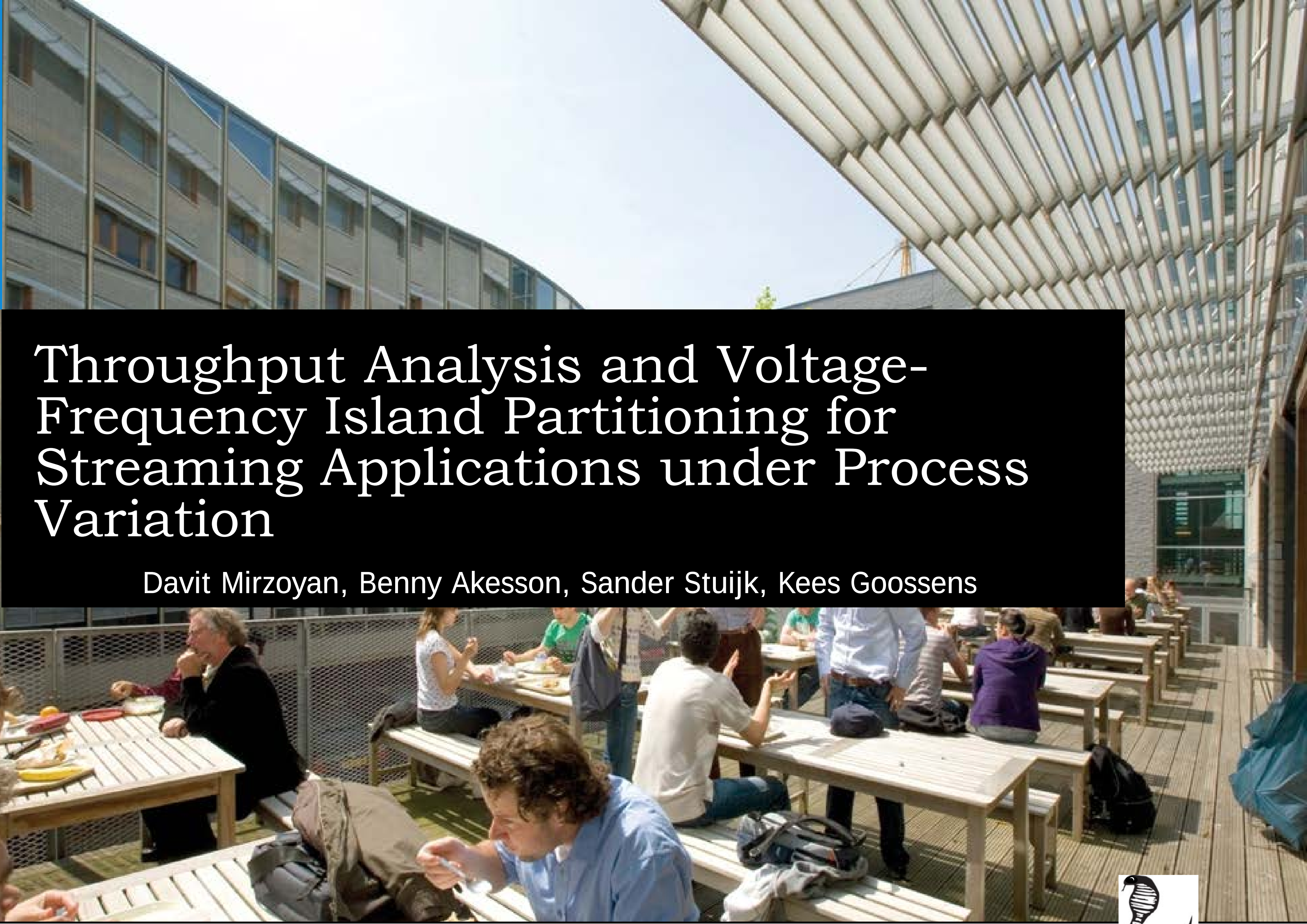


Throughput Analysis and Voltage-Frequency Island Partitioning for Streaming Applications under Process Variation

Davit Mirzoyan, Benny Akesson, Sander Stuijk, Kees Goossens

TU Delft Delft University of Technology

TU/e technische universiteit eindhoven

NEST

FlexTiles

BENEFIC

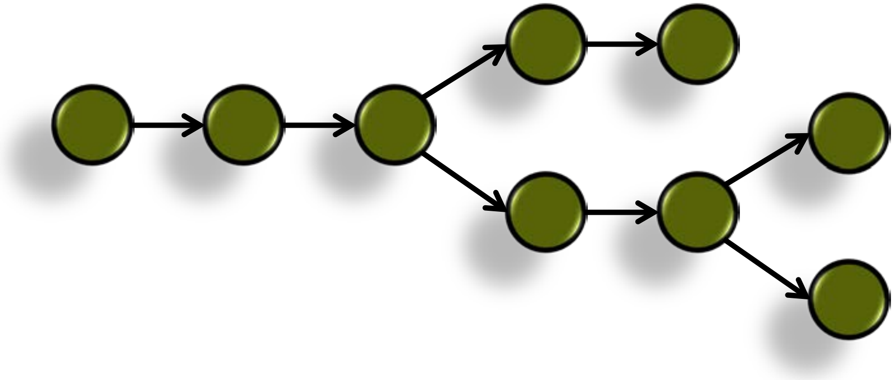
T-CREST

CA104 COBRA

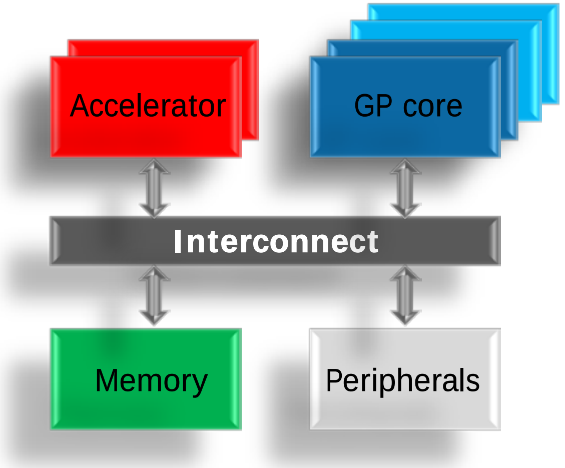
Background

- Real-time streaming applications implemented on an MPSoC

An application: a (cyclic) task graph



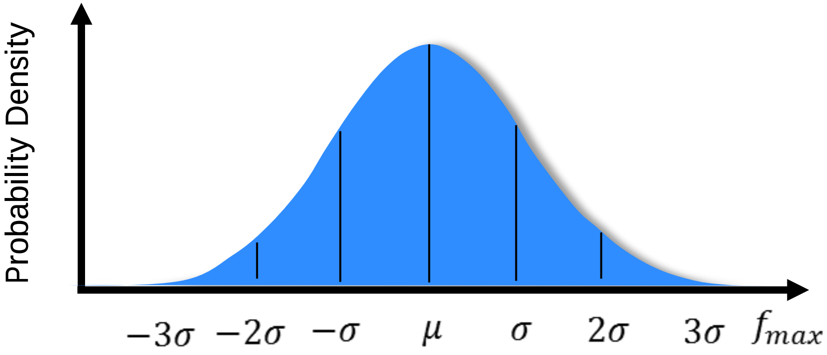
Hardware platform with ***voltage-frequency islands (VFI)***



- Mapping and scheduling to satisfy real-time requirements
- Examples
  - Software defined radio, video decoding (encoding)

Variation in Manufacturing Process

- Variation in transistor parameters
  - Die-to-die and Within-die [1]
- Variation in the maximum supported frequency of cores in an MPSoC



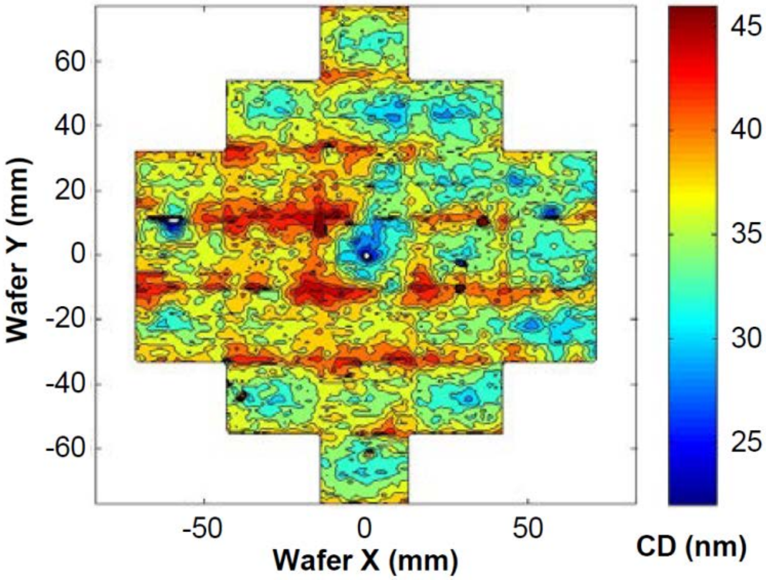
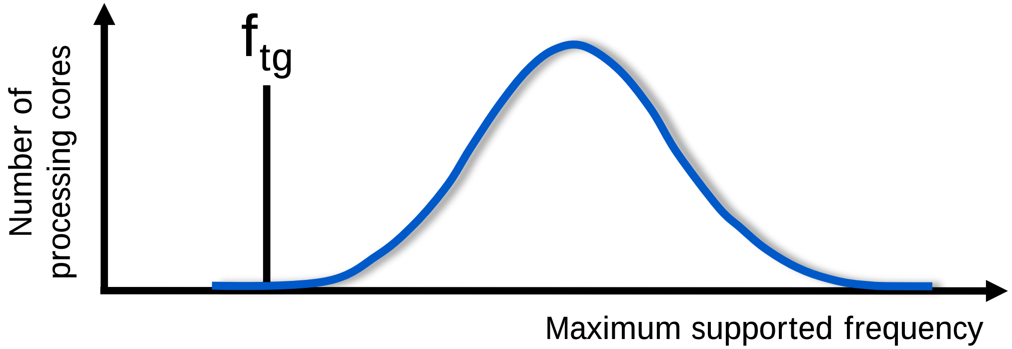


Figure 2. Full-wafer ELM CD measurements.

[1] K. P. Friedberg *et. al.* Spatial Modelling of Micron-Scale Gate Length Variation . In *Proc*. SPIE, 2006.

Conventional Worst-Case Design

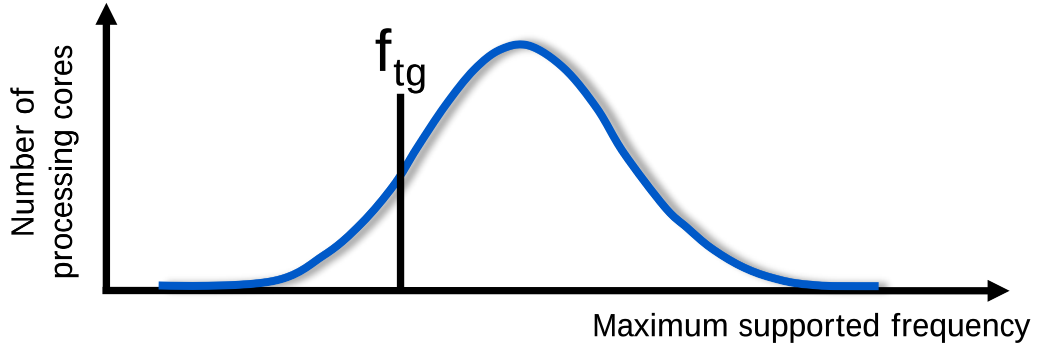
- Design margins or guard-bands (GB)
  - To guarantee the target frequency (*f_{tg}*) of cores in manufactured chips



- Deterministic core frequencies
  - Application mapping, such that a given timing requirement (e.g. ***throughput***, latency) is satisfied

Better than Worst-Case Design

- Reduced design margins
  - Smaller circuit area and thus more gross dies on a wafer
  - Target frequency of cores is not guaranteed any more

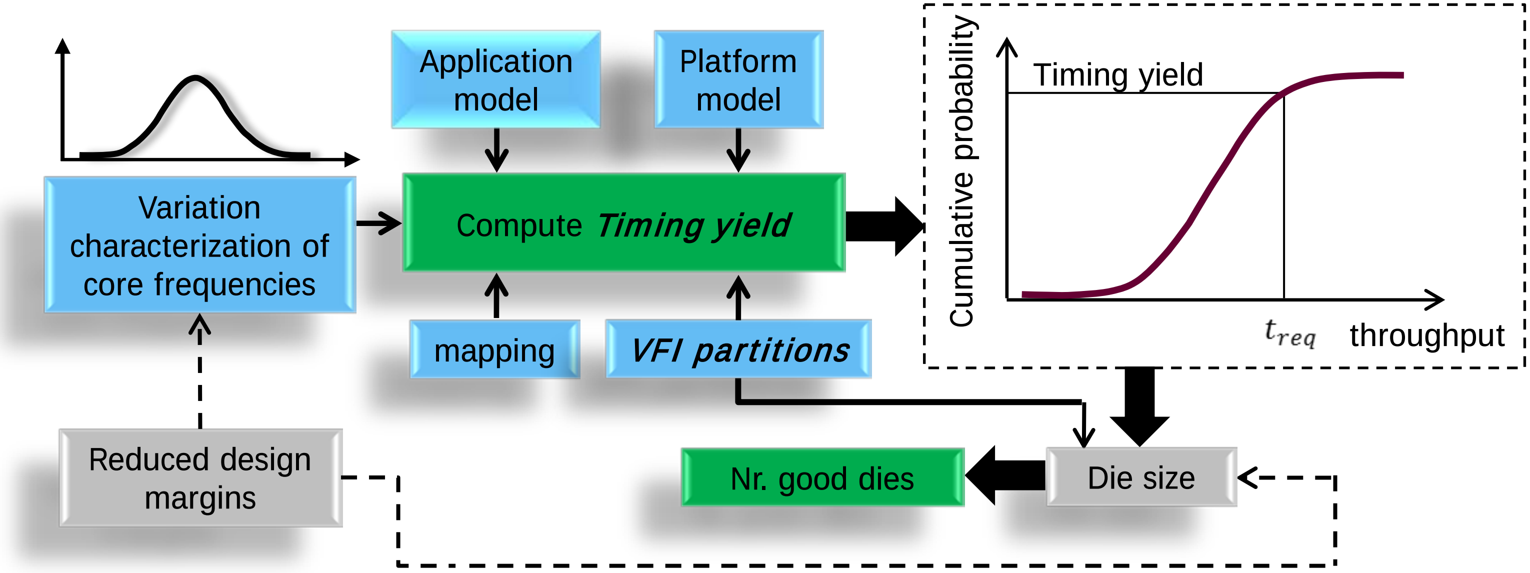


- ***We aim at maximizing the number of good dies***
  - Dies that satisfy the throughput requirement of an application
- Accomplished by exploiting frequency variation in cores in application mapping [2] and ***VFI partitioning*** steps.

[2] D. Mirzoyan *et. al.* Process-Variation Aware Mapping of Real-Time Streaming Applications to MPSoCs for Improved Yield. In *Proc*. ISQED, 2012

Contributions (Work Overview)

- Framework to compute the probability distribution of application throughput in a system with VFIs under variation
- Heuristic VFI partitioning algorithm for maximized ***timing yield***
  - Percentage of chips satisfying the throughput requirement
- Good dies = timing yield x gross dies



Outline

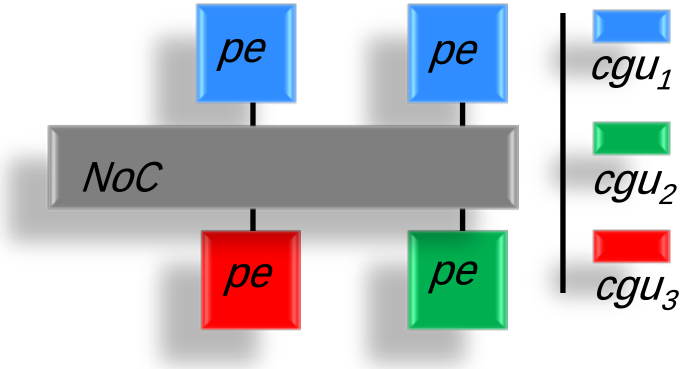
Modeling

VFI partitioning

Experimental results

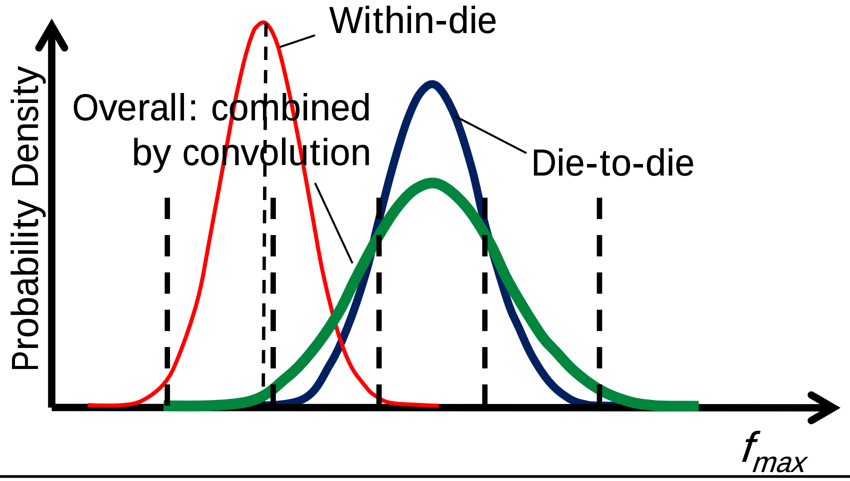
Hardware Platform

- Heterogeneous multi-processor platform
  - Multiple processing elements (PE) connected to each other by a network on chip
- Globally asynchronous and locally synchronous architecture
  - With voltage-frequency islands
- A clock-generation unit (CGU) is assigned to each VFI
  - Each CGU provides discrete frequency levels



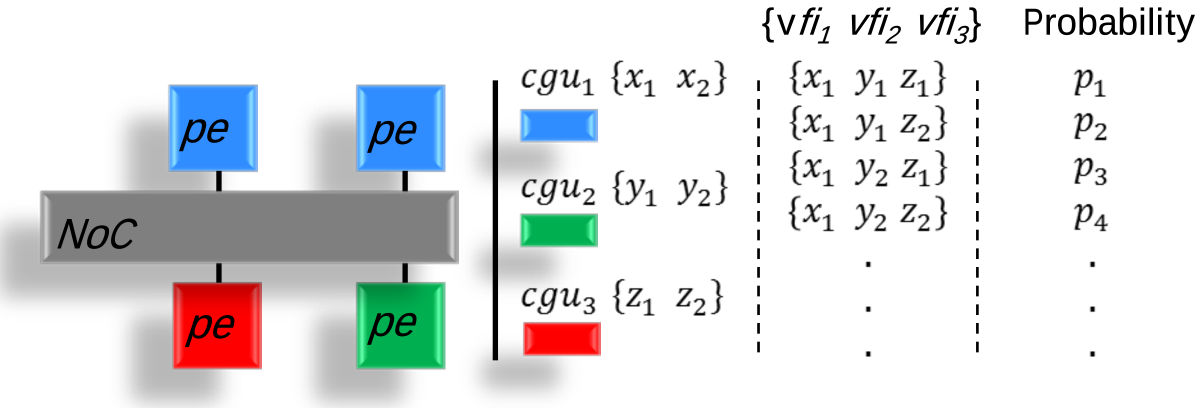
Variation in PE Frequency

- Frequency of a PE is described by normal distributions
- Two variation types
  - Die-to-die (global)
  - Within-die (local)
- Clock-frequency levels for ***VFIs*** are selected based on combined distributions



Clock-Frequency Characterization

- A set of clock-frequency levels for each VFI
- All combinations of clock-frequency levels for the chip
- Characterizing a combination with a probability
    - Probability that VFIs are operated at specified clock-frequency levels



cgu_1 $\{x_1 \;\; x_2\}$ 🟦

cgu_2 $\{y_1 \;\; y_2\}$ 🟩

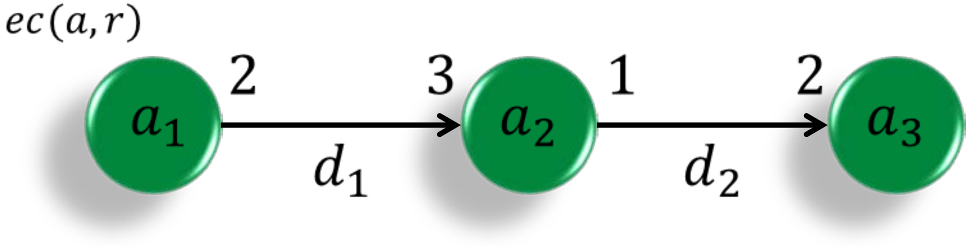
cgu_3 $\{z_1 \;\; z_2\}$ 🟥

{v*fi*$_1$ *vfi*$_2$ *vfi*$_3$}	Probability
$\{x_1 \;\; y_1 \;\; z_1\}$	p_1
$\{x_1 \;\; y_1 \;\; z_2\}$	p_2
$\{x_1 \;\; y_2 \;\; z_1\}$	p_3
$\{x_1 \;\; y_2 \;\; z_2\}$	p_4
.	.
.	.
.	.

Unbound graph

- Application modelled as a ***synchronous dataflow graph*** (SDFG)
  - Binding unaware
  - Execution times of actors are given in clock cycles of PEs

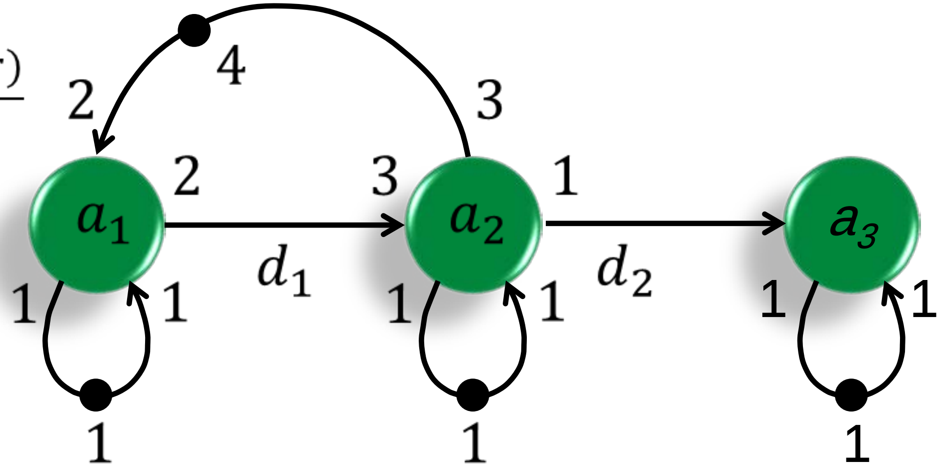
- This graph is ***decoupled*** from hardware variation



Bound graph

- SDFG model of an application mapped to PEs
  - For a given binding
    - Resource sharing: Static Order Schedules on PEs
  - Execution time of application actors (in seconds)
- ***Not decoupled*** from hardware variation

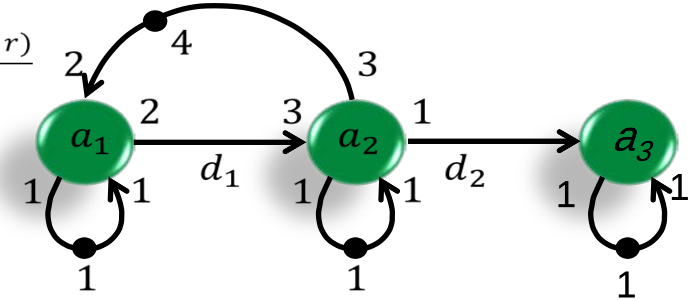
$$et(a,r) = \frac{ec(a,r)}{f_{pe}}$$

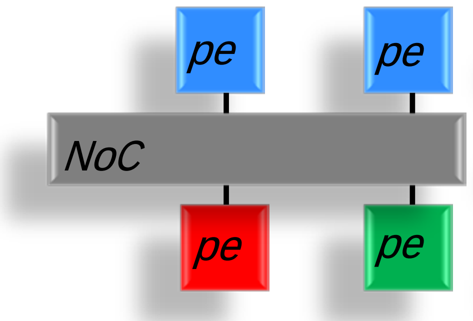


Throughput & Timing Yield

- Characterizing each combination with a ***throughput*** value

$$et(a,r) = \frac{ec(a,r)}{f_{pe}}$$



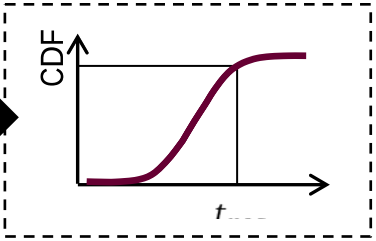


$cgu_1\ \{x_1\ \ x_2\}$ 🟦

$cgu_2\ \{y_1\ \ y_2\}$ 🟩

$cgu_3\ \{z_1\ \ z_2\}$ 🟥

{v*fi*$_1$ *vfi*$_2$ *vfi*$_3$}	Probability	Throughput
$\{x_1\ y_1\ z_1\}$	p_1	t_1
$\{x_1\ y_1\ z_2\}$	p_2	t_2
$\{x_1\ y_2\ z_1\}$	p_3	t_3
$\{x_1\ y_2\ z_2\}$	p_4	t_4
.	.	.
.	.	.
.	.	.



Timing Yield

Modeling

VFI partitioning

Experimental results

Area Cost & Timing Yield Trade-Offs

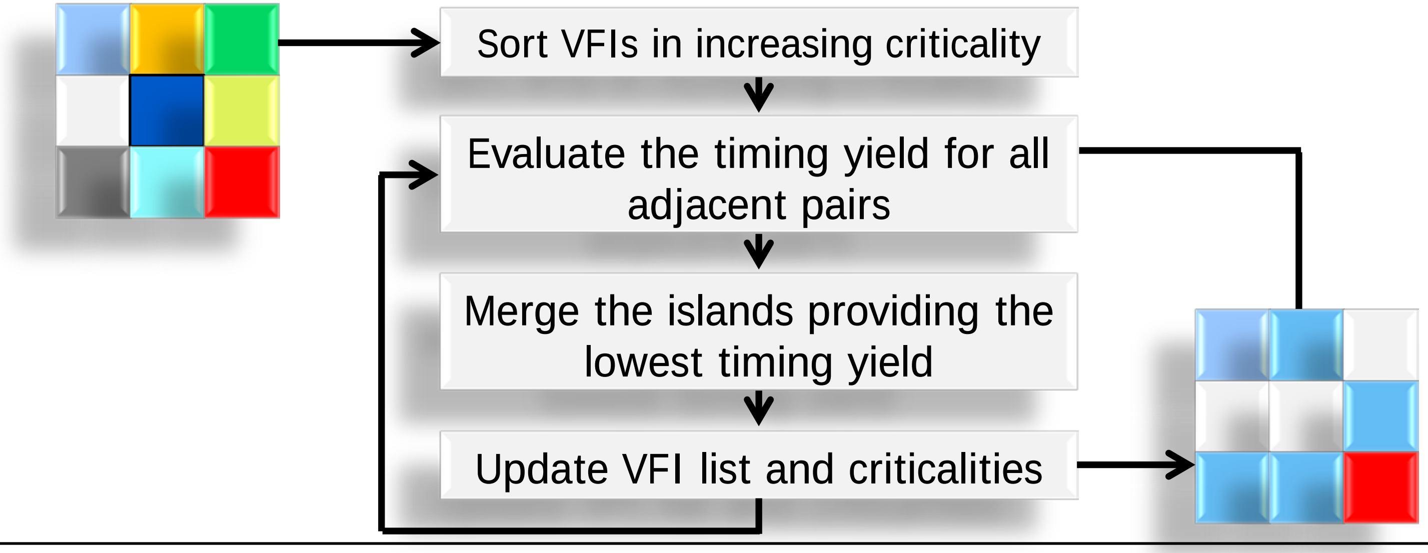
- Reducing the number of VFI partitions
  - Less clock-generation units and thus smaller circuit area
  - More dies on a wafer
  - May reduce the timing yield

- Finding a beneficial trade-off between timing yield and the number of VFI partitions, such that more good dies are obtained

- ***Partition PEs into VFIs, such that the highest timing yield is obtained***

Island criticality-based partitioning

- ***Island criticality*** metric guides the partitioning process
  - Quantifies the sensitivity of application's throughput to the frequency of a VFI
- Merge a pair of VFIs with low criticality values in iterations
  - Islands may have equal (close) criticality values



Sort VFIs in increasing criticality

Evaluate the timing yield for all adjacent pairs

Merge the islands providing the lowest timing yield

Update VFI list and criticalities

Modeling

VFI partitioning

Experimental results

Experimental setup

- MPSoC platform with 8 homogeneous PEs

- For all PEs f_n = 500 MHz
- Variation in each PE
  - Die-to-die: $3\sigma / f_n = 12\%$
  - Within-die: $3\sigma / f_n = 10\%$
    - Measurements at 45 nm technology [3]
- 8 clock-frequency levels for each VFI

- A synthetic SDFG consisting of 15 actors
  - Enough parallelism for 8 PEs

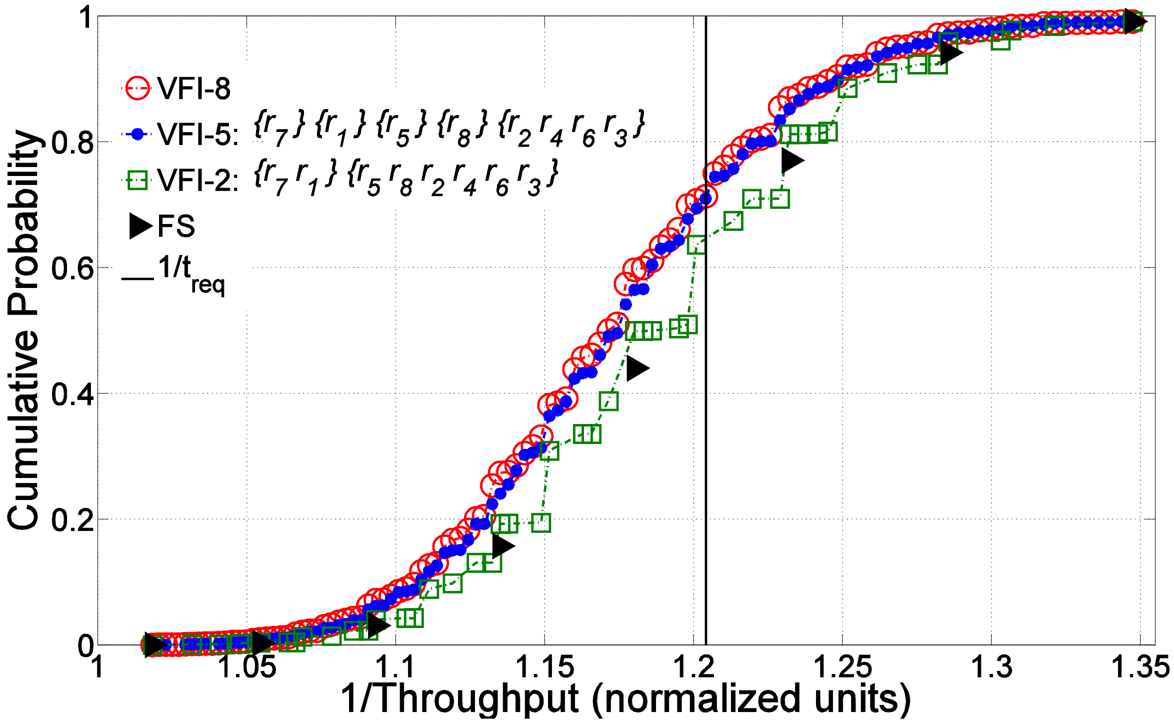
[3] L.-T. Pang *et al*. Measurement and analysis of variability in 45nm strained-si cmos technology. In *Proc.* CICC, 2008.

Timing Yield vs. VFI Granularity

- Not all reductions in Nr. VFIs reduce the timing yield
  - VFI-5 results in a negligible reduction in timing yield

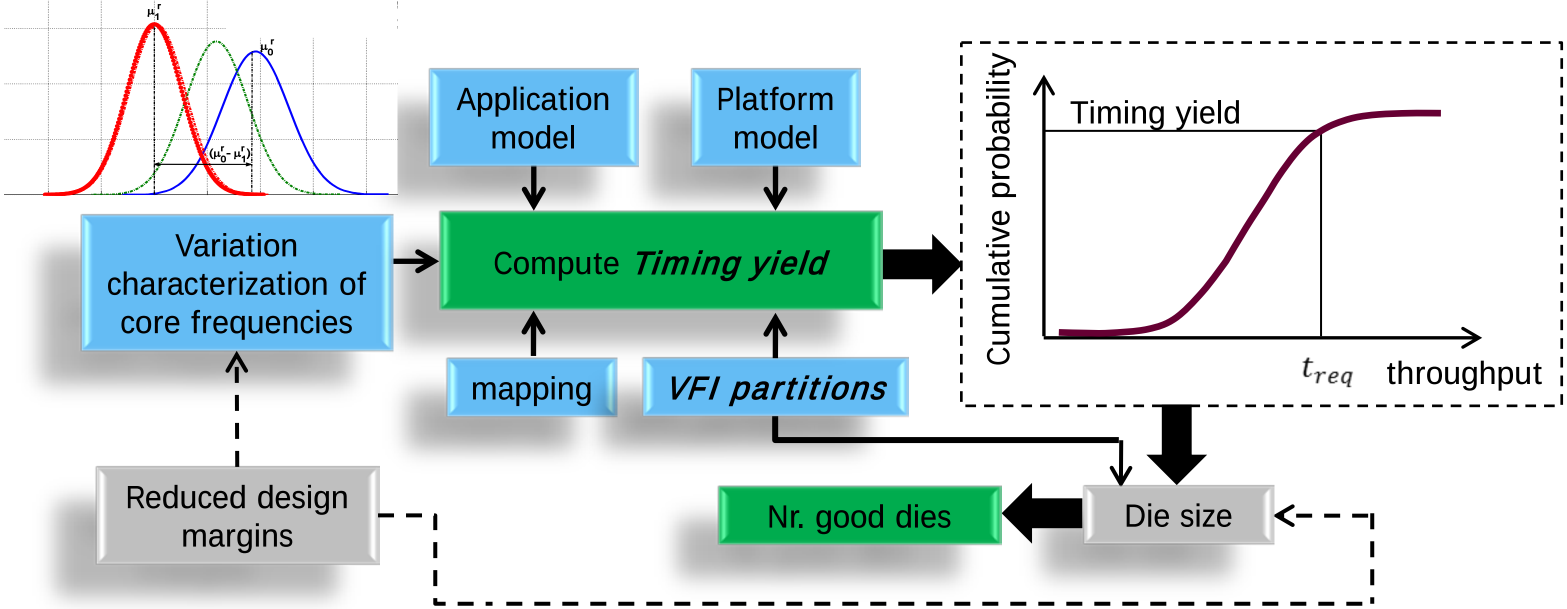
- VFI-2
  - 7% reduction
- FS:
  - 27% reduction

- VFI-5
  - More good dies?



Work Overview (recap)

- Variation and area characterization for each GB reduction



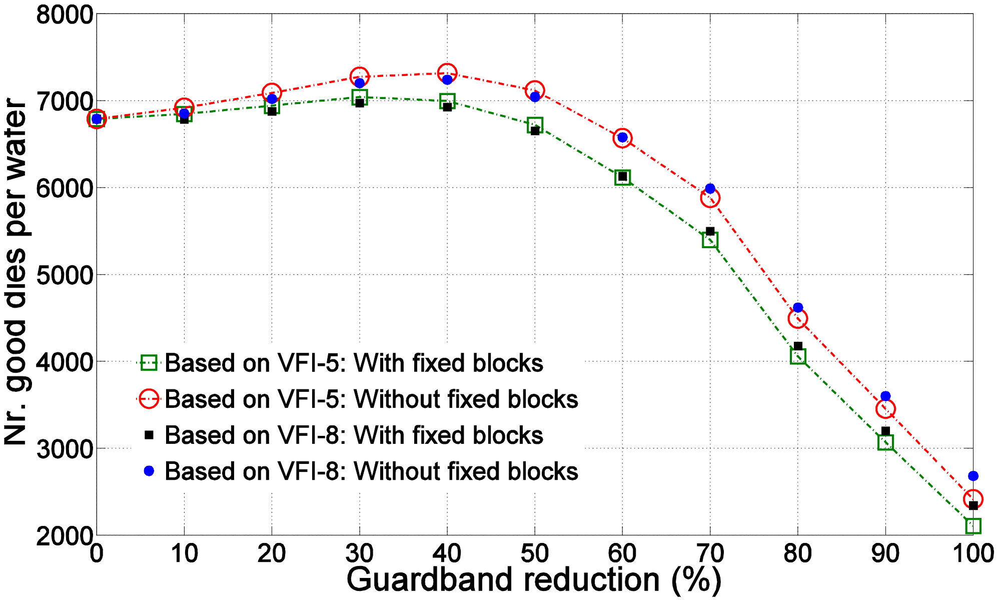
GB_{red} %	0	10	20	30	40	50	60	70	80	90	100
Y_t% (VFI-8) SB	99	99	98	97	94	88	79	69	51	38	27
Y_t% (VFI-5) SB	99	99	98	97	94	88	78	67	49	36	24

Number of Good Dies

- 10 mm^2 chip, 7 mm^2 logic, 3 mm^2 SRAM, 0.03 mm^2 per CGU
- Design with (FB) and without fixed blocks (WFB)
- VFI-8, VFI-5 architectures

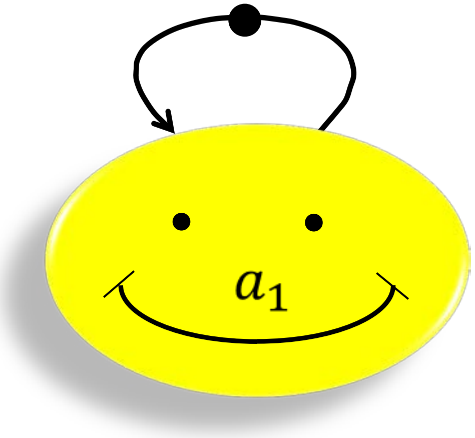
- 30% GB reduction (VFI-5)
  - 3.7% more good dies (FB)
- 40% GB reduction (VFI-5)
  - 7.7% more good dies (WFB)

- 4K wafers = 30M good dies
- 3.7% increase
  - 142 less wafers
- Wafer cost: $ 3K
  - Cost saving = $ 426K



Conclusions

- Framework to compute the probability distribution of application throughput in a system with VFIs under variation
- Heuristic VFI partitioning algorithm for maximized timing yield
- The framework is used to estimate the number of good dies
    - Dies that satisfy the throughput requirement
- It is possible to increase the number of good dies by using the proposed framework



Thank you for your attention
d.mirzoyan@tudelft.nl