



# Enabling the exploration of the design space of monitoring systems: a new system-level approach

**Giacomo Valente**, Luigi Pomante, Federica Caruso, Vittoriano Muttillio, and Tania Di Mascio

Università degli Studi dell'Aquila, DISIM/DEWS

*giacomo.valente@univaq.it*

# Motivations

- Nowadays, systems-on-chip frequently require both dynamic management of their HW/SW components and a careful coordination of the tasks carried-out.
- On-chip monitoring systems (OCMSs) can assist towards these objectives.
- OCMSs provide some level of intrusiveness and overhead, that can lead to barely optimized design choices.

# State of Art

- Literature is raising up the abstraction level at which monitoring requirements are introduced in the design flow up to system level

Work	Monitored task	HW target	Type of monitor	MREQs usage
De Matos et al.	SW-tasks	Processor	SW	Code instrumentation
Seo et al. (2017)	HW-tasks	Accelerator + CHM	HW	Configuration of CHM
Hessabi et al.	SW-tasks	ASIP	HW & SW	Code instrumentation
Seo et al. (2018)	SW-tasks	Processor + CHM	HW	Configuration of CHM
Mettler et al.	SW-tasks	Processor + CHM	HW	Configuration of CHM

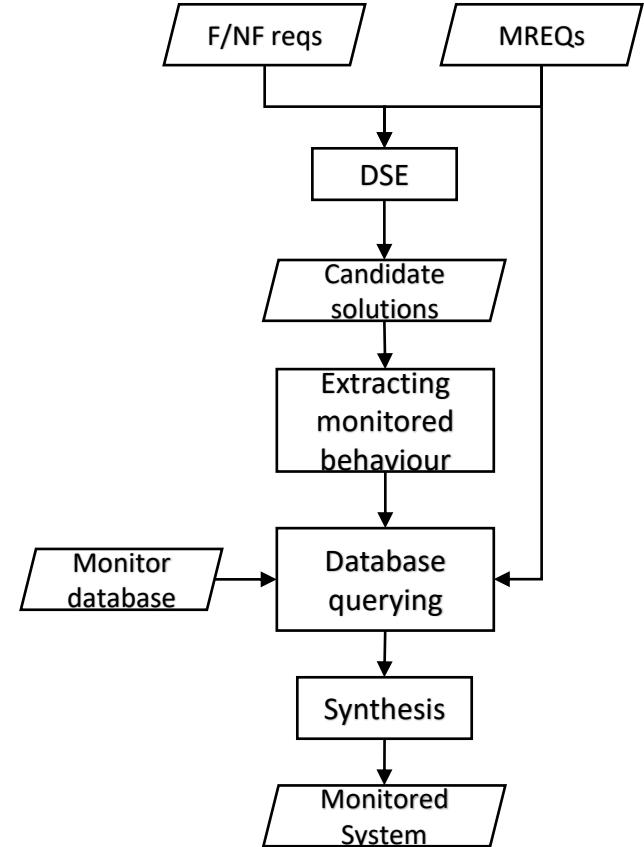
\*CHM – Configurable Hardware Monitor

# Research Question

- No literature works consider monitoring requirements as part of a design space exploration, keeping them into account only at synthesis time
- The growing of the number and complexity of monitoring requirements might blow up the final system solution
- Is it possible to consider monitoring requirements in a design space exploration, also enforcing reusability of existing OCMSs solutions?

# The proposed approach

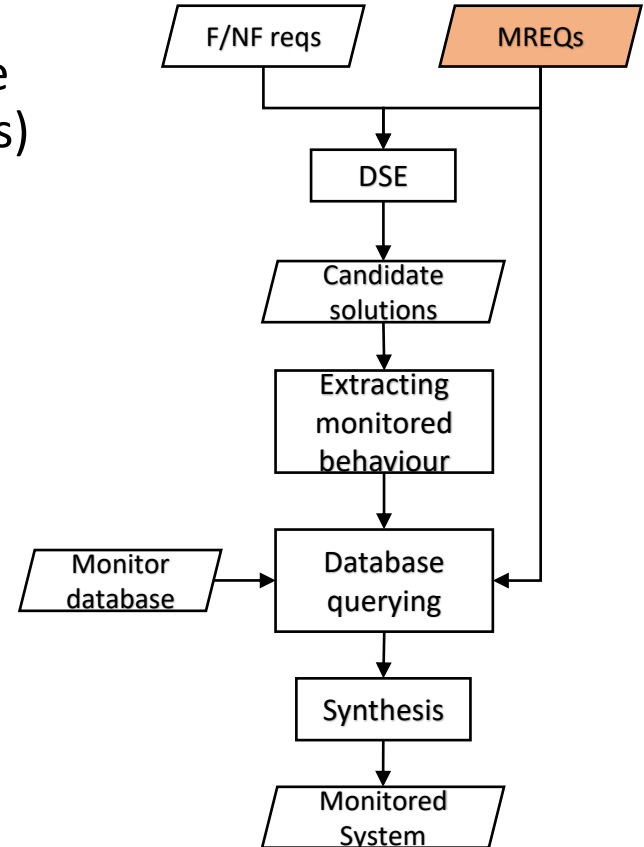
- We propose an innovative approach to enable the exploration of the monitoring systems design space at system level
- The final monitored system guarantees the compliance with system specifications, enforcing also satisfaction of the monitoring requirements (MREQs)



# Expressing monitoring requirements

- After setting F and NF requirements, designers are can also provide monitoring requirements (MREQs)

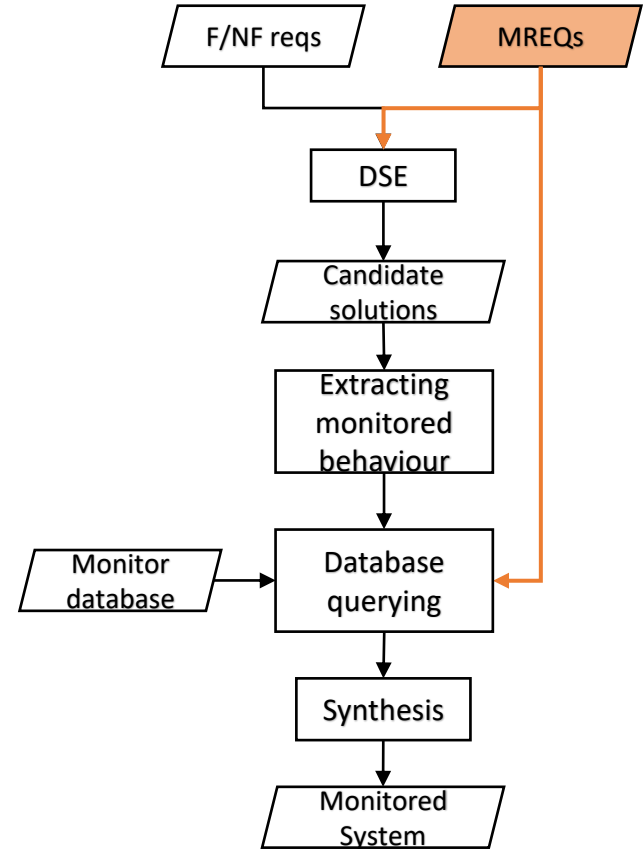
Type of MREQs	Specification	Example
Behavioural Explicit	System Behavioural Model	Number of times two processes exchange messages
Behavioural Implicit	Purpose/ Metrics	Power consumption of a process
Structural	Purpose/ Metrics	Number of cache misses of a process



# Taking into account MREQs

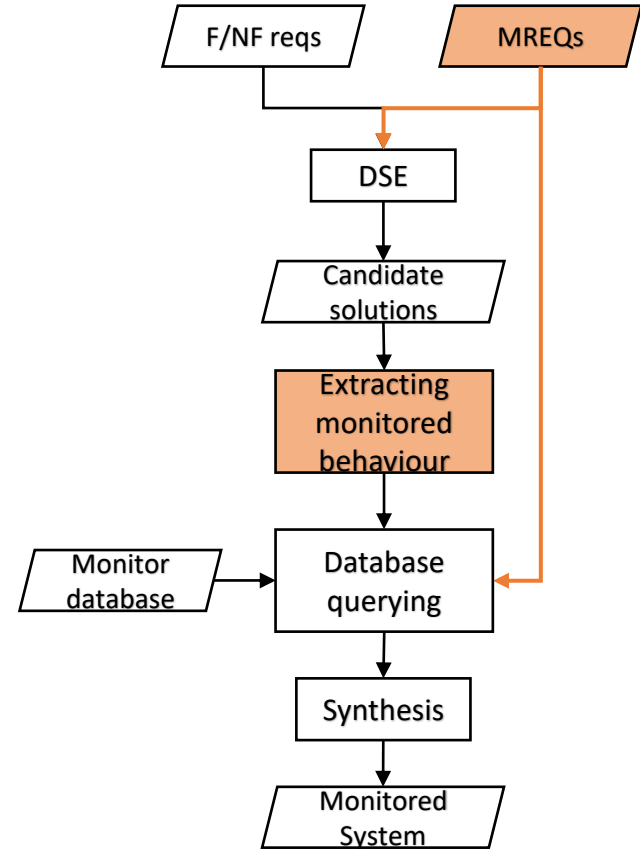
Type of MREQs	Specification
Behavioural Explicit	System Behavioural Model
Behavioural Implicit	Purpose/ Metrics
Structural	Purpose/ Metrics

- Behavioural Explicit ones taken into account in the DSE
- Behavioural Implicit and Structural ones taken into account after getting candidate solutions
  - First checking among existing monitoring systems
  - Then generating a custom solution



# Extra-parameters identification

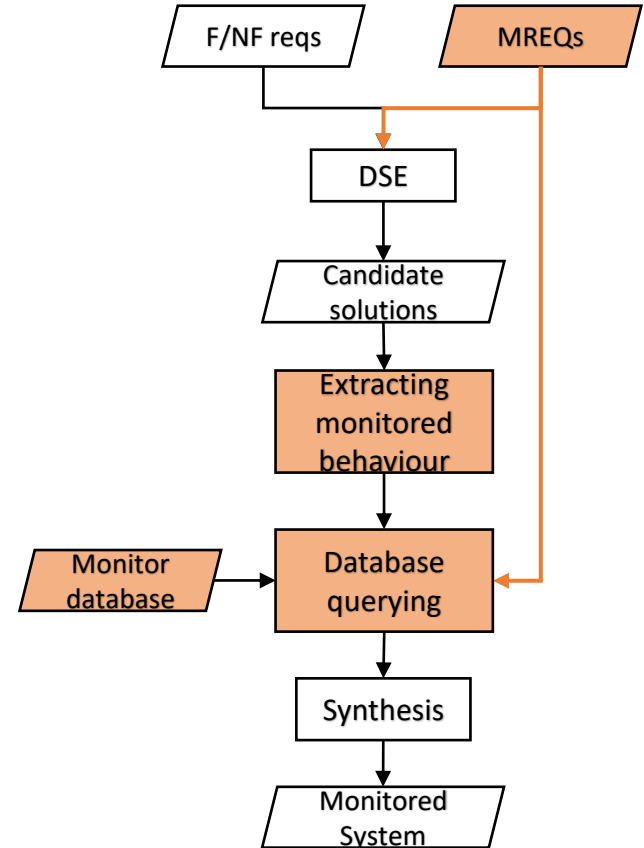
- After DSE step, different candidate solutions are available. Their analysis provides margins to satisfy Behavioral Implicit and Structural requirements
- By analyzing the candidate solutions, the proposed approach identifies some extra-parameters:
  - Platform parameters:
    - HW target, Exceptions, Multi-thread, Multi-core, Policy, Resource sharing, Software platform, System-wide, Type of physical implementation





# Querying a database of existing monitors

- Extra-parameters drive the query in a database containing all monitoring systems existing in literature.
- Database is called MONICA, and it characterizes monitoring systems by means of some parameters
- MONICA is open-source and also usable through a GUI
  - <https://monicatool.cloud/>



# MONICA GUI

Getting Started

**Find Mode**

My Queries

Insert Mode

My Instances

Still Mode

Evaluation Mode

About MONICA

Change Password

Logout

www.univaq.it

☰ Find Mode: set your monitoring requirements and get on-chip monitoring systems satisfying them.

Provide your monitoring requirements by clicking on each MONICA parameter and setting the corresponding value.

**Design**

EXCEPTIONS	<input type="checkbox"/> ...	RECORD-REPLAY	<input type="checkbox"/> ...
HARDWARE TARGET	<input type="checkbox"/> ...	RESOURCE SHARING	<input type="checkbox"/> ...
METRICS	<input type="checkbox"/> ...	SOFTWARE PLATFORM	<input type="checkbox"/> ...
MULTI-THREAD	<input type="checkbox"/> ...	SYNCHRONIZATION	<input type="checkbox"/> ...
ORGANIZATION	<input type="checkbox"/> ...	SYSTEM-WIDE	<input type="checkbox"/> ...
POLICY	<input type="checkbox"/> ...	TECHNIQUE	<input type="checkbox"/> ...
PURPOSE	<input type="checkbox"/> ...	MULTI-CORE	<input type="checkbox"/> ...

**Drawbacks**

DESIGN COST	<input type="checkbox"/> ...
DETECTION LATENCY	<input type="checkbox"/> ...
MEMORY SPACE	<input type="checkbox"/> ...
PERFORMANCE DEGRADATION	<input type="checkbox"/> ...
PHYSICAL COST	<input type="checkbox"/> ...
POWER DISSIPATION COST	<input type="checkbox"/> ...
SPECIFICATION MODIFICATION	<input type="checkbox"/> ...

**Benefits**

ACCURACY	<input type="checkbox"/> ...
EXTENSIBILITY	<input type="checkbox"/> ...
GRANULARITY	<input type="checkbox"/> ...
PORTABILITY	<input type="checkbox"/> ...

**Implementation**

AVAILABILITY	<input type="checkbox"/> ...
TYPE OF PHYSICAL IMPLEMENTATION	<input type="checkbox"/> ...

**Extensions**

AUTOMATIC GENERATION OF MONITORS	<input type="checkbox"/> ...
RUNTIME MANAGEMENT	<input type="checkbox"/> ...

**Your input**

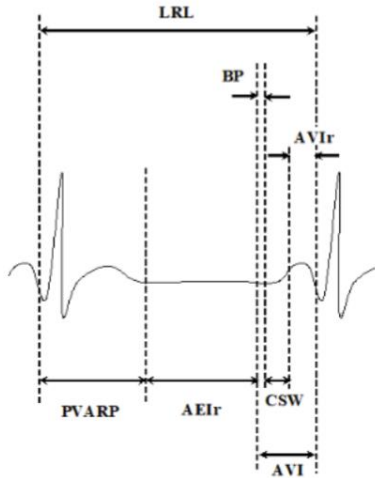
Grievous\_496563

CLEAN YOUR INPUT

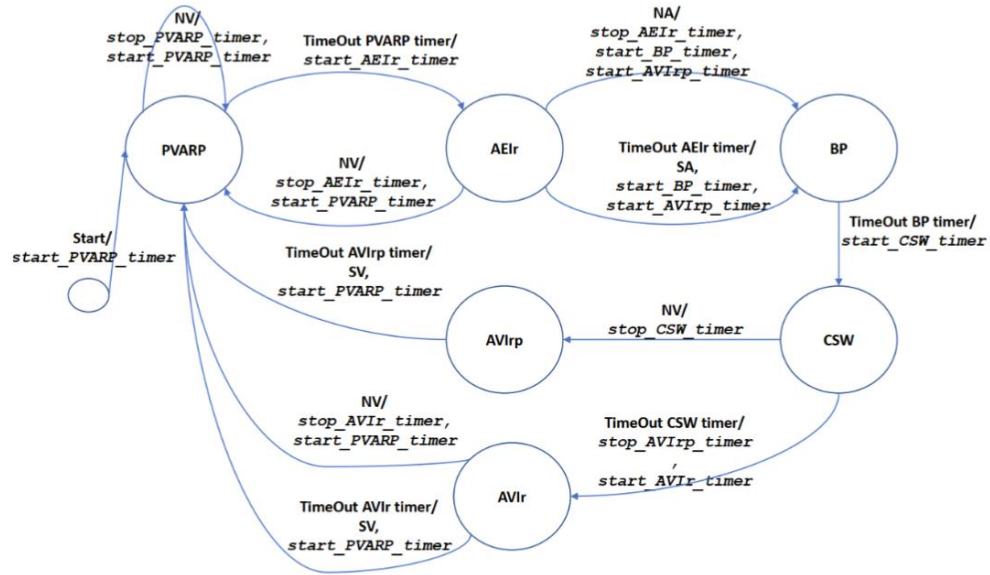
FIND

# Case study: pacemaker

- The proposed approach has been integrated in an existing HW/SW co-design tool, called Hepsycode (<https://www.hepsycode.com/>).
- As case study, we plan to design a monitored pacemaker.



Time Intervals	Duration(ms)
PVARP	400
AEIr	400
BP	25
CSW	75
AVIr	100



# Non-Functional and MREQs

- NF requirements
  - Timing: duration of LRL equal to the value that gives rise to a difference of 1 bpm.
  - Area: lowest possible.
- MREQs
  - Unexpected Timer Fired (UTF): a timer is monitored to understand when is found already fired in the CSW state (e.g., due to undetected manufacturing physical defects or ageing)
  - Unexpected Natural Event (UNE): the pacemaker receives natural events when the system is not in a state able to manage them (e.g., due to a not correctly synchronized heart rate with the actual one);
  - Unexpected States Sequence (USS): the pacemaker is moving along a state sequence that is not correct (e.g., due to undetected manufacturing physical defects or ageing)
  - Monitor the power consumption

# Preliminary Results

<i>Solutions</i>	<i>GPP</i>	<i>SPP</i>	<i>Connection</i>	Shared internal timer	Relative Size	I/O	oCMSs
#1	8051	BPP	custom	x	1.8	GPIO and custom PTPs	[29] [30] [31]
#2	-	All	custom	-	2.7	custom PTPs	[31]
#3	Leon3	-	AHB	x	3	GPIO	[29] [30]

- [29] Zoni et al., “PowerTap: All-digital power meter modeling for run-time power monitoring”
- [30] Najem et al., “A Design-Time Method for Building Cost-Effective Run-Time Power Monitoring”
- [31] Pagliari et al. “All-digital embedded meters for on-line power estimation”

# Conclusions and Future Works

- We presented an ESL approach for HW/SW Co-Design of Monitorable Embedded Systems
- Designers can provide their monitoring requirements and automatically get a monitored solution as output.
- The approach has been integrated into an existing HW/SW co-design tool, called Hepsycode, and a preliminary evaluation has been performed designing a monitored pacemaker.
- Future works:
  - to complete the integration with the Hepsycode tool, making it available to scientific community.
  - to integrate the proposed approach with other HW/SW co-design tools
  - to integrate the proposed approach with existing frameworks generating on-chip monitoring systems



**Thank you!**  
**Any Questions?**