Self-Calibrating Interconnects
Breaking the Worst-Case Design Paradigm

Paolo Ienne

joint work with
Giovanni De Micheli, Patrick Thiran, and Frédéric Worm

lap csda

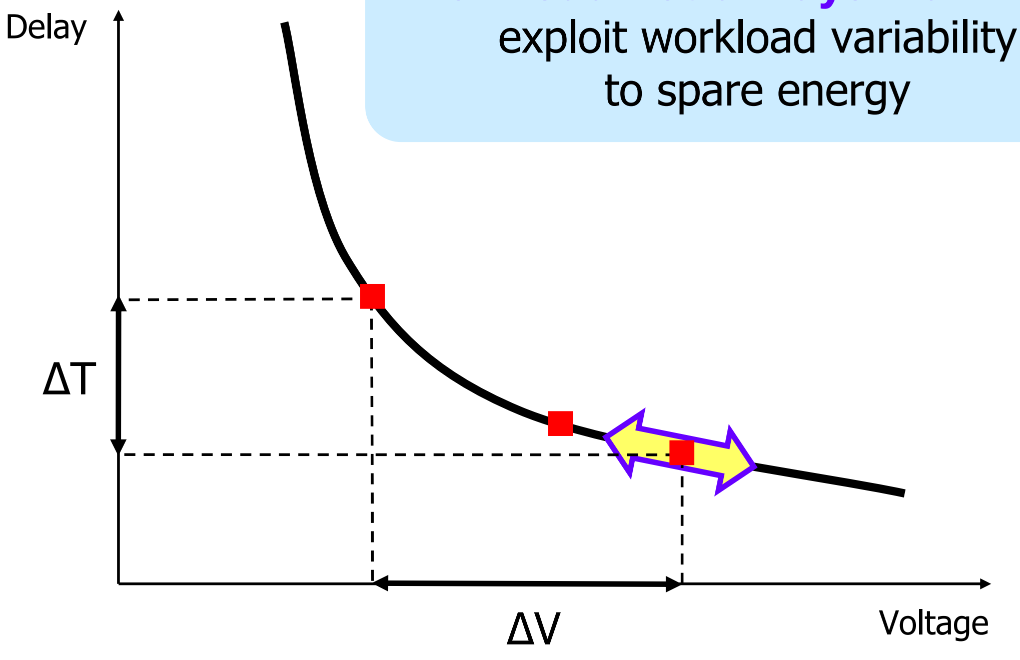
Processor Architecture Laboratory (LAP)
& Centre for Advanced Digital Systems (CSDA)

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

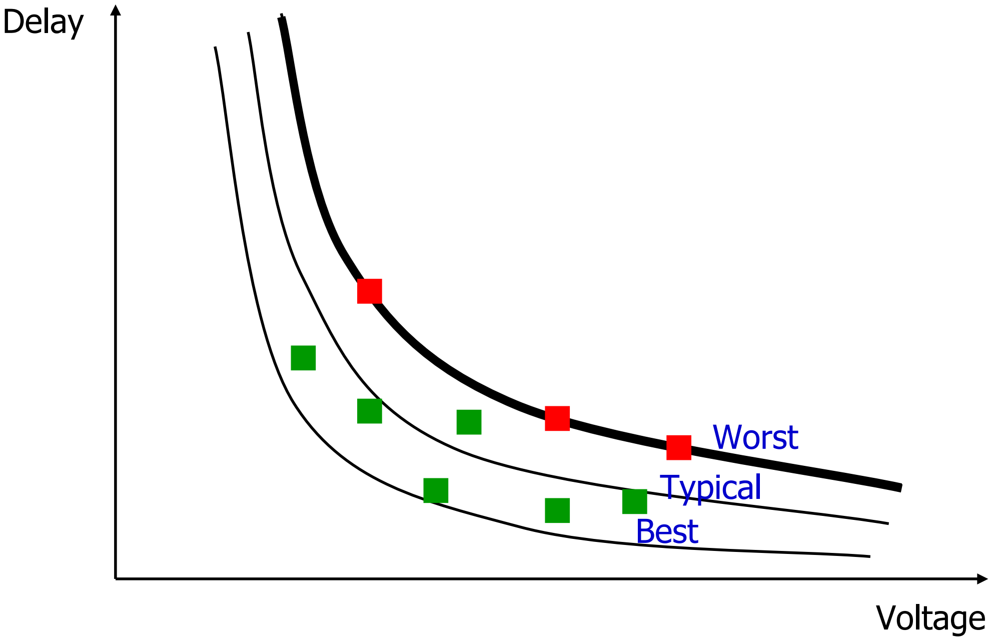
Swiss Federal Institute of Technology Lausanne (EPFL)

Dynamic Voltage and Frequency Scaling

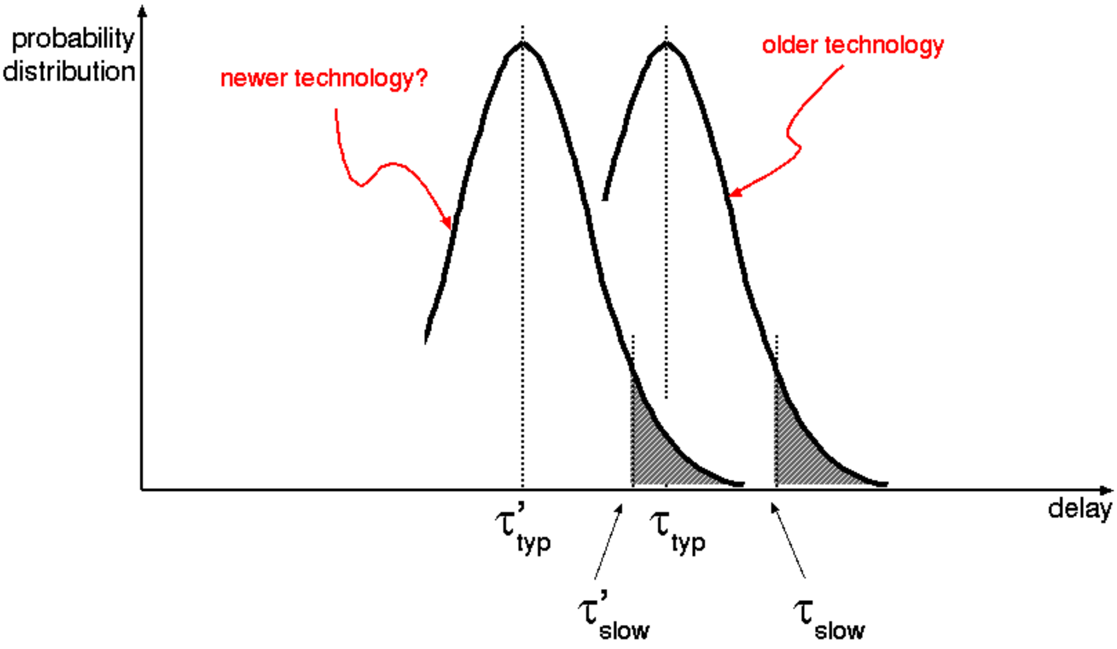
Workload not always maximum:
exploit workload variability
to spare energy



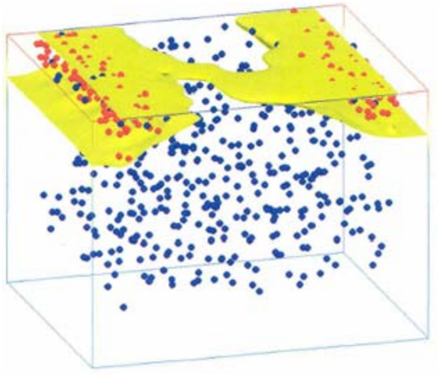
Only Worst-Case Operating Points



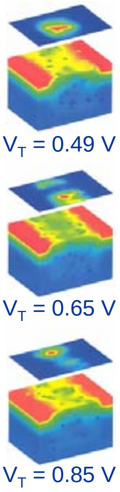
How Things Used to Evolve…



Device Unreliability



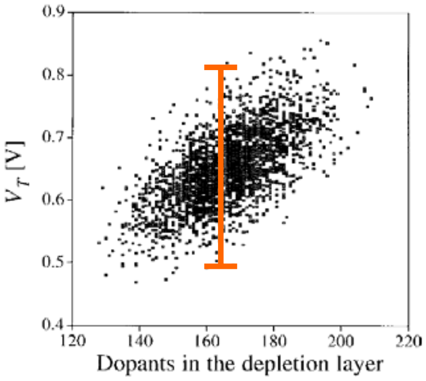
Dopant atom distribution critical to device properties (≤50nm)



V_T = 0.49 V

V_T = 0.65 V

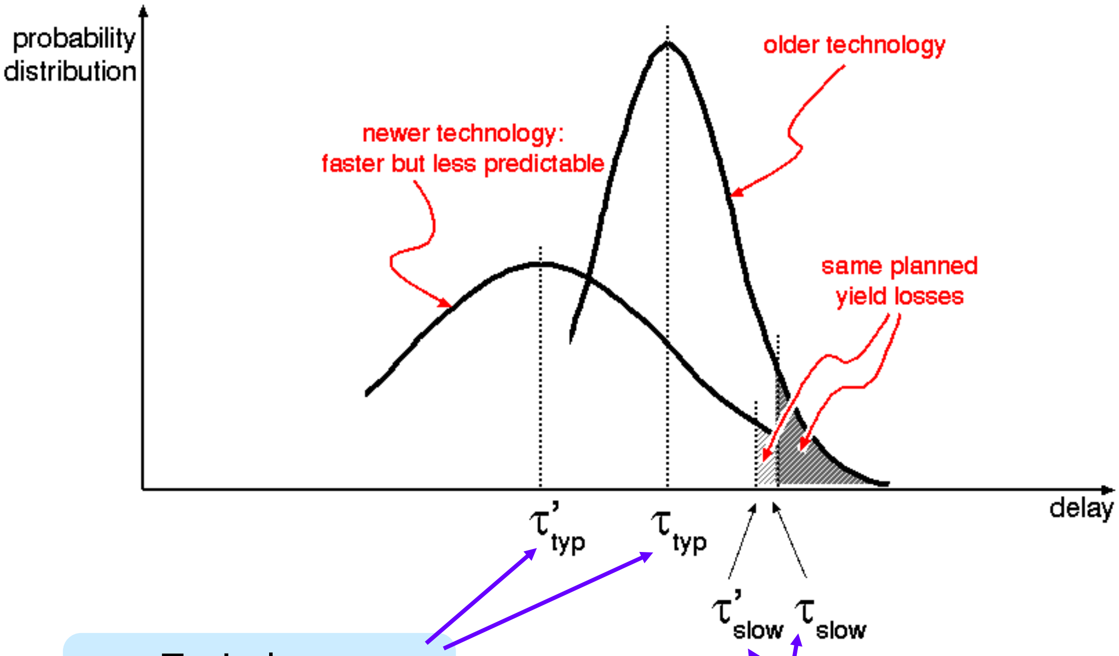
V_T = 0.85 V



Source: Asenov et al., 2003 © IEEE

Significant **widening** of the statistical parameter spread

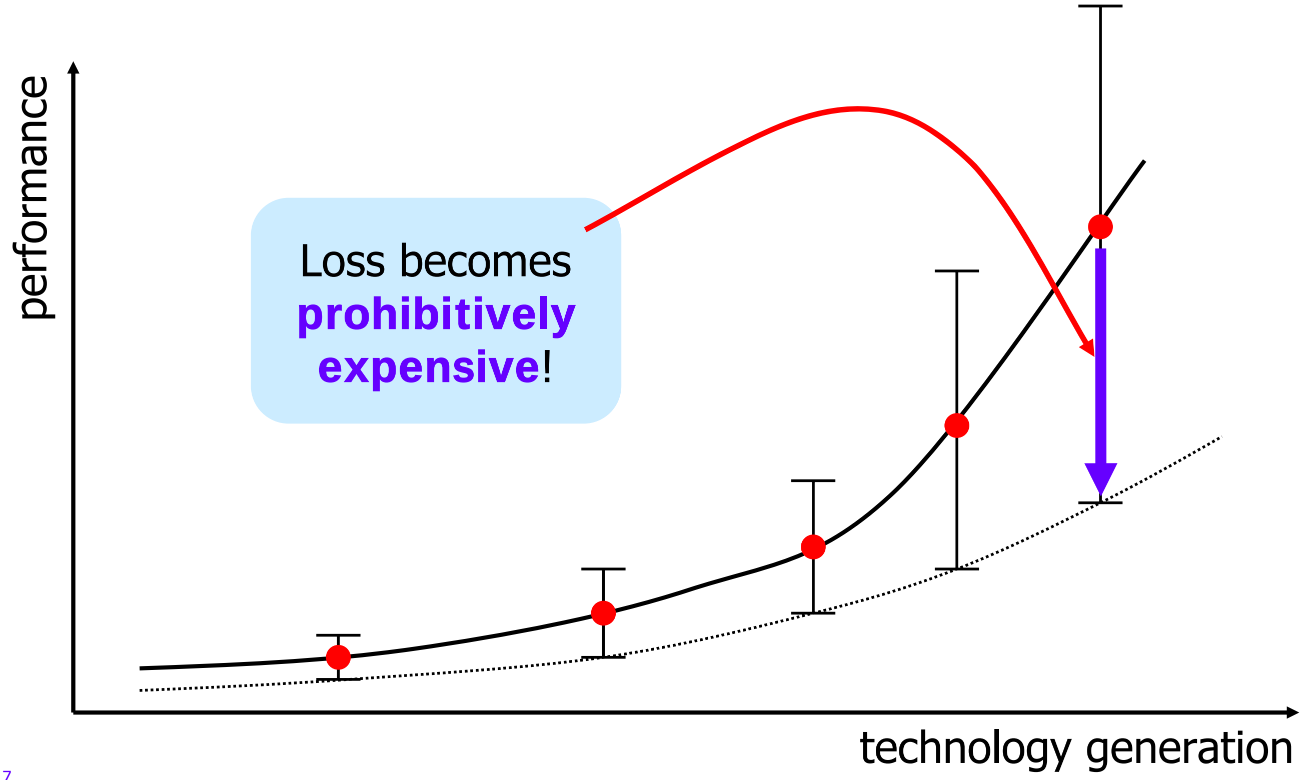
How Things Are Actually Evolving



Typical case improves as usual…

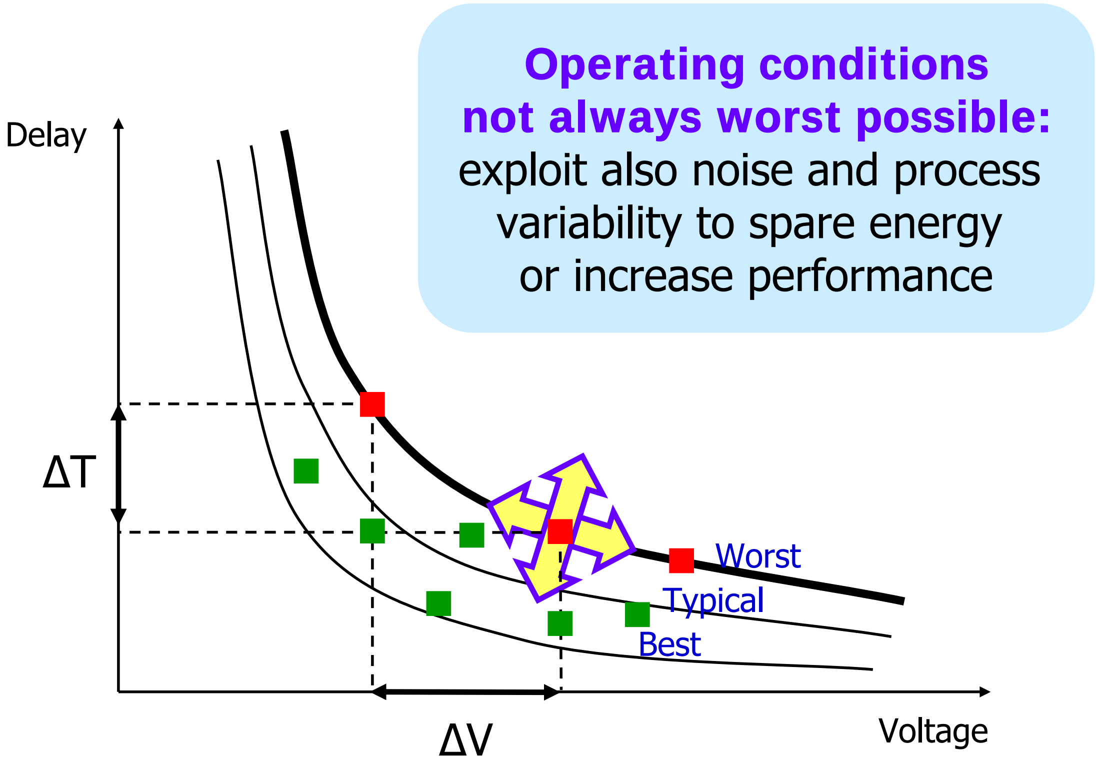
…but improvement of worst-case becomes **marginal**!

Worst-Case Design Gets Increasingly Expensive



Loss becomes **prohibitively expensive**!

Adaptive Design Space Exploration

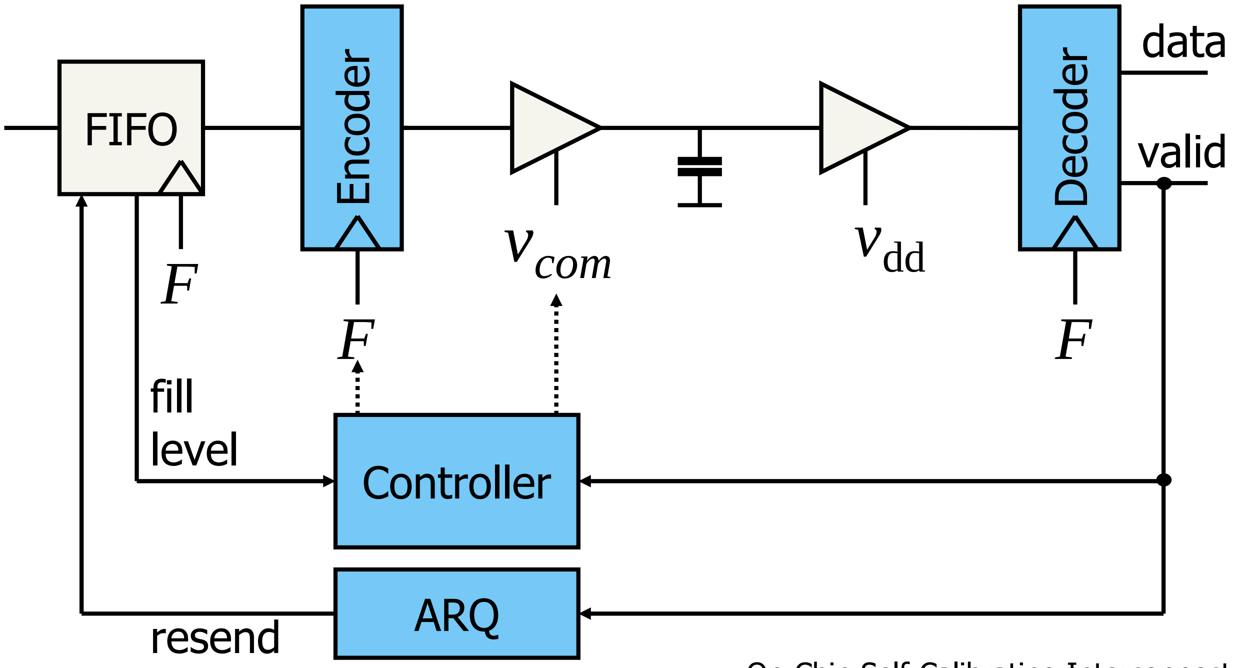


Operating conditions not always worst possible: exploit also noise and process variability to spare energy or increase performance

Outline

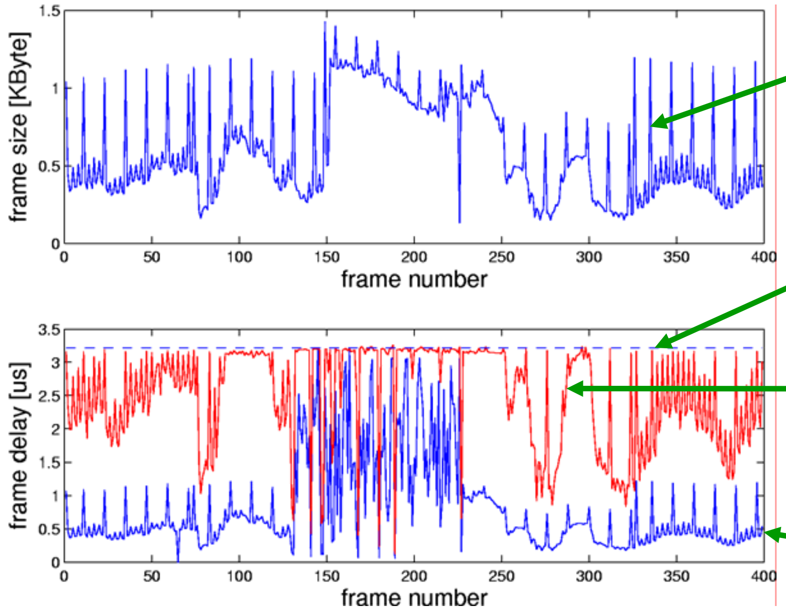
- **Self-calibrating link for on-chip networks**
- Ingredients of self-calibrating designs
  - Detection Schemes
  - (Correction Schemes)
  - Controller Design
- Conclusions

Self-Calibrating Link



On-Chip Self-Calibrating Interconnects
(Worm, Thiran, Ienne, and De Micheli,
ISSS 2002 and TVLSI 2005)

Reduced Energy Consumption



variable workload

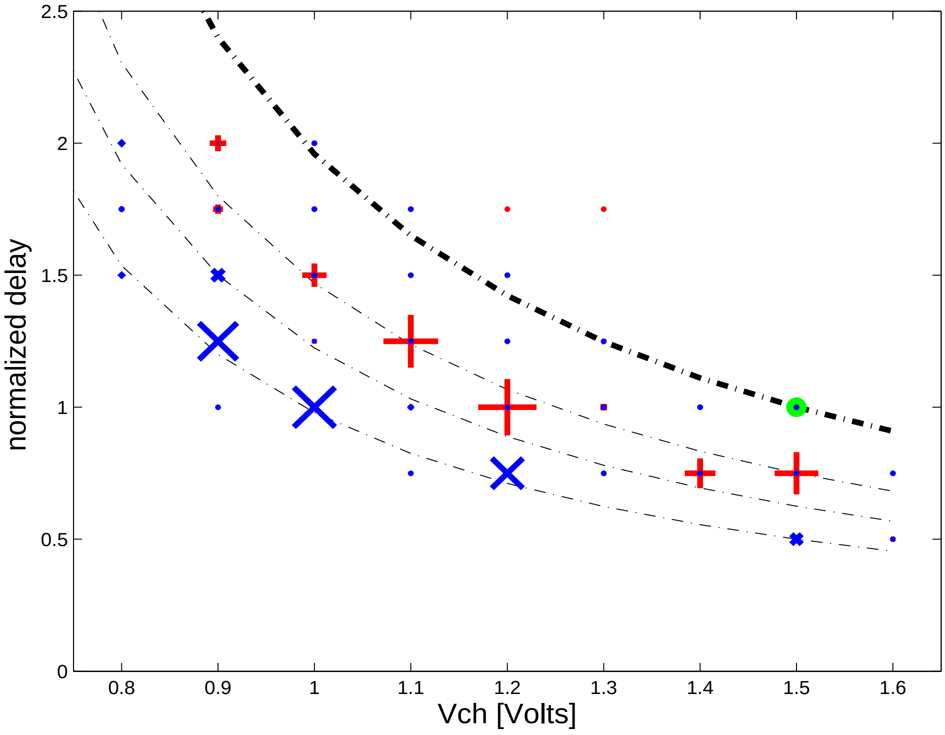
delay constraint

self-calibrating system attempts to be always as slow as possible

classic system is often unnecessarily fast

Energy saving of 53 %

Self-Calibration Robustness to Process Variation



classic a priori: ●

$$\begin{cases}\mu_{f_{cut}} = 500MHz \\ \sigma_{f_{cut}} = 36MHz\end{cases}$$

good chip: ✕

$$\begin{cases}\mu_{f_{cut}} = 570MHz \\ \sigma_{f_{cut}} = 20MHz\end{cases}$$

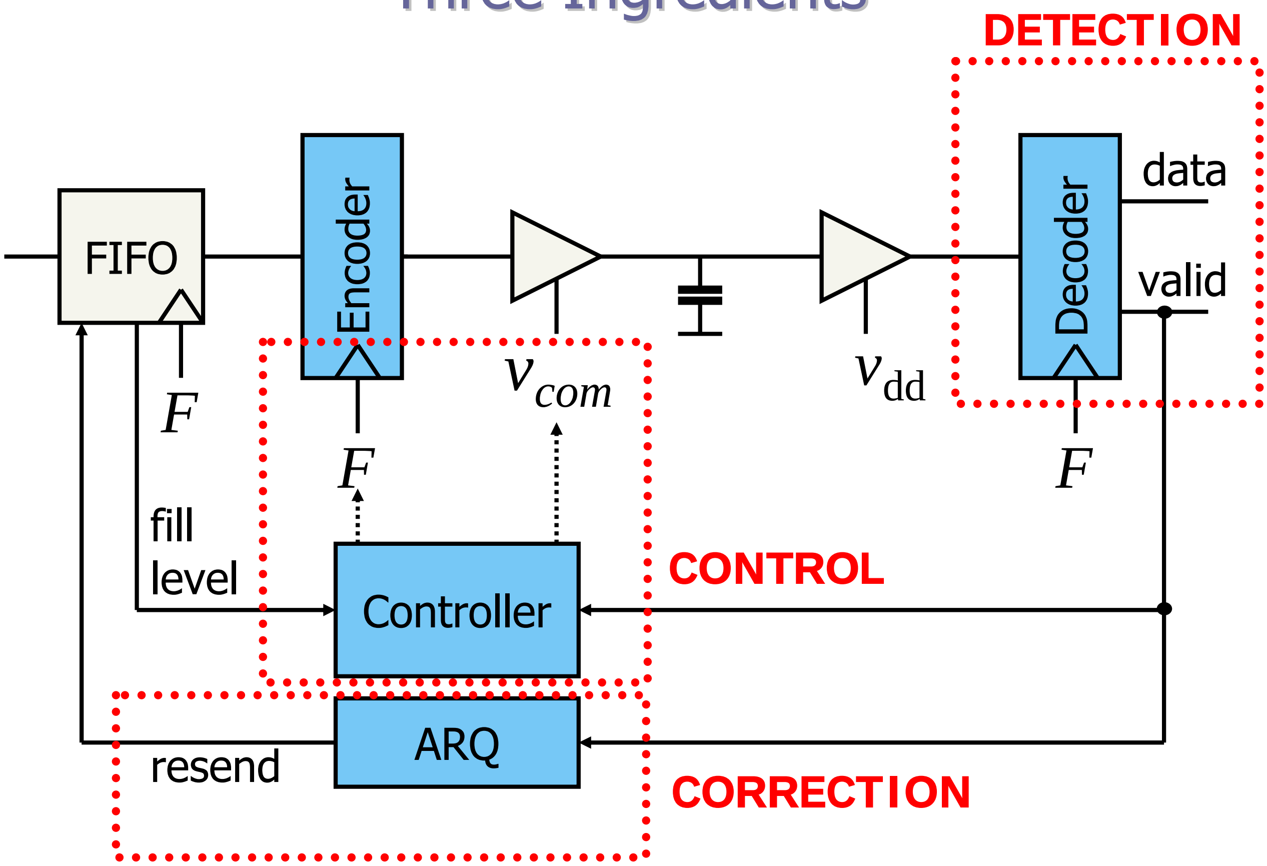
bad chip: ✚

$$\begin{cases}\mu_{f_{cut}} = 430MHz \\ \sigma_{f_{cut}} = 20MHz\end{cases}$$

Outline

- Self-calibrating link for on-chip networks
- **Ingredients of self-calibrating designs**
  - Detection Schemes
  - (Correction Schemes)
  - Controller Design
- Conclusions

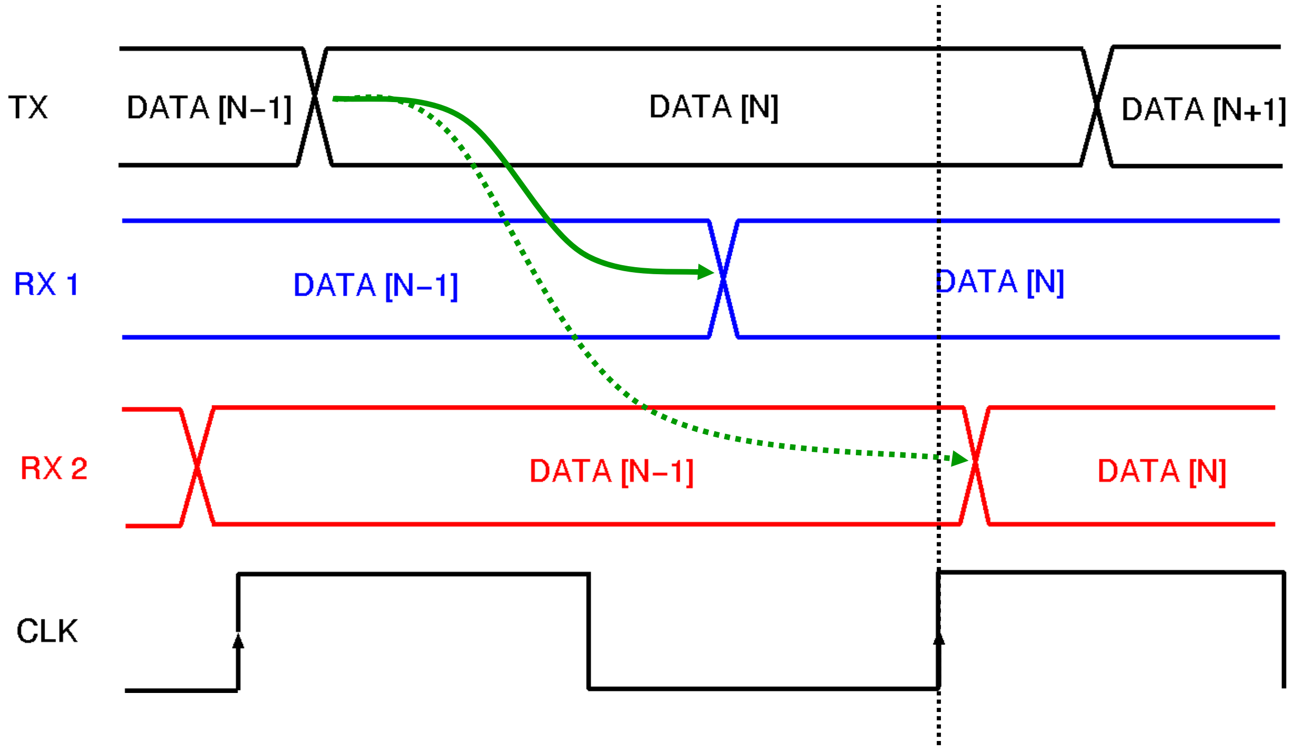
Three Ingredients



Outline

- Self-calibrating link for on-chip networks
- Ingredients of self-calibrating designs
  - **Detection Schemes**
  - (Correction Schemes)
  - Controller Design
- Conclusions

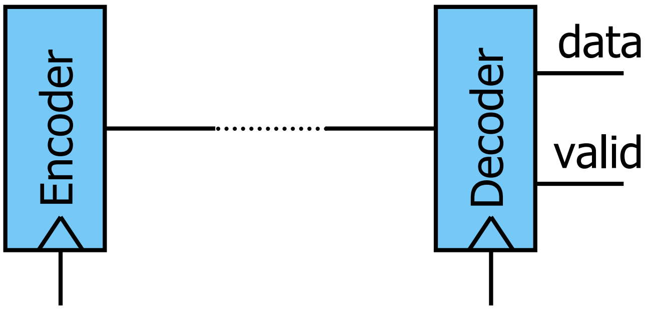
Detecting Data Readiness



Two Ways to Go for Detection

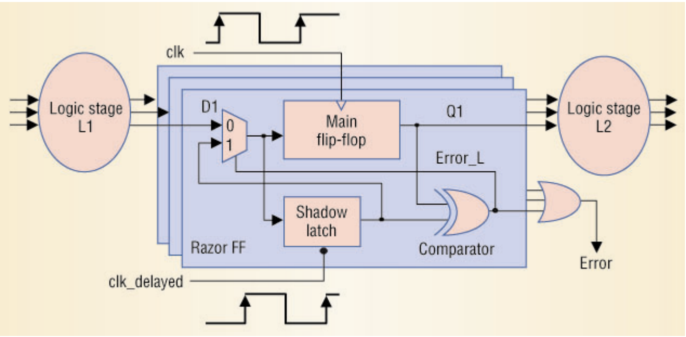
- **Coding**

Soft Self-Synchronising Codes for Self-Calibrating Communication
(Worm, Ienne, Thiran, ICCAD 2004)



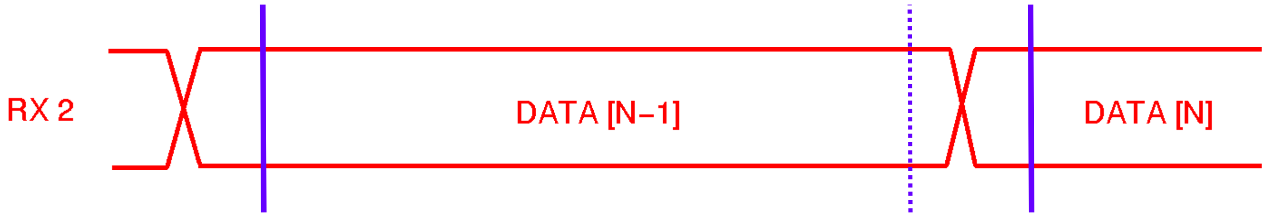
- **Shadow latches**

RAZOR: A Low-Power Pipeline Based on Circuit-Level Timing Speculation
(Blaauw, Austin, Flautner, Mudge, et al., MICRO 2003)

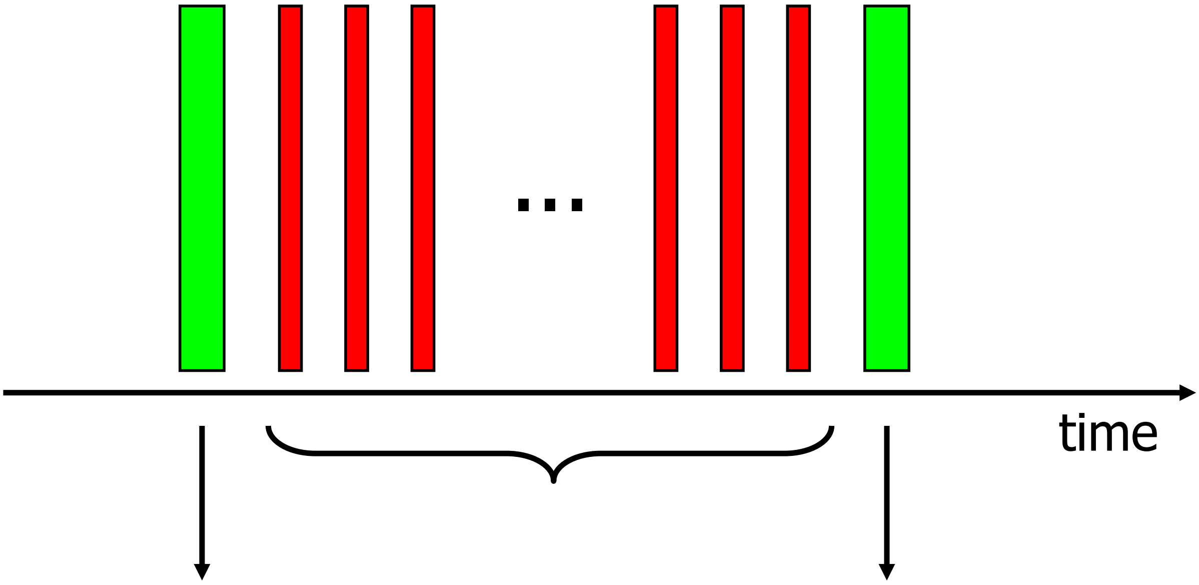


Classic Error Detecting Codes Are Not Suitable as Such

- Classic error detecting codes (e.g., CRCs) are targeted to the detection of additive errors
- When timing is extremely aggressive, the channel output is twice the same data (no time to transition)
- But, each piece of data output being a codeword, the decoder, when receiving the last piece of data again, considers it correct!

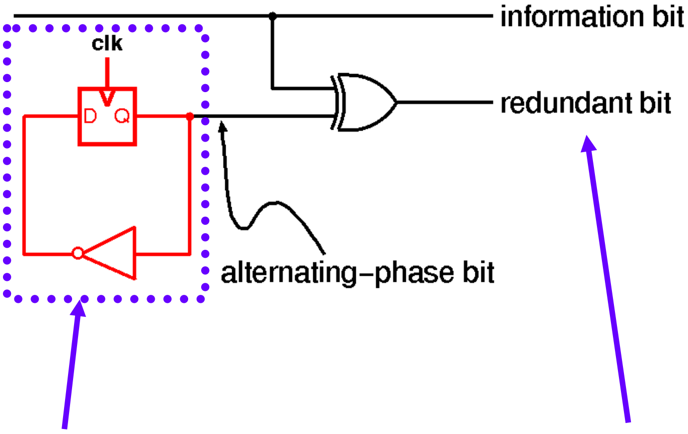


Self-Synchronizing Codes Are What We Need



$c_k \in C$ $\notin C \; (\varepsilon_{\mathrm{w}}^{\mathrm{res}} = 0)$ $c_{k+1} \in C$

Level-Encoded Dual-Rail Is an (Expensive) Solution

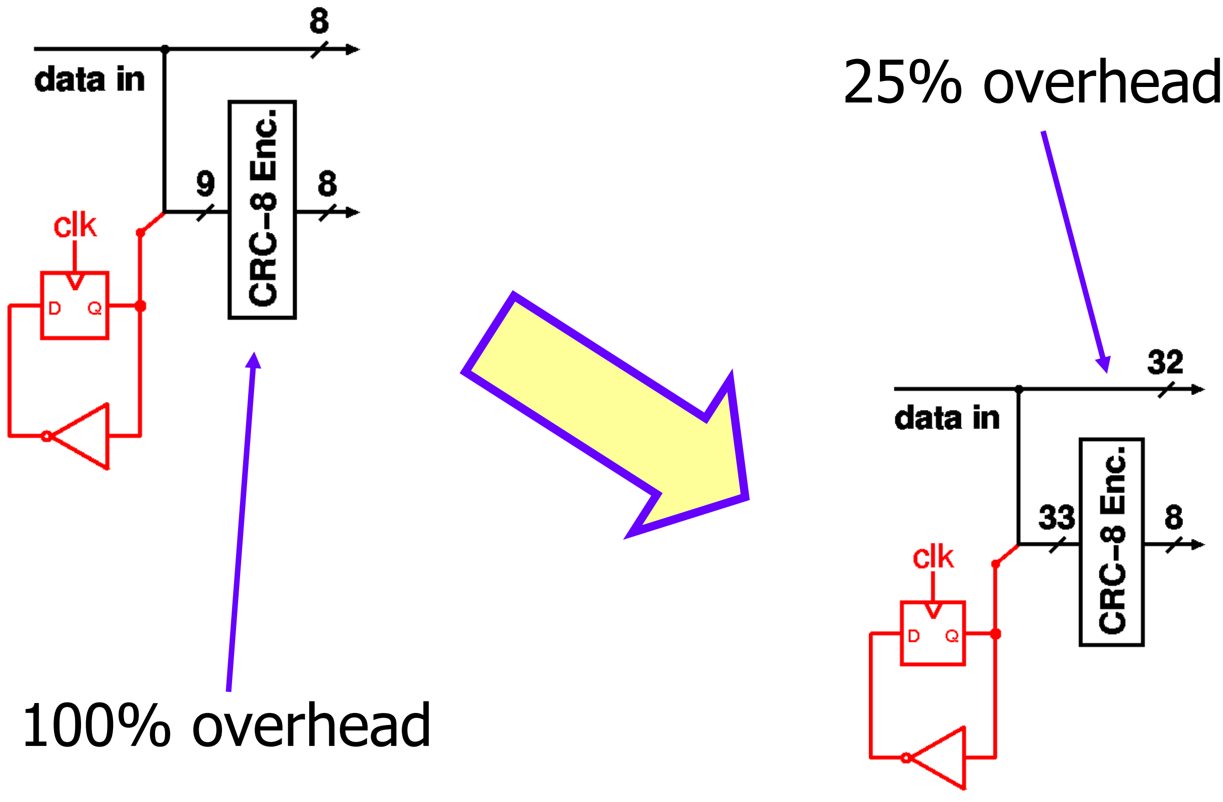


Dean et al., 1991

“changes dictionary” at each cycle

100% overhead

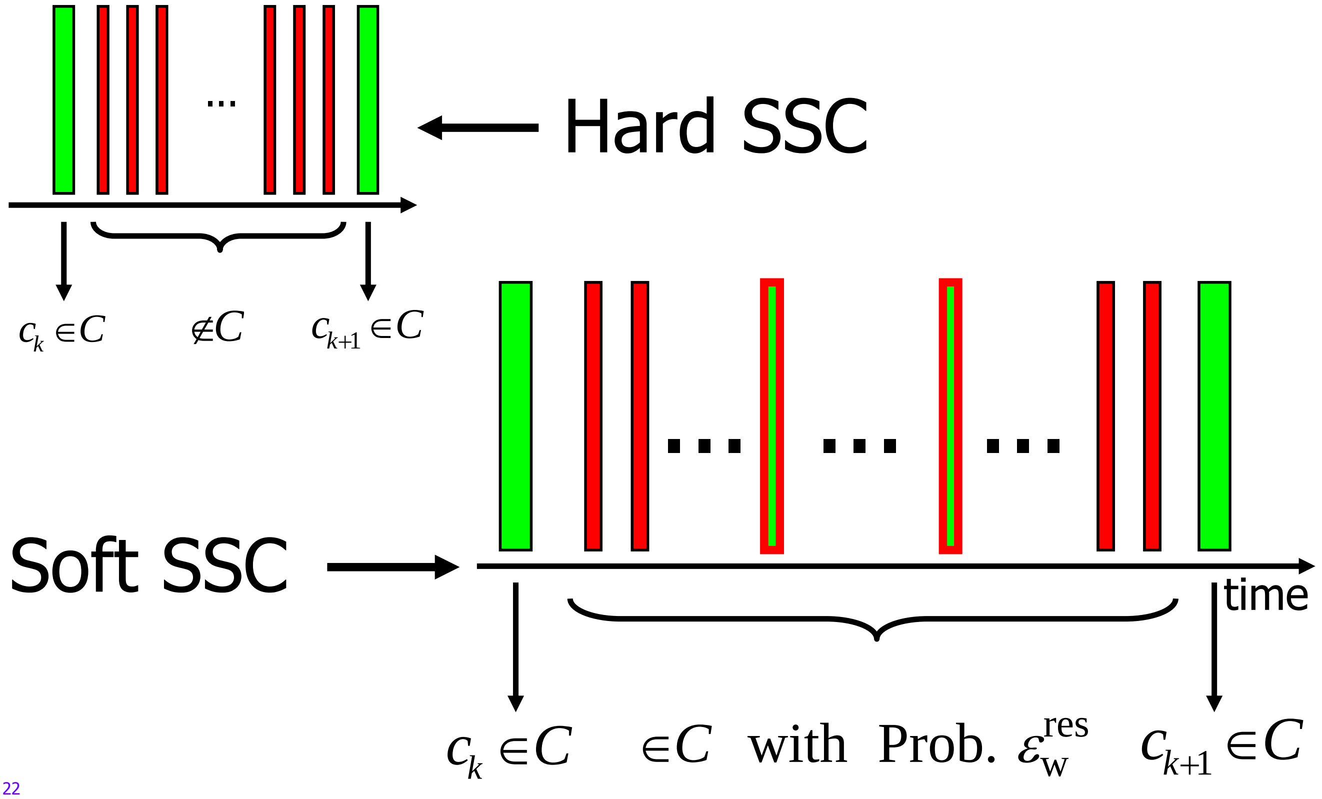
Alternating-Phase Codes



25% overhead

100% overhead

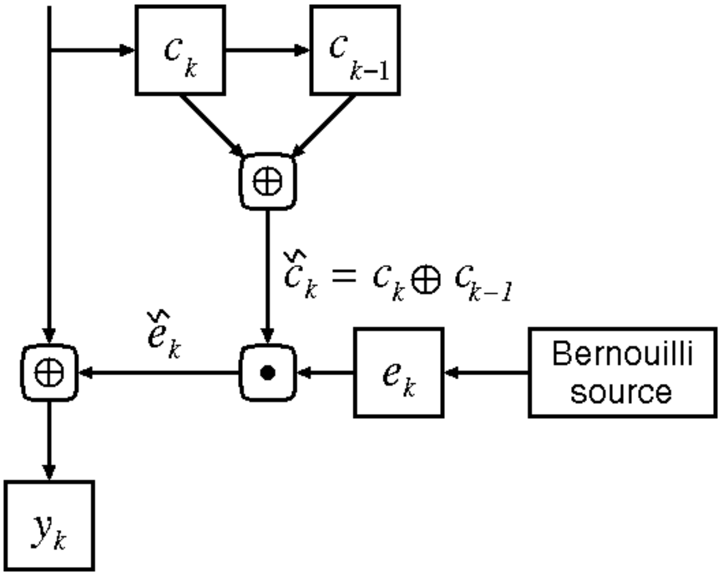
Soft Self-Synchronising Codes



Hard SSC

Soft SSC

Modelling Timing Errors



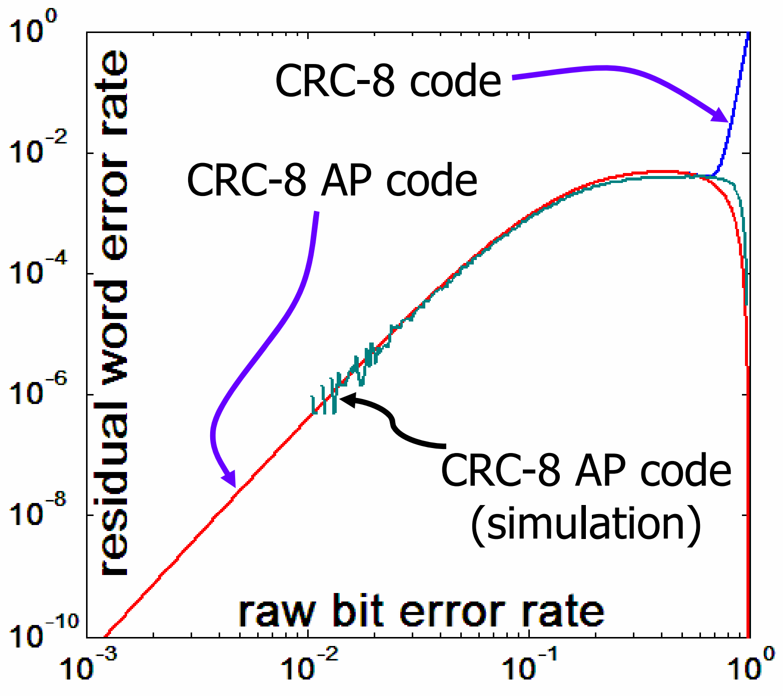
A bit error occurs if and only if $\tilde{e}_k = 1$, i.e., if and only if:

- a transition occurs

and,

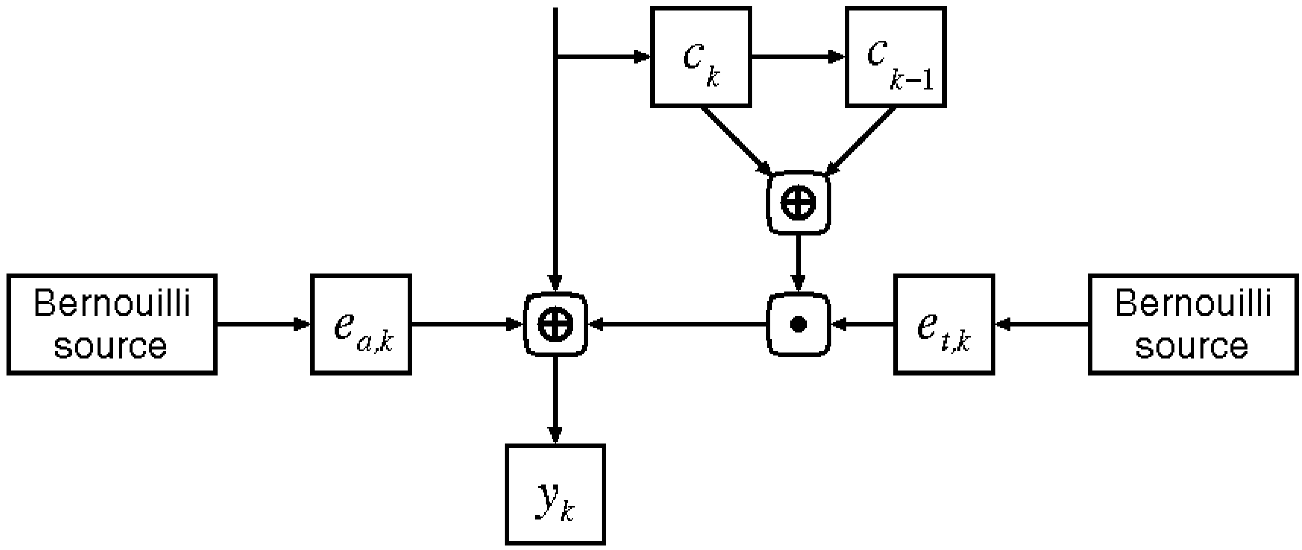
- the transition fails

Residual Word Error Rate of CRC-8 Alternating-Phase Code

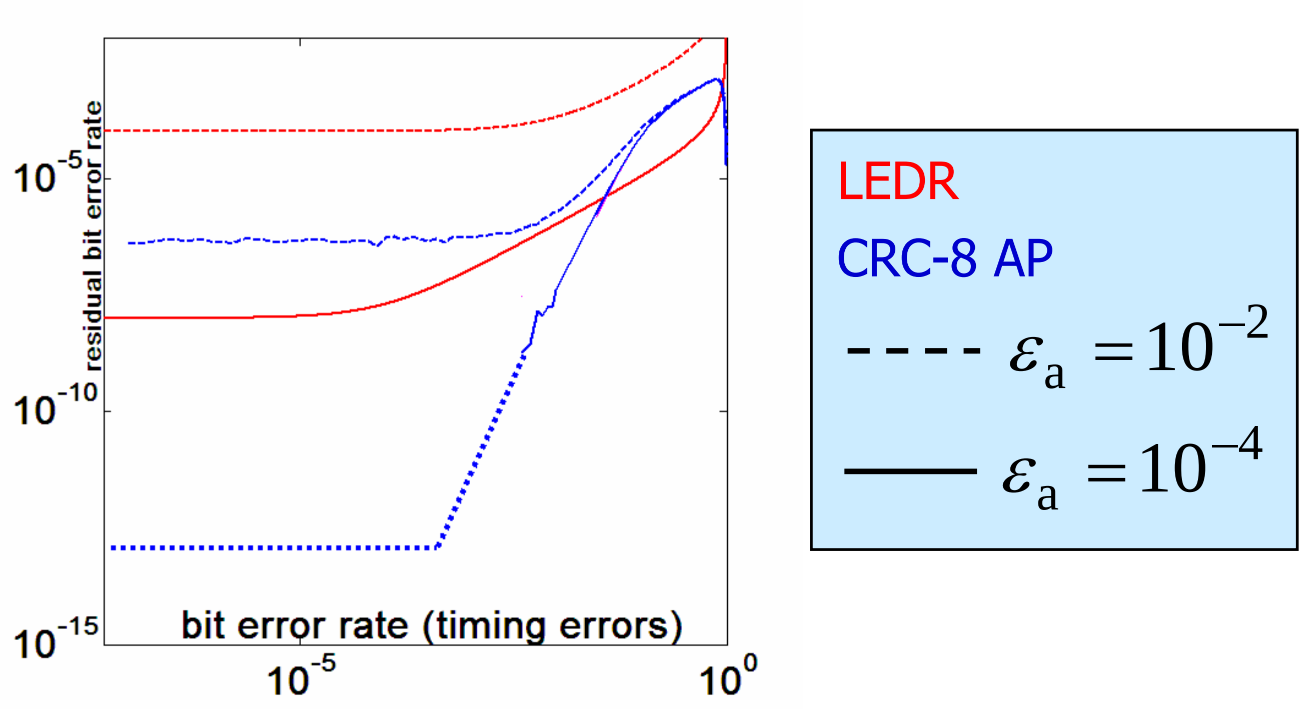


What if Also Additive Noise?

- A more realistic situation



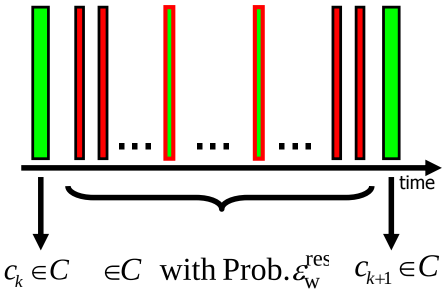
Additive and Timing Errors



- Hard Self-Synchronising Codes are no longer “perfect”
- CRC-based Alternating-Phase Codes can be better

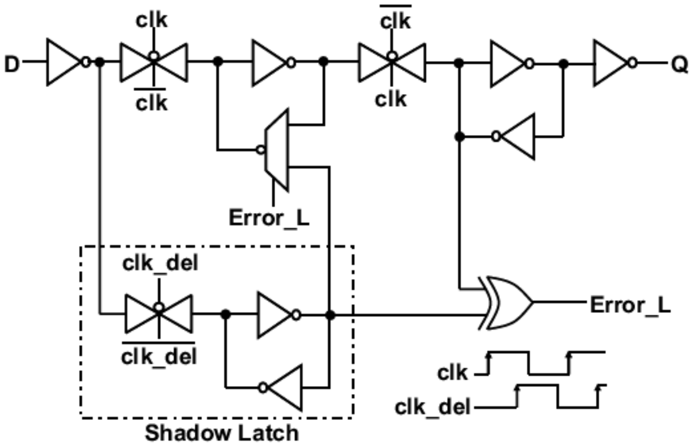
Conclusion on Coding

- There exist **cheaper** encoding possibilities than classic self-synchronizing codes
- Additive noise is even better handled by AP codes:
  - For instance, CRC-8 AP codes outperforms LEDR with 1/4th of the redundancy overhead
- Reliability can be controlled as in any coding problem and is actually better than what our model measures



- Major drawback: **not general!** We don't know how to generate codes for a nonidentity function…

Shadow Latches



Source: Kaul et al., 2005 © IEEE

- Simple but brilliant idea: resample the signal after a **safe** delay and see if it is the same—if so, the value was correct
- **General!**

Conclusions on Shadow Latches

- **Cheap** and **general** mean of detecting readiness!
- Need to satisfy several **constraints**
  - Maximum worst-case propagation time should not be above the shadow latch sampling time, or **errors will go undetected**
  - Minimum best-case propagation time should not arrive to the shadow latch before its sampling time, or **correct values will be mistakenly reported as errors**

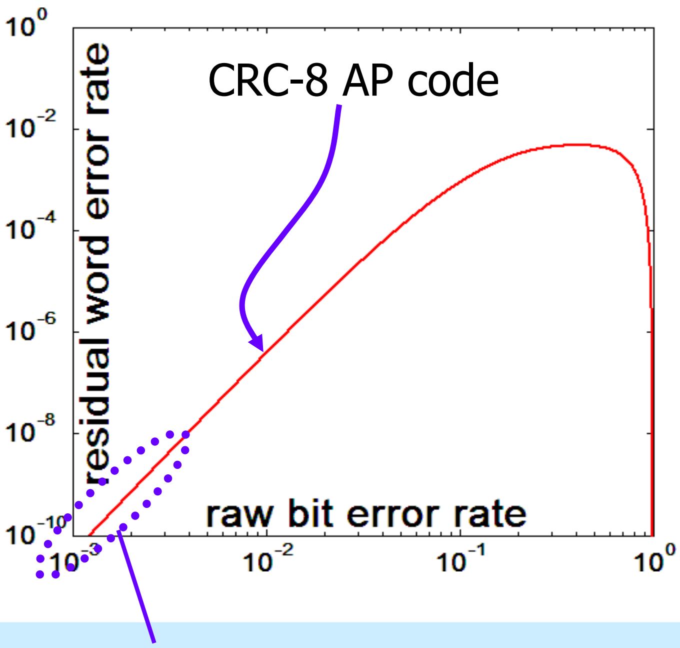
Outline

- Self-calibrating link for on-chip networks
- Ingredients of self-calibrating designs
  - Detection Schemes
  - (Correction Schemes)
  - **Controller Design**
- Conclusions

Typical Control Problem

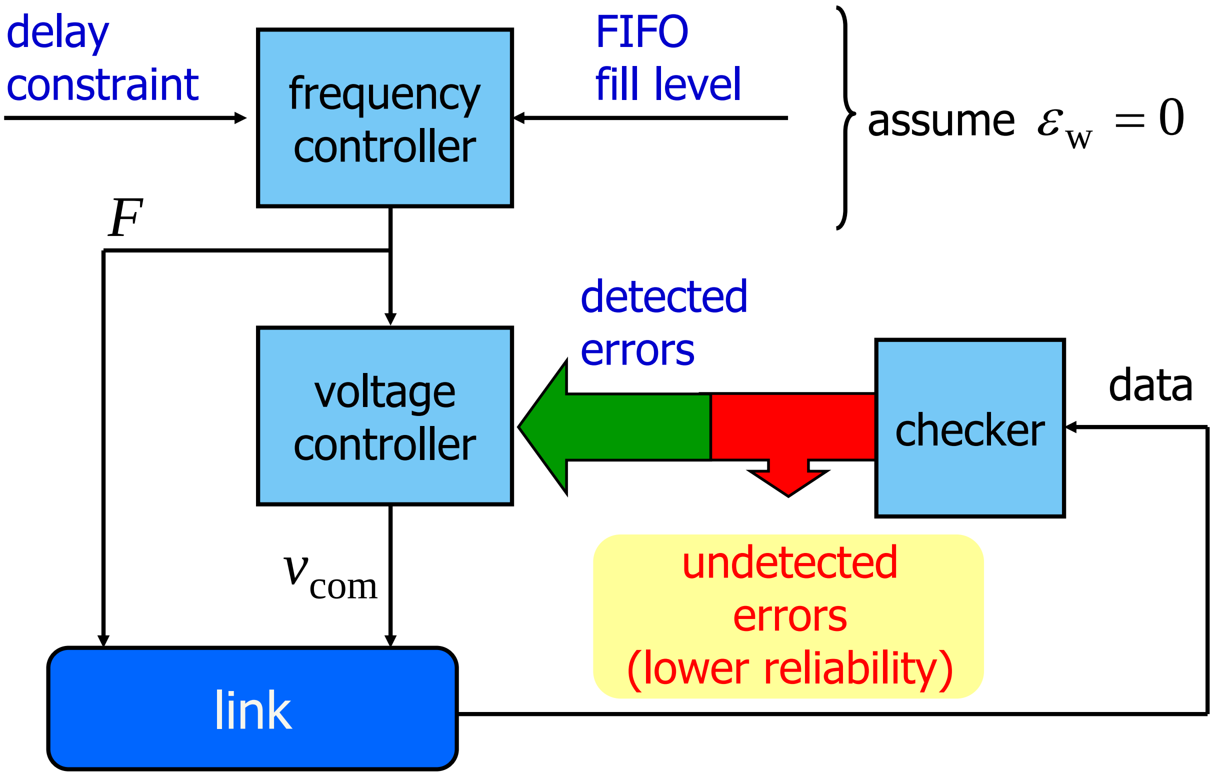
- User constraints: reliability and performance
- (**Problem**) Choose frequency and supply voltage such that:
  - Energy per information bit is minimized
  - Time required to send a word is below some user specified level
  - Residual word error rate is below some user specified level
- Remark: all variables to control have complex nonlinear dependences on problem inputs
  - Error rate depends on frequency and supply
  - Residual errors depend on raw error rate
  - Number of retransmission depends on error rate
  - Energy depends on number of retransmissions
  - …

Raw Word Error Rate Is Negligibly Low

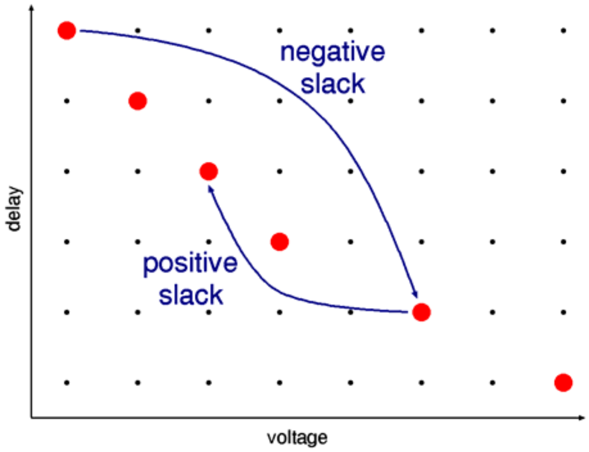


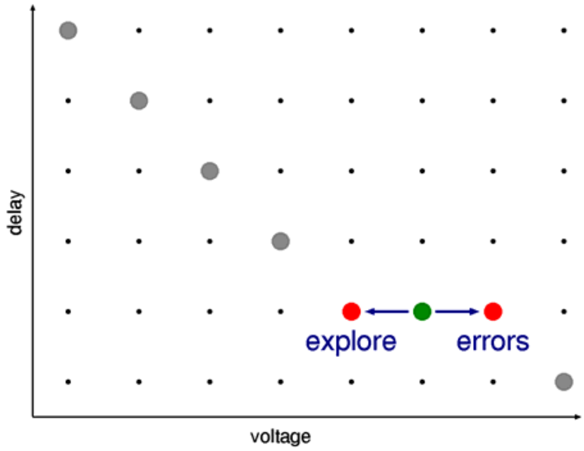
Any realistic residual error rates impose raw bit error rates which are **negligible for performance**

Decoupled Voltage and Frequency Control



Separation of Concerns Enables Cheap Controllers

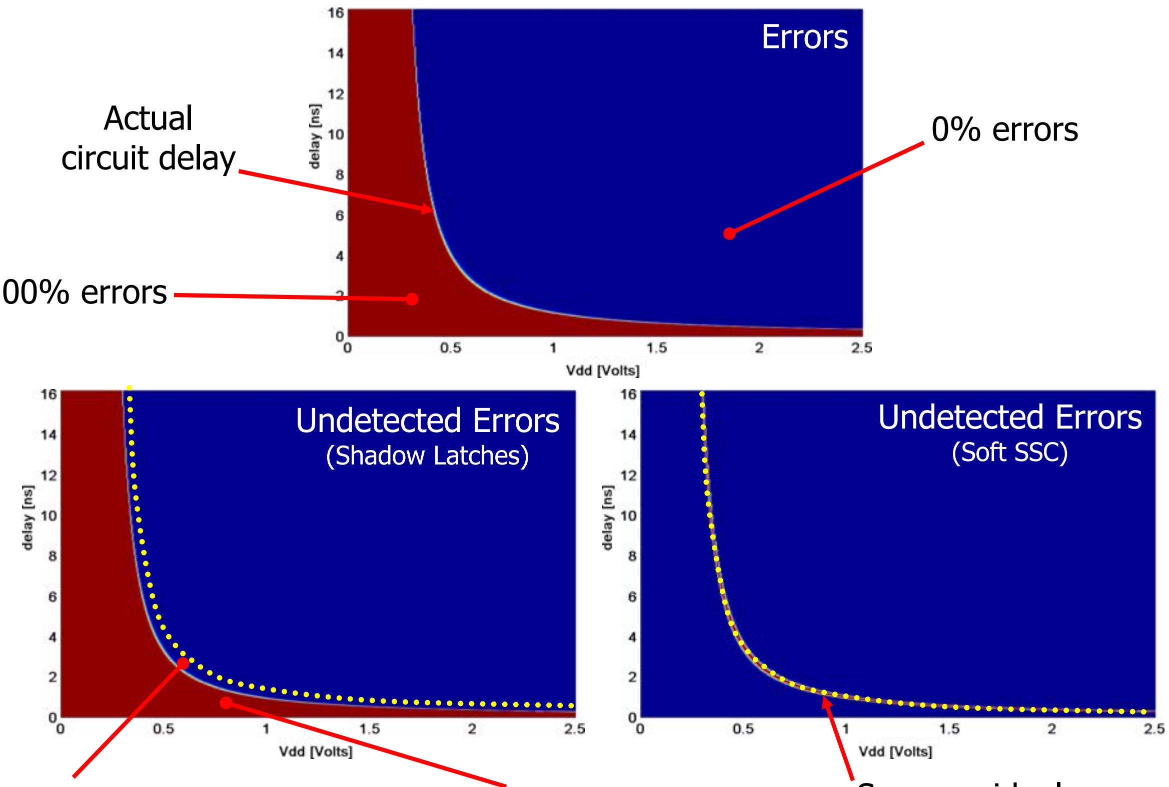




Higher level controller imposes delay among measured Pareto points

Lower level controller avoid errors and collects Pareto points

Errors and Undetected Errors



Actual circuit delay

0% errors

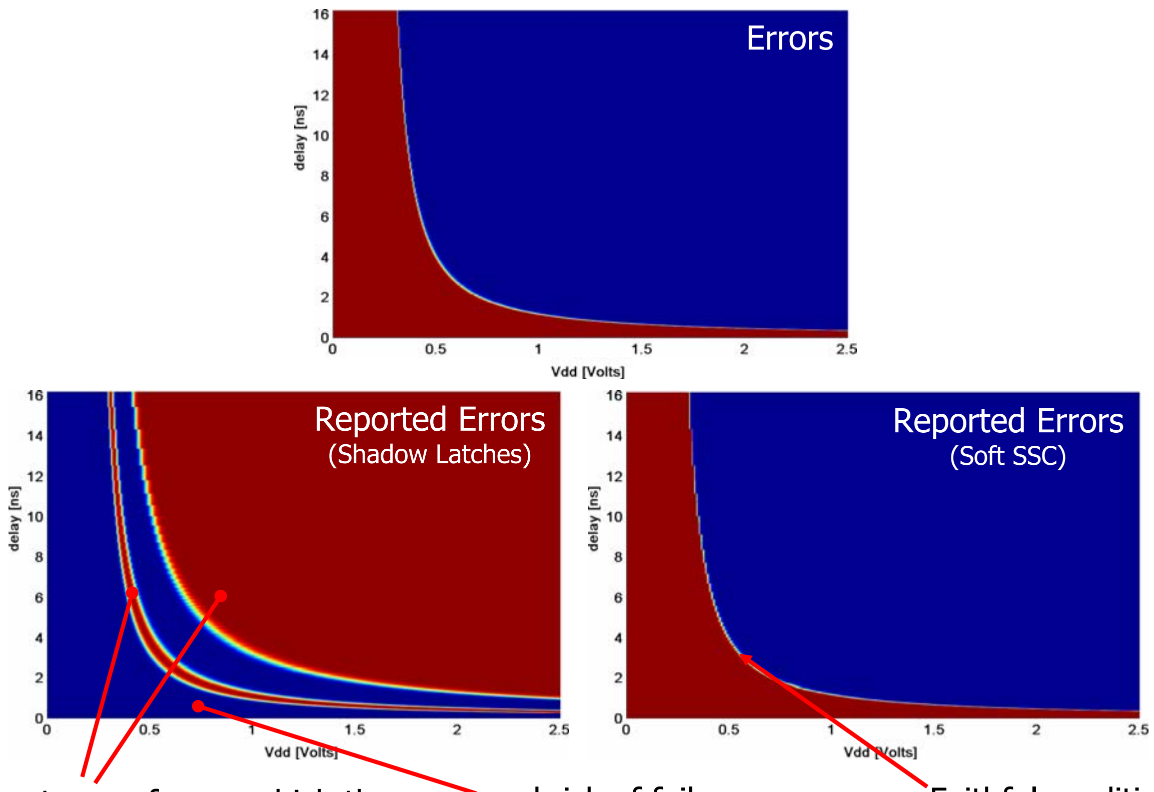
100% errors

No residual errors immediately close to actual delay…

…but 100% residual errors close by!

Some residual errors around the actual delay

But What Information to the Controller?

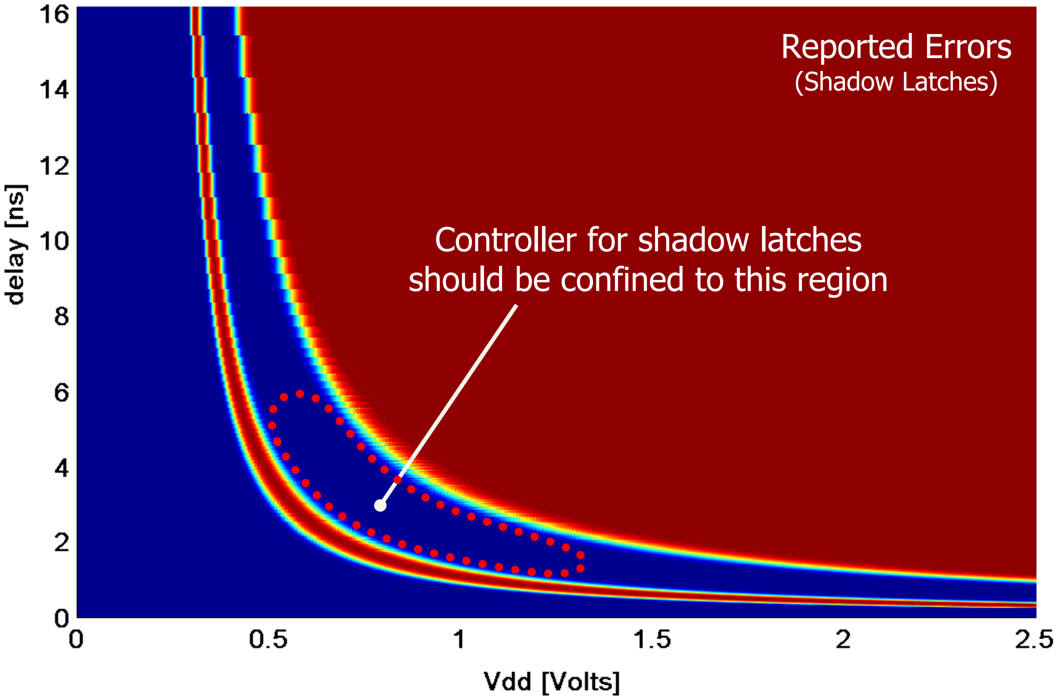


Two types of error which the controller cannot discriminate!...

…and risk of failure close by!

Faithful rendition of the error map

Dependence from the Error Model

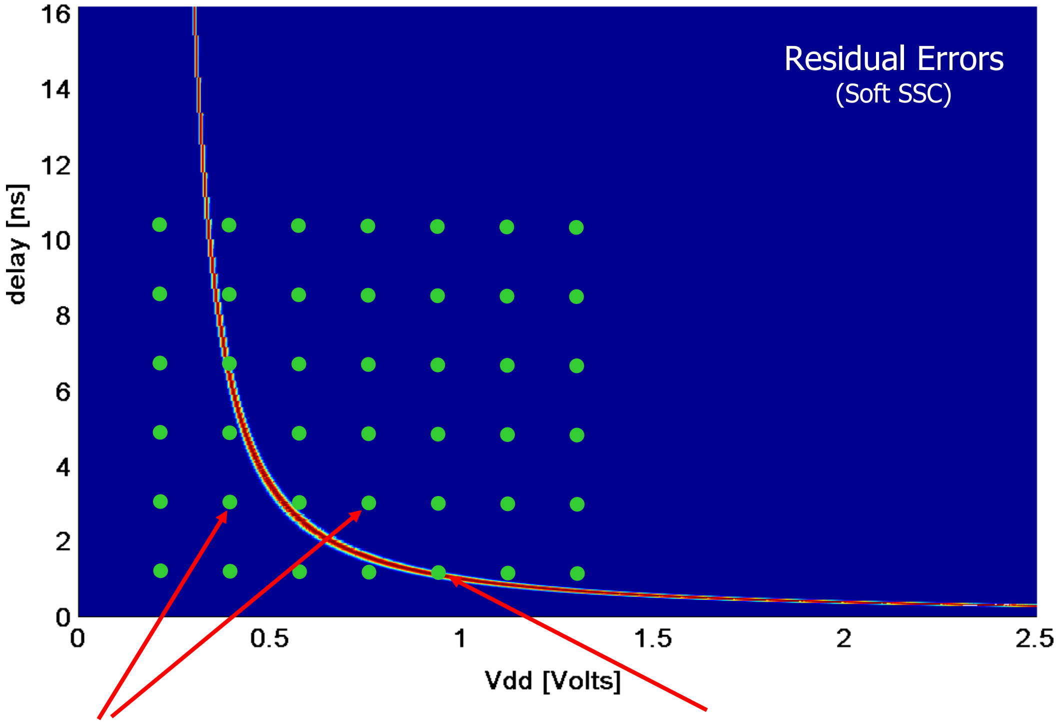


Reported Errors
(Shadow Latches)

Controller for shadow latches should be confined to this region

What if the error model is **incorrect**?

Error Model Should Not Be a Concern to the Controller



Residual Errors
(Soft SSC)

Controller for soft SSC
can explore the whole space

Low probability of residual errors
must be guaranteed statistically

Conclusions

- Self-calibrating designs may become essential to achieve returns on technological investments
- The main challenge is to **find reliable detection mechanisms which do not restrict controllers**
  - Coding is well behaved but known only for identity function—how can it be generalised?
    - Similar to completion detection in the asynchronous world but here no glitch free constraint—isn’t it much easier?
    - Prior work on parity checking for functional units, but are the techniques general and strong enough?!—we must handle error rates ~1!
  - Shadow latches are immediately available for any circuit but restrict controllers and can be challenging—what if our error models are wrong?

References

- Frédéric Worm, Patrick Thiran, Giovanni De Micheli, and Paolo Ienne. *Self-calibrating Networks-on-Chip.* In **Proceedings of the IEEE International Symposium on Circuits and Systems**, Kobe, Japan, May 2005.
- Frédéric Worm, Patrick Thiran, and Paolo Ienne. *A unified coding framework for delay-insensitivity.* In **Proceedings of the 11th International Symposium on Asynchronous Circuits and Systems**, New York, March 2005.
- Frédéric Worm, Paolo Ienne, Patrick Thiran, and Giovanni De Micheli. *A robust self-calibrating transmission scheme for on-chip networks.* **IEEE Transactions on Very Large Scale Integration (VLSI)**, 13(1), January 2005.
- Frédéric Worm, Paolo Ienne, Patrick Thiran, and Giovanni De Micheli. *On-Chip self-calibrating communication techniques robust to electrical parameter variations.* **IEEE Design and Test of Computers**, 21(6), November-December 2004.
- Frédéric Worm, Paolo Ienne, and Patrick Thiran. *Soft self-synchronising codes for self-calibrating communication.* In **Proceedings of the International Conference on Computer Aided Design**, San Jose, Calif., November 2004.
- Frédéric Worm, Paolo Ienne, Patrick Thiran, and Giovanni De Micheli. *An adaptive low-power transmission scheme for on-chip networks.* In **Proceedings of the 15th International Symposium on System Synthesis**, Kyoto, October 2002.