Technische Universiteit
Eindhoven
University of Technology

Reliable Embedded Multi-media Systems ?

Twan Basten

Department of Electrical Engineering
Electronic Systems

'Knowing is not understanding.'
Charles Kettering

2 Acknowledgements

Joint work with:

- Marc Geilen
- Sander Stuijk
- AmirHossein Ghamarian
- Valentin Gheorghita
- …

Projects:

- NWO PROMES
- EC FP7 MNEMEE

3 Overview

- Embedded Multi-media
- Analysis of Synchronous Dataflow Graphs
  - Throughput Analysis
  - Throughput-Storage Trade-off Analysis
  - Parametric Throughput Analysis
- Multiprocessor System-on-Chip Design
- Looking Forward

4 Embedded Multi-media

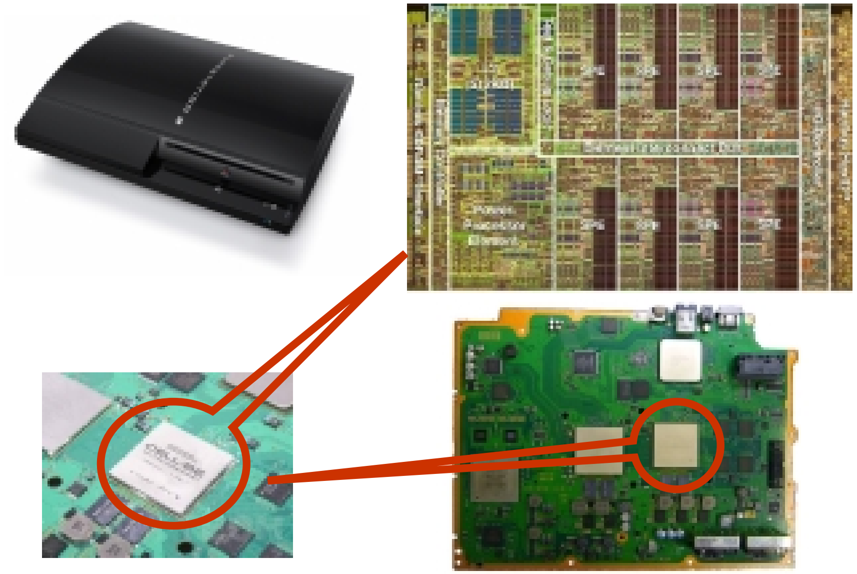
Trends
Concurrency
Connectedness
Interaction
Variability

Embedded Multi-media

TU/e

Embedded Multi-media

TU/e



Embedded Multi-media

TU/e

Approach: Models of Computation (MoCs)

TU/e

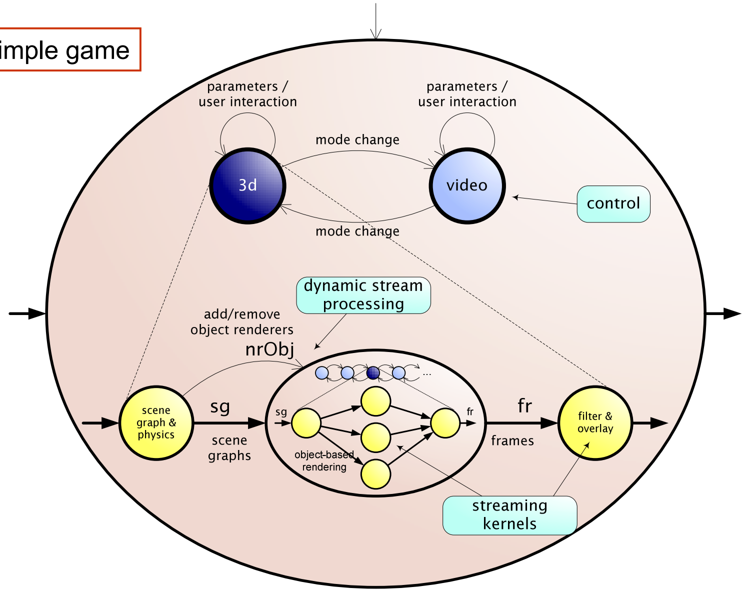
on the interface between science and engineering

essential for predictability

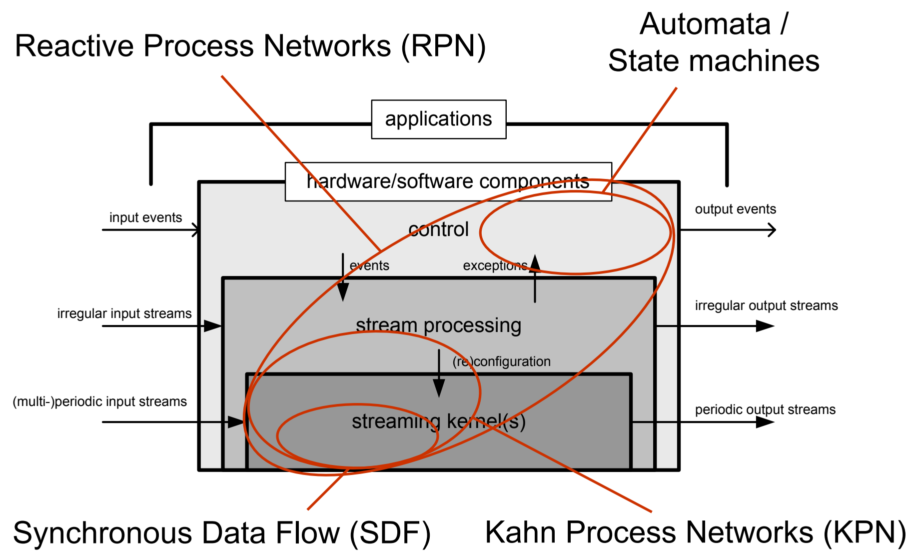
Basic dimensions	Aspects	Derivatives
processing	functionality concurrency timing energy quality …	execution time energy dissipation
communication	functionality concurrency timing energy quality …	perceived quality reconfiguration time
storage		…

9 An Example Model

A simple game



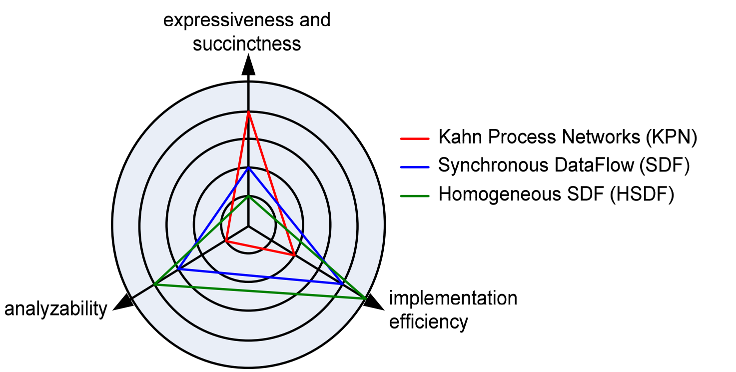
10 Multi-media MoCs



11 Essential Challenges

predictability and efficiency

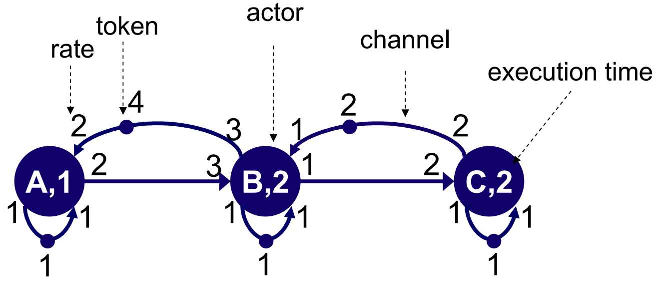
… are contradictory



12 Overview

- Embedded Multi-media
- Analysis of Synchronous Dataflow Graphs
  - Throughput Analysis
  - Throughput-Storage Trade-off Analysis
  - Parametric Throughput Analysis
- Multiprocessor System-on-Chip Design
- Looking Forward

13 Synchronous Data Flow Graphs, SDFGs [Lee 1986]

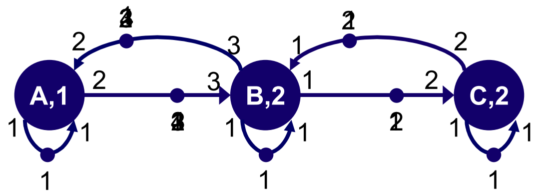


Throughput: average number of actor firings over time

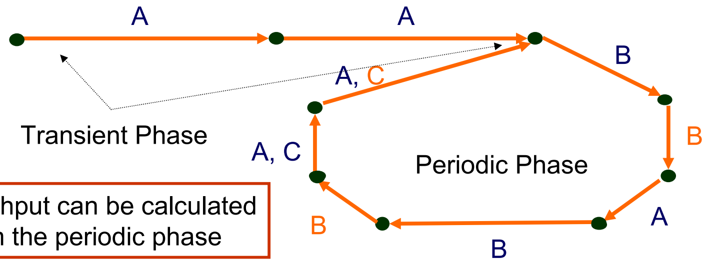
14 Self-timed Execution

- Each actor fires as soon as it can fire
- Known to maximize throughput

15 Our Throughput Analysis Method



State: distribution of tokens + remaining execution times



Throughput can be calculated from the periodic phase

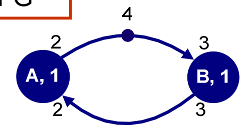
Efficient implementation
Consider a subset of the firings of one actor only to detect a recurrent state

16 Traditional Method for Throughput Analysis

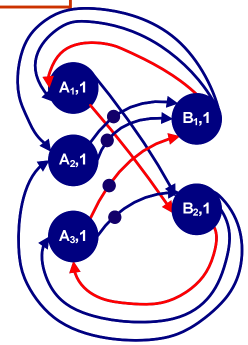
- Convert the SDFG to a Homogeneous SDFG (HSDFG)
- Compute the throughput of the HSDFG

Disadvantage:
conversion may lead to an explosion in the size of the graph

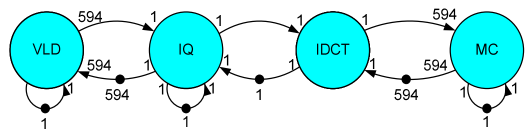
SDFG



HSDFG



Dataflow graph H.263 decoder



HSDFG with 1190 actors

17 Throughput Analysis: Our Result

All use conversion to HSDFG

	State Space	Dasdan Gupta	Howard	Young Tarjan Orlin
MP3 dec.	$1. 10^{-3}$	$1. 10^{-3}$	$1. 10^{-3}$	$1. 10^{-3}$
Modem	1.10^{-3}	$82. 10^{-3}$	$81. 10^{-3}$	$81. 10^{-3}$
Sample Rate	$2. 10^{-3}$	>1800	>1800	>1800
Satellite	$56. 10^{-3}$	>1800	>1800	>1800
H.263 decoder	$10. 10^{-3}$	>1800	>1800	>1800

Runtimes of various throughput analysis methods (in seconds)

18 Conclusions Throughput Analysis

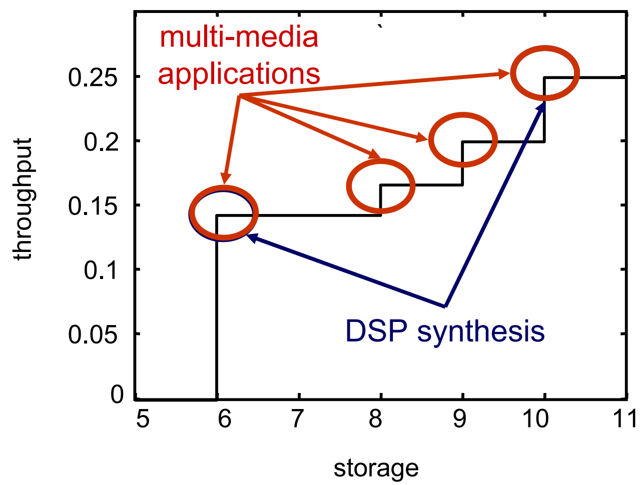
- The first method to perform throughput analysis directly on SDFGs
- Experimental results show that the state-space technique outperforms all other existing techniques
- Can be easily integrated in simulation tools

19 Overview

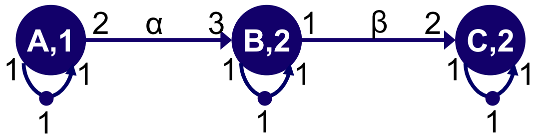
- Embedded Multi-media
- Analysis of Synchronous Dataflow Graphs
  - Throughput Analysis
  - Throughput-Storage Trade-off Analysis
  - Parametric Throughput Analysis
- Multiprocessor System-on-Chip Design
- Looking Forward

DAC 2006, IEEE T on Comp

20 Trade-off: Storage vs. Throughput

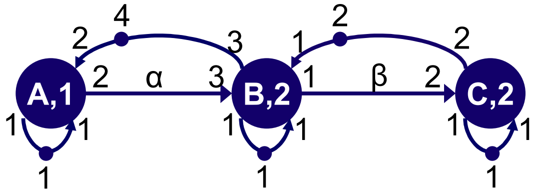


21 Storage Distribution



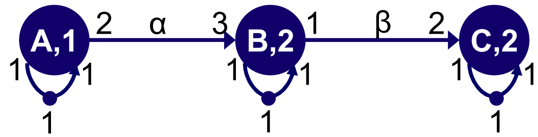
Tokens on channels must be stored in memory.

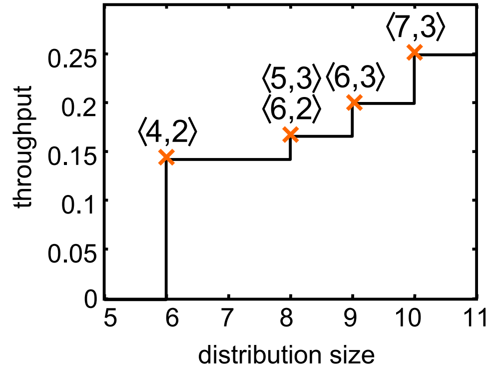
- Separate memory for each channel.
- Backward channels used to model storage space.



Storage distribution ⟨α, β⟩ → ⟨4,2⟩

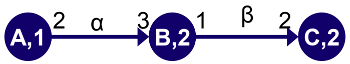
22 Problem Definition





Find all minimal storage distributions for any possible throughput

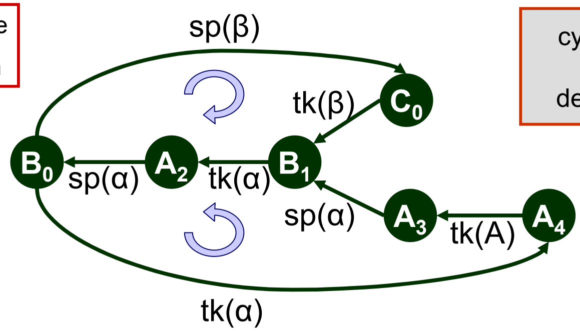
23 Dependency Graph



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A_0	A_1	B_0	B_0	A_2	B_1	B_1	C_0	C_0							
							A_3	A_4	B_2	B_2	A_5	B_3	B_3	C_1	C_1
														A_6	A_7

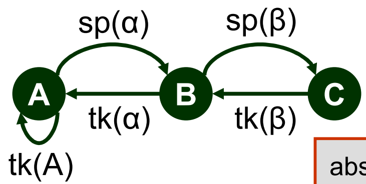
sp - space

tk - token

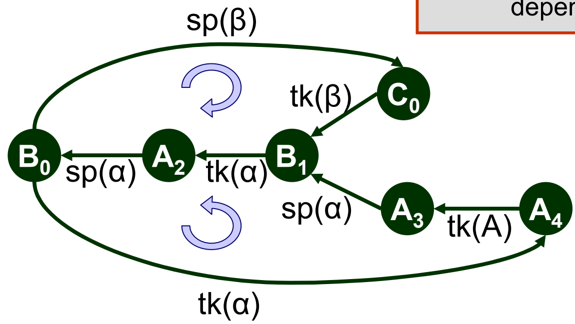


cycles denote storage dependencies

24 Abstract Dependency Graph



abstract dependency graph contains all storage dependencies



25 Design-Space Exploration Algorithm

Given an SDFG G with storage distribution δ

1. Compute throughput and abstract dependency graph Δ for G with δ
2. For each channel c with a storage dependency in Δ
3. Create new storage distribution by enlarging c
4. Repeat steps 1-2

All minimal storage distributions are found

when starting from storage distribution ⟨0,...,0⟩

26 Experimental Results

	Example	Satellite	MP3	H.263
#actors	3	22	13	4
#channels	2	26	12	3
min throughput > 0	1/7	0.18	1/137	1/21876
Distr. size	6	1542	12	4753
Max throughput				1/10000
Distr. size				8006
#Pareto points				3254
#Distributions	5	2	3	3254
Exec. time	1ms	7ms	2ms	53min

Approximation

increase buffers with n times
the minimal step size

27 Conclusions Trade-off Analysis

- Exact minimum memory requirements for any possible throughput.
- Technique to search space of storage distributions efficiently.
- Experiments show feasibility of the approach.
- Technique can be combined with heuristics to prune the search space if necessary.

28 Overview

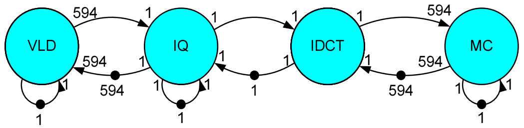
- Embedded Multi-media
- Analysis of Synchronous Dataflow Graphs
  - Throughput Analysis
  - Throughput-Storage Trade-off Analysis
  - Parametric Throughput Analysis
- Multiprocessor System-on-Chip Design
- Looking Forward

29 Parametric Throughput Analysis

TU/e

H.263 decoder

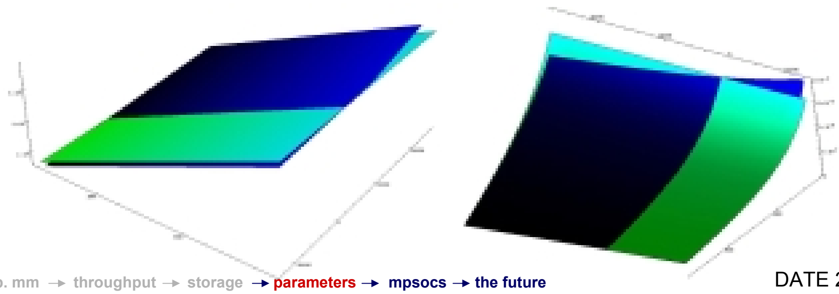
Parametric actor execution times



Divide and Conquer: divide space in throughput regions

throughput expression

$(\max\{593\ IQ + 594\ IDCT + MC,\ VLD + 594\ IQ + 593\ IDCT\})^{-1}$



DATE 2008

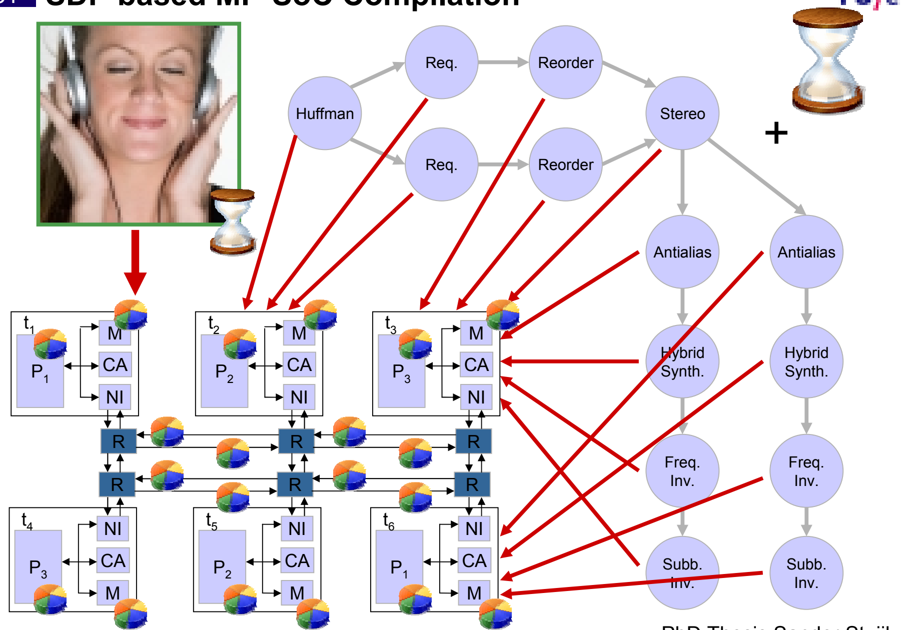
30 Overview

TU/e

- Embedded Multi-media
- Analysis of Synchronous Dataflow Graphs
  - Throughput Analysis
  - Throughput-Storage Trade-off Analysis
  - Parametric Throughput Analysis
- Multiprocessor System-on-Chip Design
- Looking Forward

31 SDF-based MP-SoC Compilation

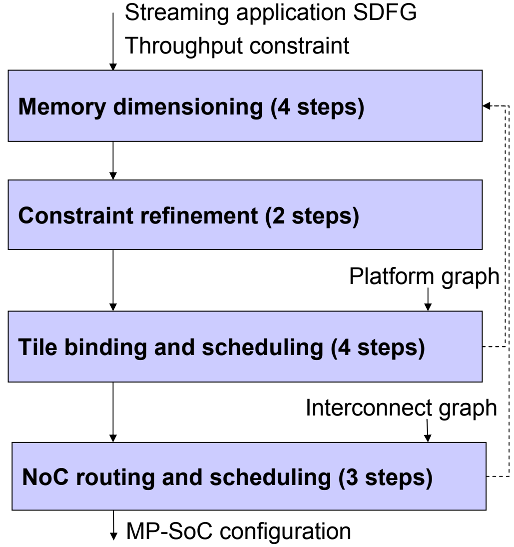
TU/e



PhD Thesis Sander Stuijk

32 SDF-based MP-SoC Compilation

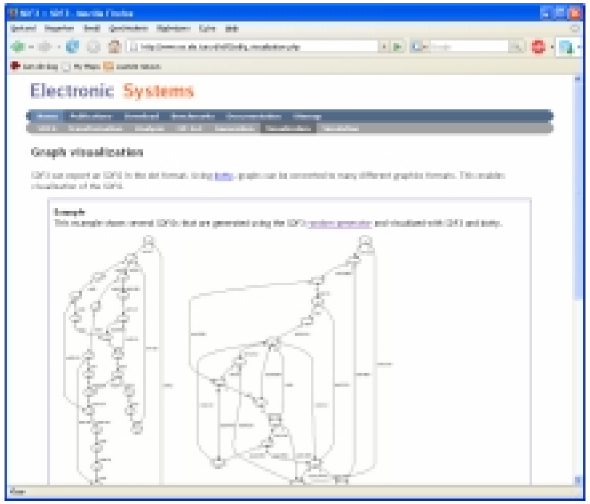
TU/e



PhD Thesis Sander Stuijk

33 SDF3 toolkit

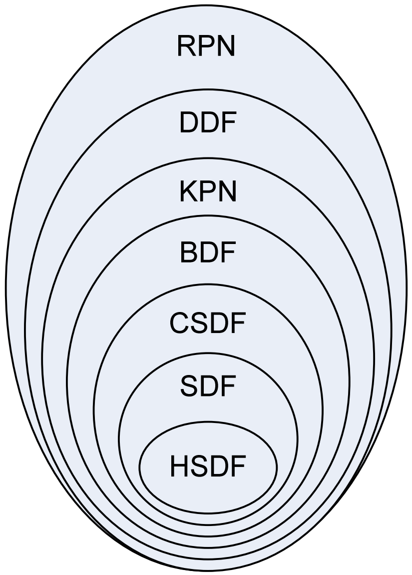
www.es.ele.tue.nl/sdf3



34 Overview

- Embedded Multi-media
- Analysis of Synchronous Dataflow Graphs
  - Throughput Analysis
  - Throughput-Storage Trade-off Analysis
  - Parametric Throughput Analysis
- Multiprocessor System-on-Chip Design
- Looking Forward

35 Horizon: Dataflow MoCs Expressiveness Hierarchy



- BDF and larger: turing complete
- Better notions of expressiveness needed

RPN: Reactive Process Networks
KPN: Kahn Process Networks
DF: DataFlow
DDF: Dynamic DF
BDF: Boolean DF
CSDF: Cyclo-Static DF
SDF: Synchronous DF
HSDF: Homogeneous SDF

36 Challenges

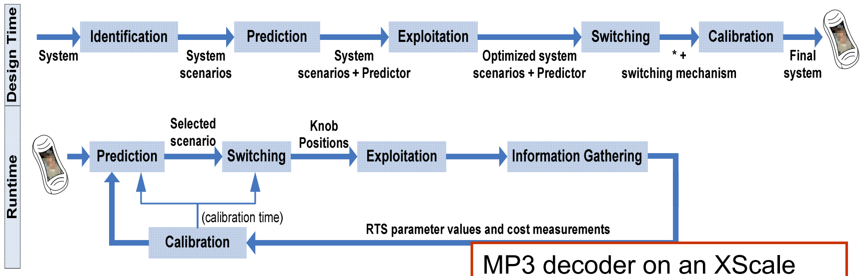
- Suitable notions of expressivity
- Expressivity while maintaining analyzability and synthesizability
- Abstraction without loosing accuracy
- Composability and compositionality
- MoCs for non-functional aspects
- Unification of MoCs
- Multi-objective trade-off analysis
- Parametric analysis
- Model-driven design and synthesis flows
- Model-driven run-time systems

37 Horizon: Scenario-based Design

TU/e

handling variability

- Design time: classification and cost characterization of behaviors
- Run time: exploitation



MP3 decoder on an XScale
- 24% energy reduction
- very small deadline miss ratio

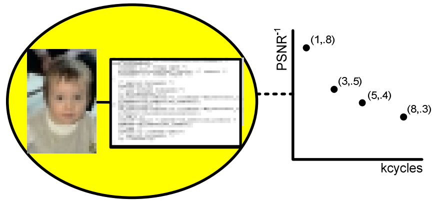
SoC 2006, IEEE D&ToC

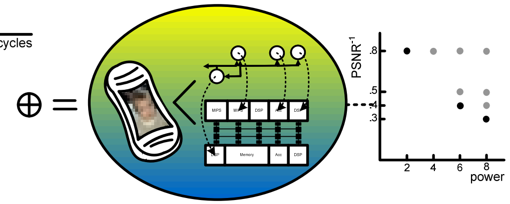
38 Horizon: Pareto Algebra

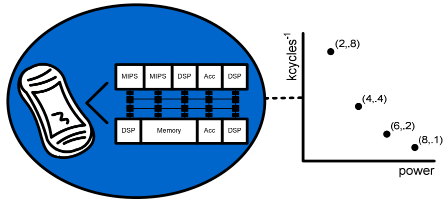
TU/e

handling variability

essential for efficiency







$S \oplus T =$

$((S \bullet T) \cap \{c \mid kcyc_S(c) \leq 1/kcyc_T(c)\})$

$\downarrow\{PSNR^{-1}, power\}$

Fundamenta Informaticae, 2007

39 Thank you !

TU/e

Questions ?

More info: www.es.ele.tue.nl/~tbasten/

'An understanding of the natural world and what's in it is a source of not only a great curiosity but great fulfillment.'
David Attenborough