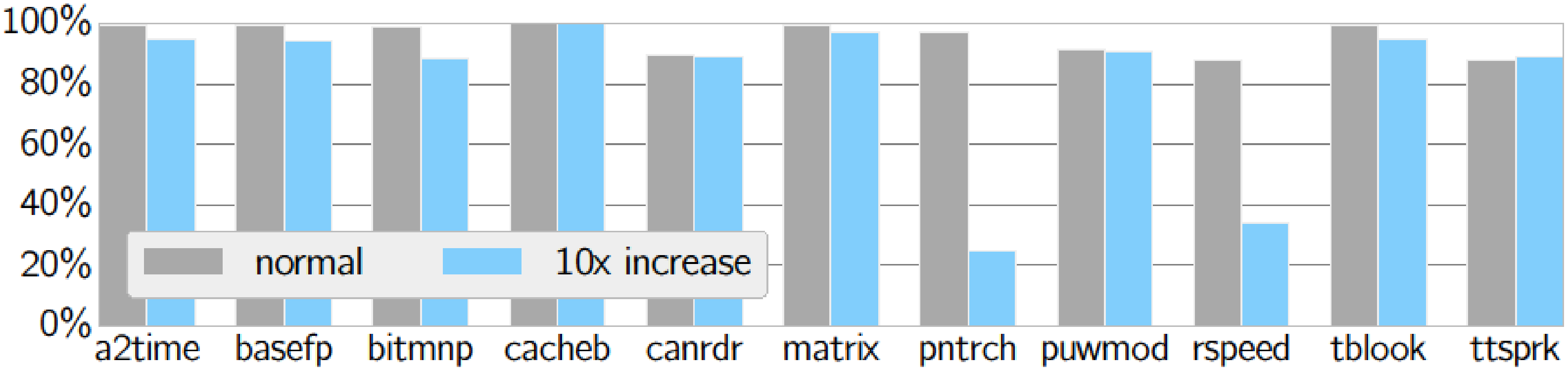
- Difference negligible:
  - `a2time` 0.007%; `tblook` 0.004%
  - Most of them below 0.001%

- Multiply TA effect by 10x
  - Difference below 0.1%
  - Below 0.01% in most cases

- Statistical assessment of the impact of TA

Statistical assessment

- Compare pWCET with execution time distributions
- Normalized confidence interval overlap w/wo TA
- For 1x impact (normal) and 10x the potential impact of TA



- Overlap is big (avobe 88% for 1x), so distributions cannot be proven different
- pntrch and rspeed have narrow confidence intervals (100s of cycles), so the estimates differ by tens of cycles

Conclusions

Conclusions

- Timing anomalies challenge timing analysis

- Timing anomalies already studied in STA

- We study timing anomalies in MBTA and MBPTA
  - Problems and classification
  - Propose solutions for handling them with MBPTA

- With an MBPTA-compliant processor in an FPGA
  - We increase execution time with the upper-bound of TA
  - Results show negligible impact on pWCET

Towards Limiting the Impact of Timing Anomalies in Complex Real-Time Processors

Thank you!
Any questions?

Pedro Benedicte, Jaume Abella,
Carles Hernandez, Enrico Mezzetti,
Francisco J. Cazorla

Asia and South Pacific
Design Automation Conference

January 22nd
Tokyo, Japan