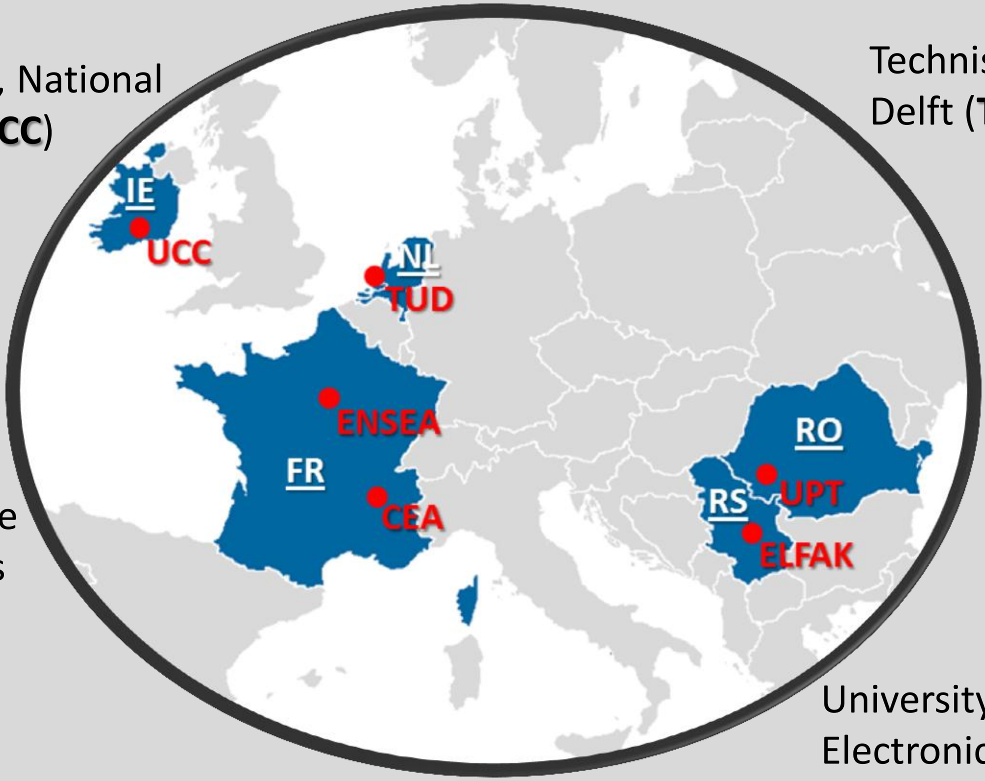
Innovative Reliable Chip Designs from Low-Powered Unreliable Components

Valentin Savin, CEA-LETI, Minatec, France
Sorin Cotofana, TU-Delft, The Netherlands

i-RISC Consortium

University College Cork, National University of Ireland (**UCC**)

Technische Universiteit Delft (**TUD**)



Ecole Nationale Supérieure de l’Electronique et de ses Applications (**ENSEA**)

Universitatea Politehnica Timisoara (**UPT**)

University of Nis, Faculty of Electronic Engineering (**ELFAK**)

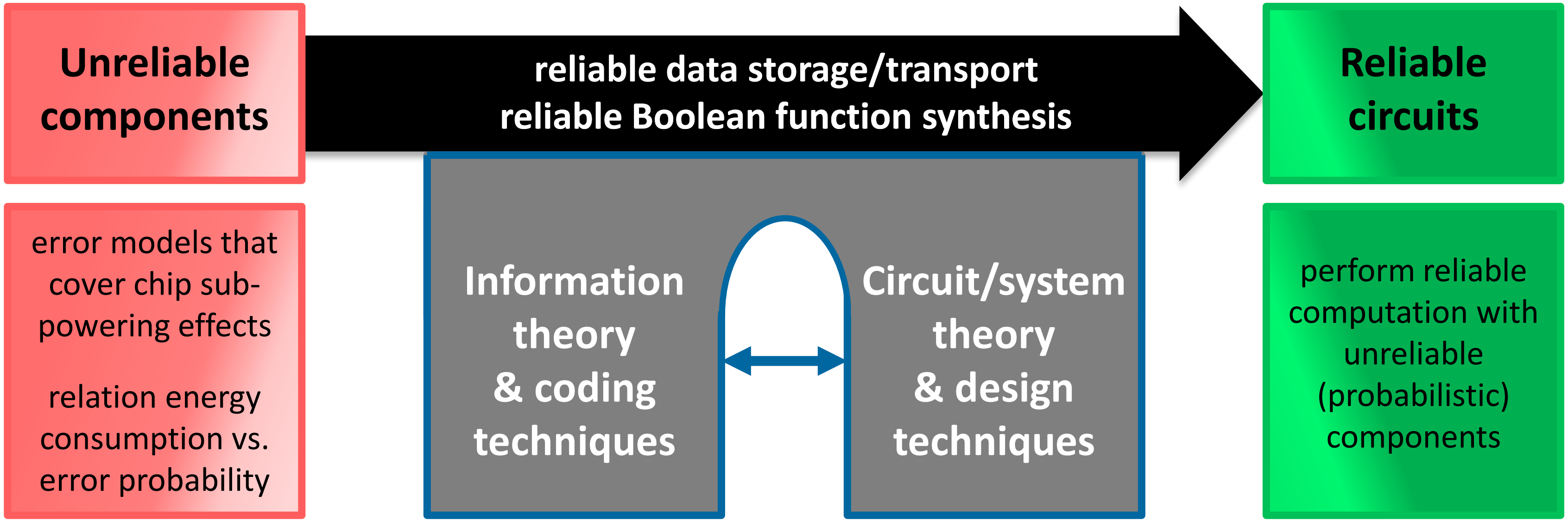
Commissariat à l’Energie Atomique et aux Energies Alternatives (**CEA-LETI**)

FP7-ICT / FET-OPEN
02/2013 – 01/2016

Low-powered unreliable components

- **Emerging nanoelectronic devices**
  - density integration increases, variations in technological process
  - lower supply voltages (reduction by 20% per technology node)
  - fault tolerance, and in particular soft-error correction, became one of the Top-5 overall design technology challenges of the ITRS (2010)
- **Sustainability concerns:** lower the energy consumption
  - aggressive supply voltage scaling
  - bringing the signal level closer to the noise level reduces noise immunity and leads to unreliable computing
- **Unreliable components:** subject to transient errors
  - may be due to many different reasons: physical reasons (neutrons and alpha particles), timing errors, low noise margin, etc.
  - may be independent or not of the gate/circuit history
  - manifest themselves at particular time steps, but do not necessarily persist for later times ⇒ **probabilistic behavior**

i-RISC


Unreliable components
error models that cover chip sub-powering effects
relation energy consumption vs. error probability
reliable data storage/transport
reliable Boolean function synthesis
Information theory & coding techniques
Circuit/system theory & design techniques
Reliable circuits
perform reliable computation with unreliable (probabilistic) components

Synergistic utilization of:

- Information theory and coding techniques
  - traditionally utilized to improve the reliability of communication systems
  - exploit the large body of knowledge learned for reliable communication systems in the context of reliable computing systems ⇒ **fault-tolerant error correcting codes**
- Circuit and system theory and design techniques
  - design method able to combine fault-tolerant error correcting codes with the hardware they protect
  - multi-objective optimization of the design with respect to energy consumption, latency, and reliability

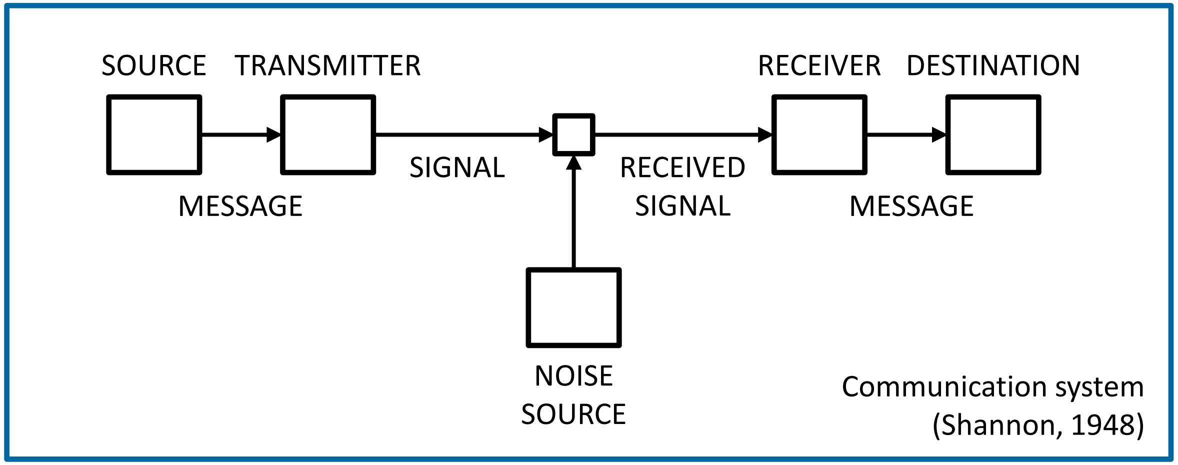
i-RISC

1. **Coding theory: from communication systems to computing devices**

   For both communications and computing the aim is to design reliable systems, while reducing the system power consumption

2. **Circuit design theory**

Coding theory: communication systems



Communication system (Shannon, 1948)

Reliable transmission is possible at any rate below the channel capacity:

$$C = \max\left(H(x) - H_y(x)\right)$$

Combat channel noise

- Increase signal power
- Use coding
  - add redundant information to the transmitted message, which can be exploited at the receiver side to correct transmission errors
  - transmission rate = useful data / transmitted data

maximize transmission rate ⇔ minimize redundancy

redundancy (ratio) = 1 / transmission rate = transmitted data / useful data

Coding theory: communication systems

- Repetition codes
  - poor compromise between reliability and redundancy
  - to make the bit error probability arbitrarily small, the transmission rate must go to zero (i.e. reliable transmission requires infinite redundancy)
- **Linear codes**
  - Transmitted message (vector of bits) belongs to a **linear codebook**: the set of solutions of a system of parity-check equations
    - **encoder**: responsible of computing the redundant information to be transmitted along with the useful data over the noisy channel
    - **decoder**: responsible with the error correction at the receiver side
  - **Shannon:** there exist linear codes that allow reliable communication at a transmission rate arbitrarily close to the channel capacity

parity codes

Goppa codes

Reed-Solomon codes

LDPC codes

BCH codes

Golay code

product codes

Hadamard codes

Reed-Muller codes

cyclic codes

AG codes

convolutional codes

Hamming codes

Turbo codes

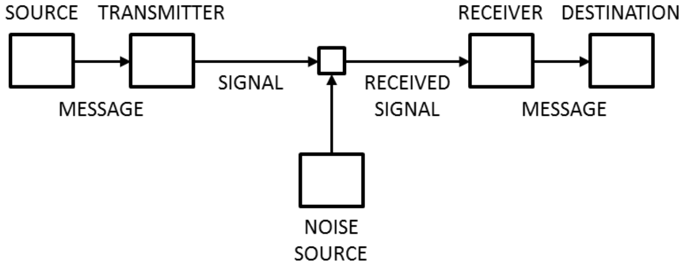
multi-dimensional parity codes

Justensen codes

Coding theory: communication systems

- Repetition codes
  - poor compromise between reliability and redundancy
  - to make the bit error probability arbitrarily small, the transmission rate must go to zero (i.e. reliable transmission requires infinite redundancy)
- **Linear codes**
  - Transmitted message (vector of bits) belongs to a **linear codebook**: the set of solutions of a system of parity-check equations
    - **encoder**: responsible of computing the redundant information to be transmitted along with the useful data over the noisy channel
    - **decoder**: responsible with the error correction at the receiver side
  - **Shannon:** there exist linear codes that allow reliable communication at a transmission rate arbitrarily close to the channel capacity
  - **LDPC codes:**
    - provide error-correction performance very close to the channel capacity
    - iterative message-passing decoding with constant computational complexity per decoded bit (i.e. linear complexity in the code length)

Coding theory: from Shanon to Shanon



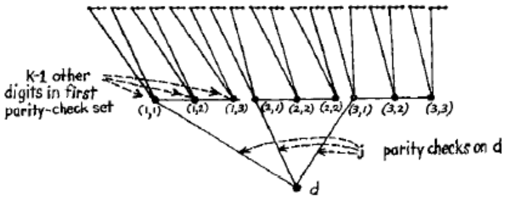
$$C = \mathrm{Max}\big(H(x) - H_y(x)\big)$$

Shannon Limit

Shannon, 1948

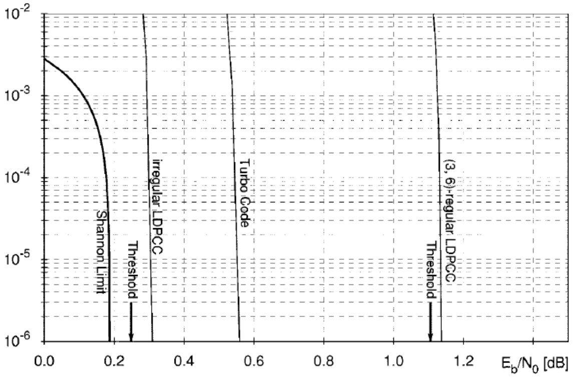
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1

Fig. 2—Example of a low-density code matrix; $N = 20, j = 3, k = 4$.



Invention of LDPC codes

Gallager, 1960



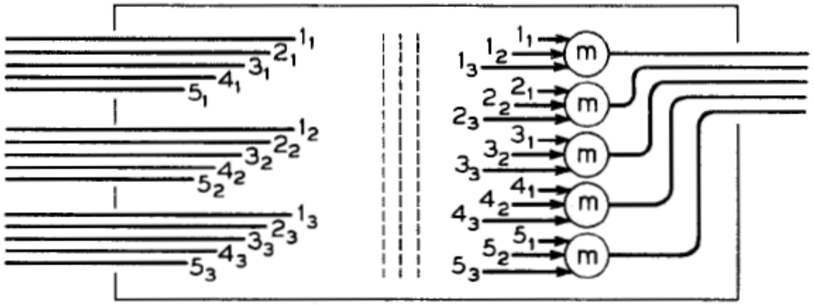
$$P_\ell = P_0 \otimes \lambda(\Gamma^{-1}(\rho(\Gamma(P_{\ell-1}))))$$

Design of LDPC codes closely approaching the Shanon Limit

Richardson, 2001

Reliable computing & storage

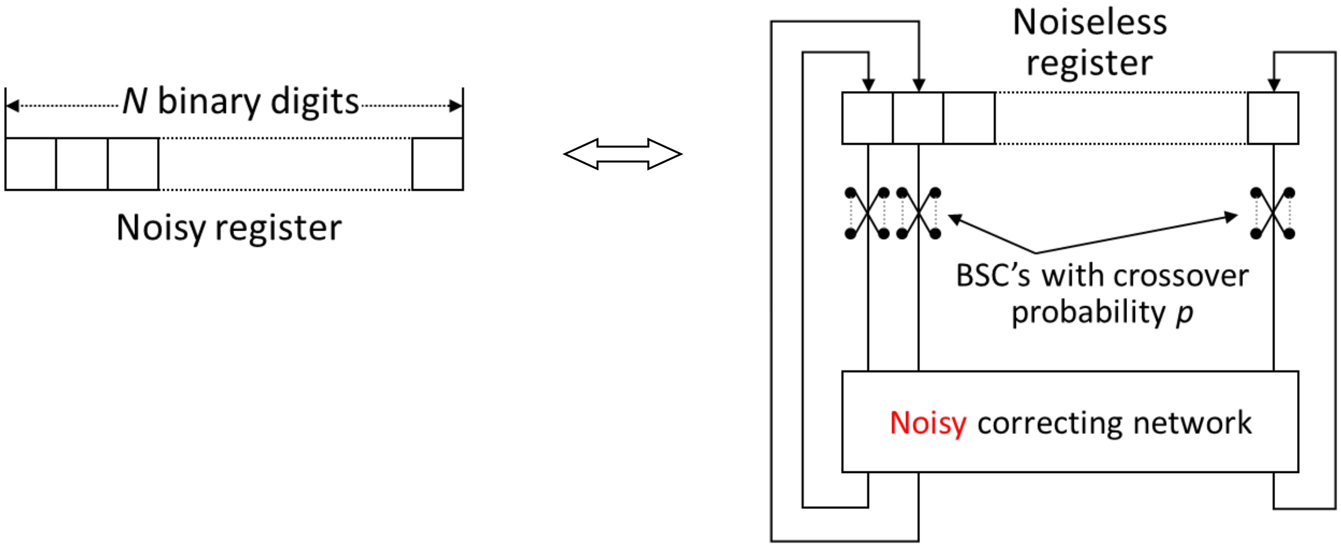
- **Von Neumann (1956):** multiplexing and majority voting
  - Similar to repetition coding
  - Simple, but poor compromise between redundancy and reliability
  - Error probability goes to zero only if redundancy (ratio) goes to infinity



- **Elias (1958):** first attempt to apply more general coding techniques to design reliable computing systems
  - Failed to design reliable computing systems with finite redundancy
  - Showed that finite redundancy can be achieved only if the complexity of both encoder & decoder grows at most linearly with the code length!
  - LDPC codes invented some years later by Gallager have this property ☺

Reliable computing & storage

- **Taylor (1968):** the first to investigate the capacity of storage systems built entirely from unreliable components
  - Results refined by Kuznetsov (1973) ⇒ Taylor-Kuznetsov (TK) model

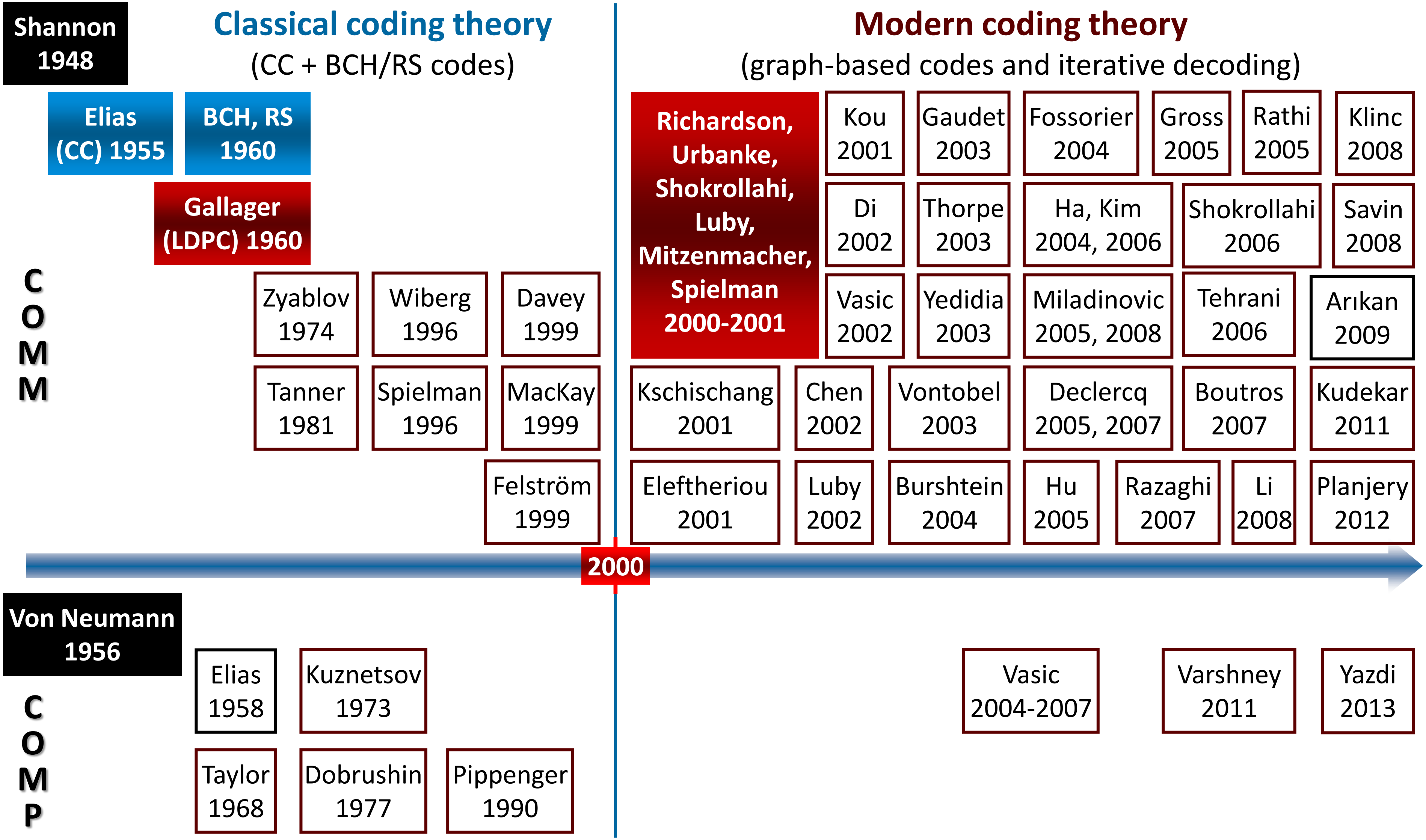


  - Information is stored in coded form in a register connected to a correcting circuit; values in the register are periodically updates acc. to some decoding scheme
  - Correcting circuit is also assumed to be built from unreliable components
  - Proved that **iterative LDPC decoding** (introduced a few years earlier by Gallager) can achieve **non-zero storage capacity**

Reliable computing & storage

- **Taylor (1968):** “A function computed by a system of *k* reliable components can also be computed by a system of O(*k* log(*k*)) unreliable components”
  - Heuristically argued by von Neumann in 1952
  - If an LDPC decoder is used for the correcting circuit, the theoretical log(*k*) factor comes from the asymptotical number of decoding iterations
  - Practical systems: only the computing components corresponding to a single decoding iteration have to be actually implemented in hardware (i.e. hardware redundancy can be replaced with temporal redundancy)
  - Moreover, practical systems use LDPC codes with finitely many decoding iterations, while preserving a very high reliability level
- **Dobrushin and Ortyukov (1977):** there exist Boolean functions for which the correcting circuit would necessarily require O(*k* log(*k*)) computations
  - Logarithmic redundancy is the best that can be done ☹
- **Pippenger (1990):** “*although some circuits of size k require size k* log(*k*) *when designed in a fault-tolerant style, **almost all don’t***”

Communication vs. Computing Systems



Shannon 1948

Classical coding theory

(CC + BCH/RS codes)

Modern coding theory

(graph-based codes and iterative decoding)

COMM

Elias (CC) 1955

BCH, RS 1960

Gallager (LDPC) 1960

Zyablov 1974

Wiberg 1996

Davey 1999

Tanner 1981

Spielman 1996

MacKay 1999

Felström 1999

Richardson, Urbanke, Shokrollahi, Luby, Mitzenmacher, Spielman 2000-2001

Kou 2001

Gaudet 2003

Fossorier 2004

Gross 2005

Rathi 2005

Klinc 2008

Di 2002

Thorpe 2003

Ha, Kim 2004, 2006

Shokrollahi 2006

Savin 2008

Vasic 2002

Yedidia 2003

Miladinovic 2005, 2008

Tehrani 2006

Arıkan 2009

Kschischang 2001

Chen 2002

Vontobel 2003

Declercq 2005, 2007

Boutros 2007

Kudekar 2011

Eleftheriou 2001

Luby 2002

Burshtein 2004

Hu 2005

Razaghi 2007

Li 2008

Planjery 2012

2000

COMP

Von Neumann 1956

Elias 1958

Kuznetsov 1973

Taylor 1968

Dobrushin 1977

Pippenger 1990

Vasic 2004-2007

Varshney 2011

Yazdi 2013

i-RISC challenges

- Propose fault tolerance solutions built upon **modern sparse graph coding** theory
  - LDPC codes already identified as a promising approach

    (but exploited in suboptimal way: only majority-voting decoding was considered)
  - The goal is to exploit the body of knowledge gained in past decade, after the rediscovery of graph-based codes and iterative decoding techniques

- Provide solutions that allow both the **encoder** and the **decoder** to **operate on faulty hardware**
  - New paradigm in coding theory
  - Provide a rigorous analysis of iterative LDPC decoders, through the generalization of existing information theoretic tools to this new paradigm
  - Propose new encoding/decoding algorithms & architectures that can effectively deal with the probabilistic behavior of the circuit

i-RISC challenges

- Design of fault tolerant error correcting modules for **reliable storage & transfer** of digital information throughout the chip
  - Rely on fault-tolerant error correcting codes to perform reliable bus transfer and memory R/W access (encoded information processing)
  - Combine error-correction and constrained coding for crosstalk avoidance
- Integrate error correcting modules into the chip design, in a way to **combine the codec with the hardware it protects**
  - Investigate potential links between the graph representation of a digital circuit and error correcting codes in order to generate fault tolerant implementation of the logical functionality of the circuit
  - Develop a new framework for the synthesis of chips made from unreliable components, through the concept of "error-correcting codes driven" graph augmentation

2. Fault-Tolerant Circuits: Traditional Strategy

- Tolerate Permanent & Temporary Faults
- Key Ingredient – Redundancy
  - Temporal
  - Spatial
  - Information
- Suppress Noise
- Reasonably Well Behaved Devices
- Mostly Design Time
- Always Get the Right Result
- Output is Correct or Faulty
- No In-Between Situation is Acceptable.

2. Fault-Tolerant Circuits: Traditional Methods

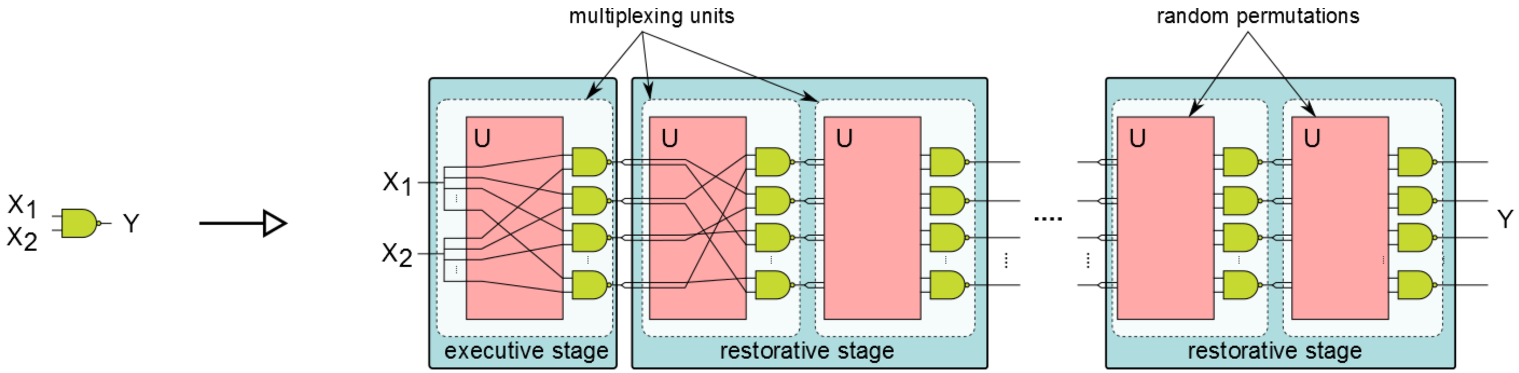
- Logic
  - von Neumann-multiplexing
  - Modular Redundancy, e.g., TMR
  - Averaging Cell
  - Self Diagnosis & Healing
- Memory
  - Parity Check (CRC)
- Interconnect
  - Redundancy
  - Serialization
  - Encoding (≈ Transmission Channel)
- Architecture
  - Multiple Execution w/ Majority Voting (instruction, task)

Performance Penalty

- Energy
- Latency/Throughput
- Area

2. von Neumann Multiplexing

- Fault-tolerant computing with highly unreliable devices
- Concept: Replace the processing unit with parallel working replicas



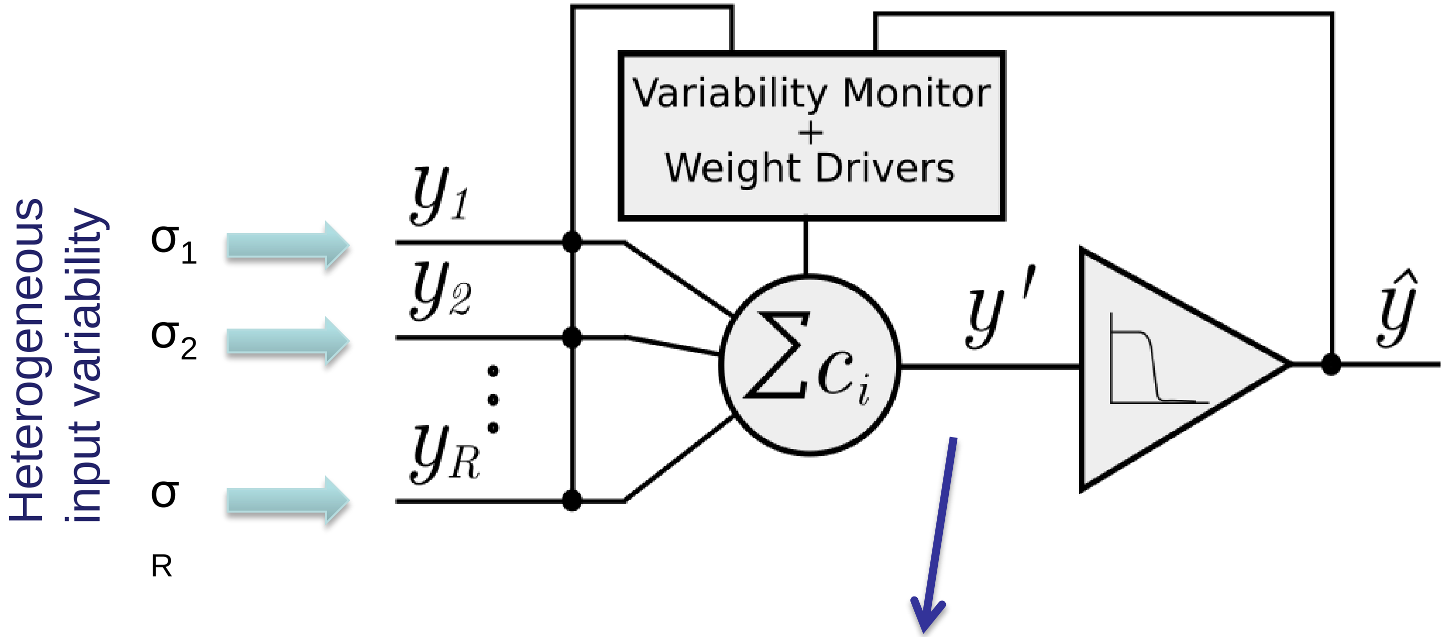
System failure = number of incorrect outputs larger than a defined fraction Δ of the total number of outputs (N)

$P_{fail}(gate) \leq$ 0.0107 & high redundancy factor (N) $\Rightarrow P_{fail}\,(vN{-}MUX) \rightarrow 0$

2. Modular Redundancy

- Compute the r-Replicas $F_{j,\ j=1,r}$ of $F(x_1, x_2, \ldots, x_n)$
- Select F out of F_j via Majority Voting
- TMR, k-MR
- Large Area Overhead
- Majority Voting is the Weak Point
- May Fail if Simultaneous Faults Are Present
- Diversity Helps

2. Adaptive Averaging Cell (AD-AVG)



Heterogeneous input variability

R input replicas:

$$y_i = y + \eta_i$$

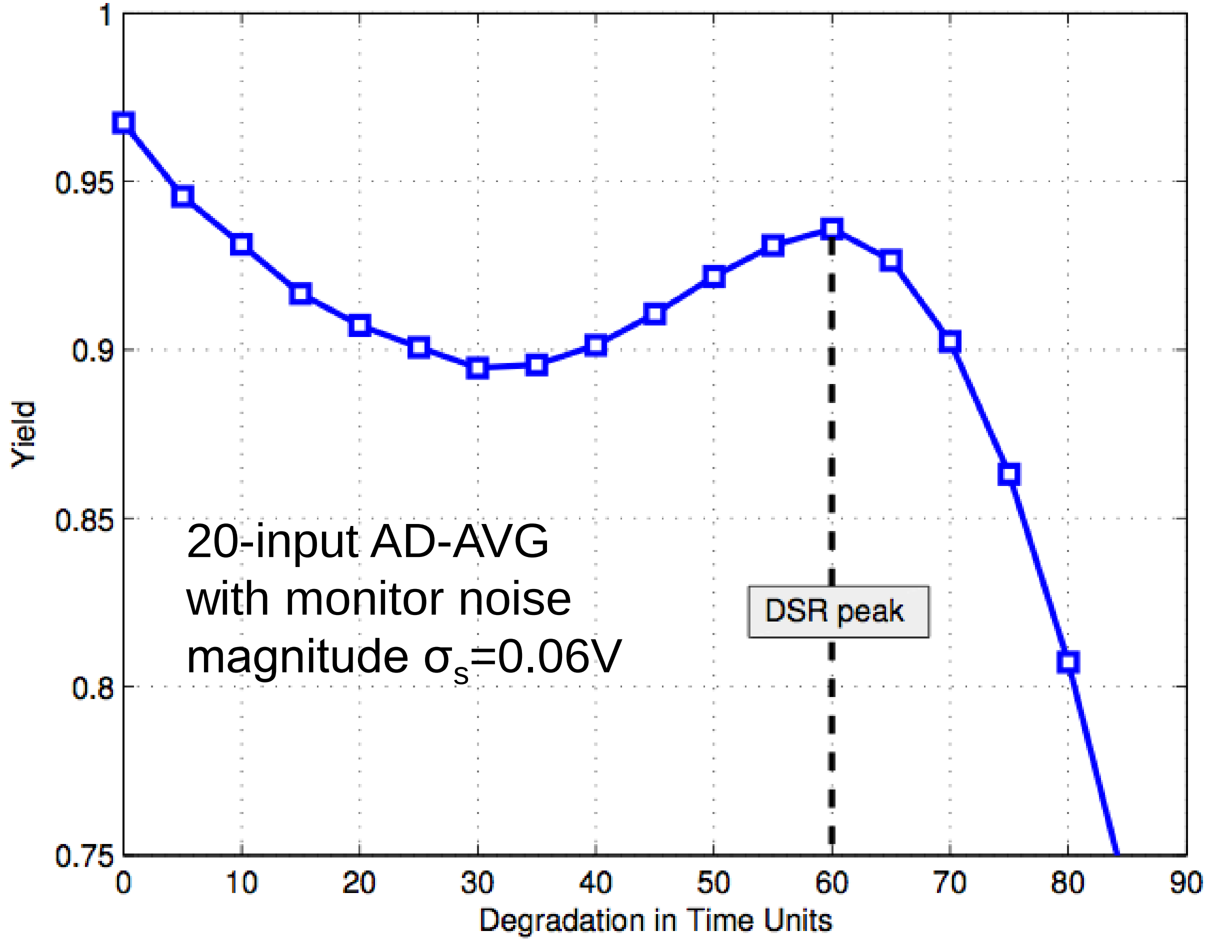
$$\eta_i \sim N(0, \sigma_i)$$

$$\sigma_{y'} = \sqrt{\sum_{i=1}^{R} c_i^2 \sigma_i^2}$$

2. Degradation Stochastic Resonance (DSR)

Optimum weights

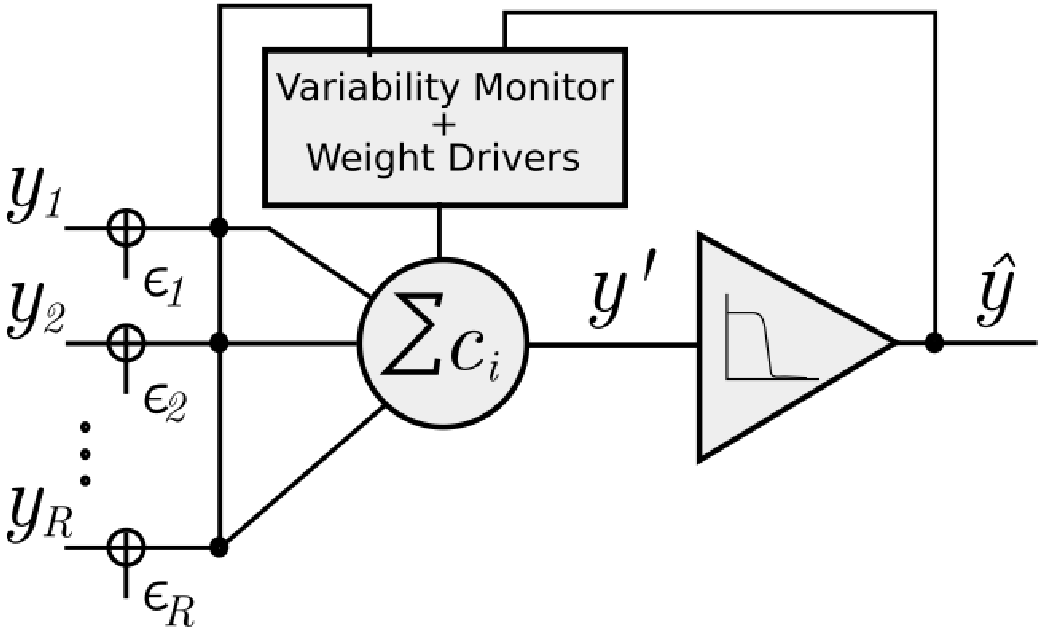
$$c_i^{opt} = \frac{\sigma_{y'\ \min}^2}{\sigma_i^2}$$



20-input AD-AVG with monitor noise magnitude σ_s=0.06V

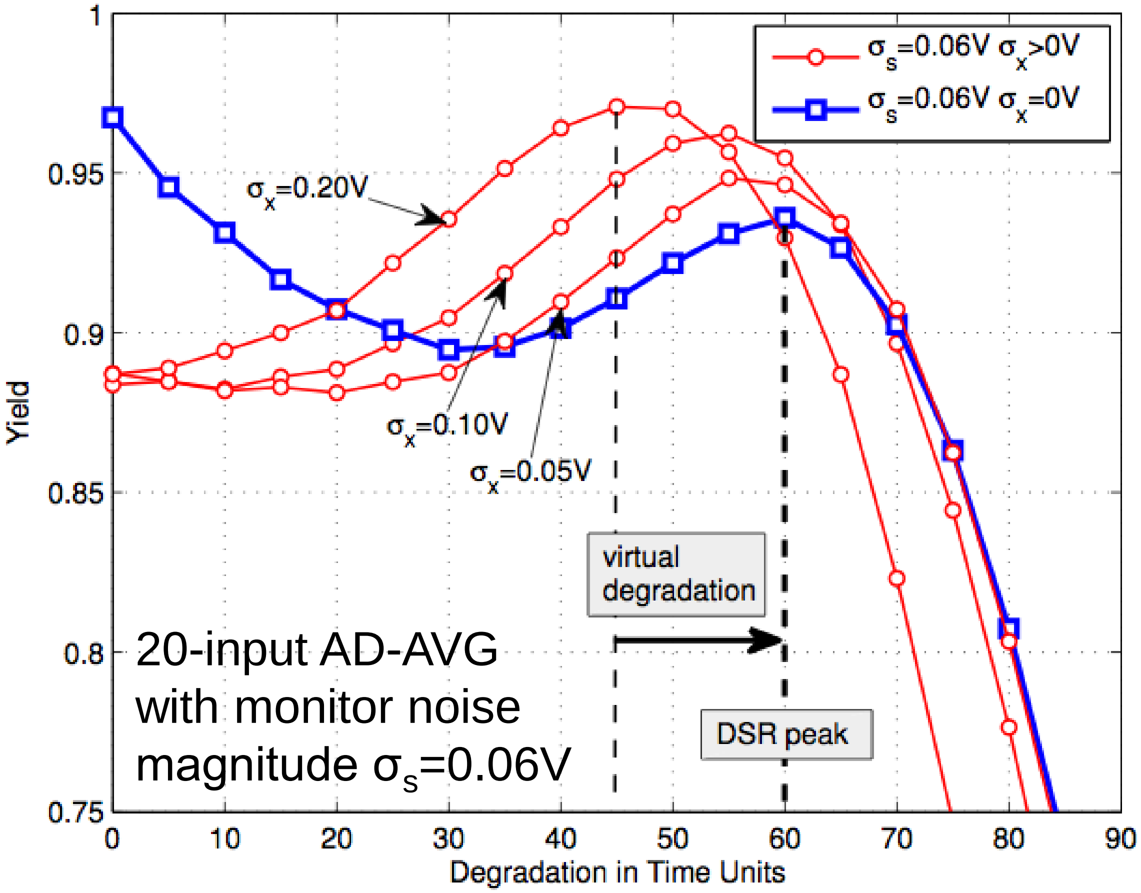
DSR peak

2. Embrace the Noise



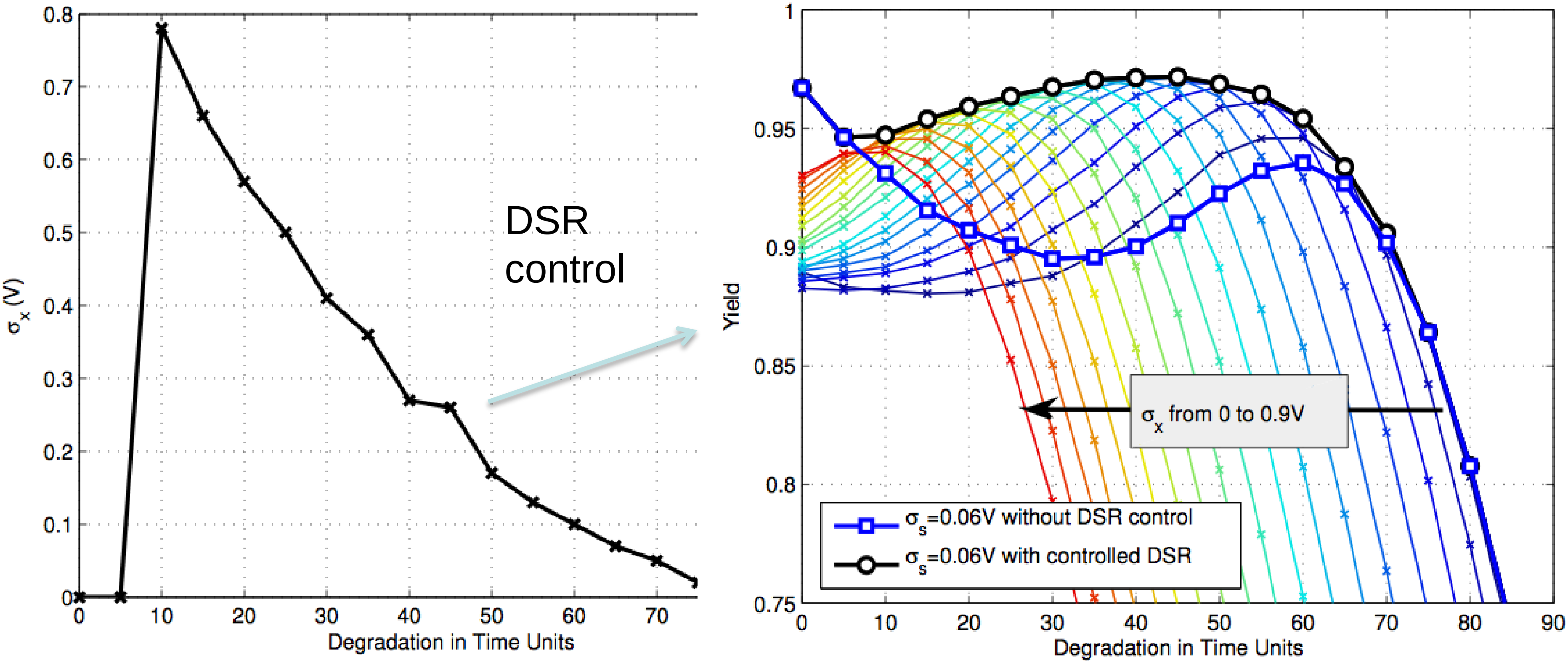
Input noise injectors $\varepsilon_i \sim N(0,\sigma_x)$ to create the DSR peak stochastic conditions regardless of the degradation level.

2. Induced DSR



20-input AD-AVG with monitor noise magnitude σ_s=0.06V

2. DSR Based Reliability Management



Applying the correct amount of input noise we can move along the involute of the colored curves and obtain a yield even higher than that provided by the DSR peak.

2. Reliability Evaluation

Given

- A gate level realization Boolean function $F(x_1, x_2, \ldots, x_n)$,
- $ERR_I = \{ierr_1, err_2, \ldots, err_k\}$,
- $ERR_g = \{gerr_1, gerr_2, \ldots, gerr_l\}$,
- $ERR_{PI} = \{xerr_1, xerr_2, \ldots, xerr_n\}$,

find ERR_F.

- Fast & Accurate
- Design Time
  - Guide the Reliability Aware Synthesis
  - Provide FU Output Error Bounds
  - Enable Evaluation & Optimization of Algorithms on Not Ideal HW
- Runtime
  - Guide Reliability Aware QoS (Energy vs Error Rate Tradeoffs)

2. Reliable Function Implementation

- Redundancy Based FT Computing Solutions Rely on Coding (Temporal & Spatial Redundancy = Repetition Coding)
- Use Telecom Coding Style in Function Implementation – Learn to Cohabitate w/ Errors!
- Overhead
  - Area (Not that Important Nowadays)
  - Energy
- One Way
  - Encode the Inputs $x_i \rightarrow x'_i$
  - Compute $F'(x'_1, x'_2, \ldots, x'_n)$ Instead of F - $Err(F') < Err(F)$!
  - Transform F' Value in F (Only When Explicitly Needed)
- F' is an Acceptable Solution Only If
  - $E(F')_{AT} \leq E(F)_{OT}$ & $ERR(F')_{AT} \leq ERR(F)_{OT}$
  - Otherwise Make Use of OT.

Workshop program

Opening

09:00 -09:30 The i-RISC Project: Innovative Reliable Chip Designs from Low-Powered Unreliable Components

V. Savin (CEA) and S. Cotofana (TUD)

Fault Tolerant Algorithms for Error-Correction

09:30-10:00 Analysis and Design of Min-Sum-based Decoders Running on Noisy Hardware

C. Kameni-Ngassa, V. Savin (CEA), and D. Declercq (ENSEA)

10:00-10:30 Performance of Finite Alphabet Iterative Decoders (FAID) Under Faulty Hardware

D. Declercq, S. Planjery (ENSEA), and B. Vasic (ELFAK)

10:30 – 11:00 **Coffee Break**

Workshop program

Reliable Data Storage and Transport

11:00-11:30 Bit-flipping Decoders for Fault-Tolerant Memories

B. Vasic, S. Brkic, P. Ivanis, and G. Djordjevic (ELFAK)

11:30-12:00 The Analysis of Taylor-Kuznetsov Fault-Tolerant Memories Under Correlated Gate Failures

S. Brkic, P. Ivanis, G. Djordjevic, and B. Vasic (ELFAK)

12:00 – 13:30 **Lunch**

Error Models & Energy Measures

13:30-14:00 Gate Level Simulated Fault Injection for Probabilistic CMOS Circuits

S. Nimara, A. Amaricai, O. Boncalo (UPT), J. Chen, and E. Popovici (UCC)

14:00-14:30 Interconnect Crosstalk Analysis in Sub-powered Integrated Circuits

A. Amaricai, O. Boncalo, and S. Nimara (UPT)

Workshop program

14:30-15:00 Delay Relevant Reliability of CMOS Circuits

C. Jiaoyan, C. Spagnol, S. Kumar, and E. Popovici (UCC)

15:00 – 15:30 **Coffee Break**

Reliable Function Synthesis and Design

15:30 – 16:00 A systematic Approach for Reliability Evaluation of Combinational Circuits

C. Spagnol, S. Kumar, C. Jiaoyan, and E. Popovici (UCC)

16:00 – 17:00 Reliability Assessment Framework for Large Scale Causal Logic Networks

N. Cucu-Laurenciu and S. Cotofana (TUD)

17:00 -17:30 **Open Discussion and Closing Remarks**

Moderators: Valentin Savin (CEA) & Sorin Cotofana (TUD) leti

LABORATOIRE D'ÉLECTRONIQUE
ET DE TECHNOLOGIES
DE L'INFORMATION

CEA-Leti
MINATEC Campus, 17 rue des Martyrs
38054 GRENOBLE Cedex 9
Tel. +33 4 38 78 36 25

www.leti.fr

Merci de votre attention

LLOYD'S REGISTER QUALITY ASSURANCE ISO9001

cea

MINATEC

INSTITUT CARNOT CEA LETI