Reliable Circuits Design with Nanowire Arrays

M. Haykel Ben Jamaa and Giovanni De Micheli

1 Introduction

The emergence of different fabrication techniques of *silicon nanowires (SiNWs)* raises the question of finding a suitable architectural organization of circuits based on them. Despite the possibility of building conventional CMOS circuits with SiNWs, the ability to arrange them into *regular arrays*, called *crossbars*, offers the opportunity to achieve higher integration densities. In such arrays, molecular switches or phase-change materials are grafted at the *crosspoints, i.e.,* the crossing nanowires, in order to perform computation or storage. Given the fact that the technology is not mature, a hybridization of CMOS circuits with nanowire arrays seems to be the most promising approach.

This chapter addresses the impact of variability on the nanowires in circuit designs based on the hybrid CMOS-SiNW crossbar approach. A large part of this chapter has been published in [1].[1] The variability stemming from the shrinking nanowire dimensions is modeled and its impact on the interface between the CMOS circuit and the nanowire arrays, the *decoder*, is investigated. The approach presented is based on the abstract representation of nanowires as a sequence of codes. Based on the impact of variability on codes, optimized design methodologies for encoding the nanowires and for testing the array decoder are derived.

2 Fabrication Technologies

Nanowire crossbars have attracted increasing interest over the last few years because the fabrication techniques have become more mature and versatile. Parallel research works have been carried out at different levels of the IC design hierarchy, ranging from device to circuit and system level, in order to identify and address the

[1] The authors would like to acknowledge those who contributed to [1].

G.D. Micheli (✉)
Institute of Electrical Engineering, EPFL, Lausanne, Switzerland
e-mail: giovanni.demicheli@epfl.ch

N.K. Jha and D. Chen (eds.), *Nanoelectronic Circuit Design*,
DOI 10.1007/978-1-4419-7609-3_5, © Springer Science+Business Media, LLC 2011

challenges facing the utilization of this emerging paradigm in the future. Circuit design depends on properties of the fabrication techniques. Thus, understanding the fabrication techniques and device properties enables a better assessment of the global problem. In the following discussion, we survey the different fabrication techniques for bare nanowires and nanowire crossbars.

2.1 Nanowire Fabrication Techniques

The existing nanowire fabrication techniques follow two main paradigms: the so-called bottom-up and top-down approaches. Bottom-up approaches are based on the growth of nanowires from nanoscale metallic catalysts. In contrast, top-down approaches use various types of patterning techniques.

2.1.1 Bottom-Up Techniques

One of the widely used bottom-up techniques is the *vapor–liquid–solid (VLS)* process, in which the generally very slow adsorption of a silicon-containing gas phase onto a solid surface is accelerated by introducing a catalytic liquid alloy phase. The latter can rapidly adsorb vapor to a supersaturated level; then crystal growth occurs from the nucleated catalytic seed at the metal–solid interface. Crystal growth with this technique was established in the 1960s [2] and silicon nanowire growth is today mastered with the same technique. A related technique to VLS is the laser-assisted catalytic growth. The silicon-containing gas is generated by irradiating a Si substrate with high-powered, short laser pulses [3]. On the other hand, the *chemical vapor deposition (CVD)* method uses materials that can be evaporated at moderate temperatures [4, 5].

2.1.2 Top-Down Techniques

The top-down fabrication approaches have in common the utilization of CMOS steps or hybrid steps that can be integrated into a CMOS process, while keeping the process complexity low and the yield high enough. They also have in common the ability to define the functional structures (nanowires) directly onto the functional substrate.

Standard photolithography techniques: These techniques use standard photolithography to define the position of the nanowire. Then, by using smart processing techniques, including the accurate control of the etching, oxidation and deposition of materials, it is possible to scale the dimensions down far below the photolithographic limit [6–11].

Miscellaneous mask-based techniques: The electron-beam lithography [12, 13] offers a higher resolution below 20 nm than standard photolithography.

It, however, has a lower throughput. The highest resolution can be achieved by using *extreme ultraviolet interference lithography (EUV-IL)* [14]. However, this approach needs a highly sophisticated setup in order to provide the required EUV wavelength.

The stencil technique [15] is a different approach that requires no photoresist patterning. It is based on the definition of a mask that is fully open in the patterned locations. The mask is subsequently clamped onto the substrate, and the material to be deposited is evaporated or sputtered through the mask openings onto the substrate.

Spacer techniques: The spacer technique is based on the idea of transforming thin lateral dimensions, in the range of 10–100 nm, into a vertical dimension by means of an anisotropic etch of the deposited materials. In [16], spacers with a thickness of 40 nm were demonstrated with a line-width roughness of 4 nm and a low variation across the wafer. The nanowire count can be duplicated by using the spacers themselves as sacrificial layers for a following set of spacers [17].

Nanomold-based techniques: Alternative techniques use *nanoimprint lithography (NIL)*, which is based on a mold with nanoscale features [18] that is pressed onto a resist-covered substrate in order to pattern it. The substrate surface is scanned by the nanomold in a stepper fashion. The as-patterned polymer resist is processed in a similar way to photolithographically patterned photoresist films. The *superlattice nanowire pattern transfer technique (SNAP)* and the *planar edge defined alternate layer (PEDAL)* [19] are examples of NIL.

2.2 Crossbar Technologies

The previously surveyed techniques yield parallel or mashed nanowires. In order to arrange them into arrays that are generally called *crossbars*, additional techniques are required and will be explained in the following.

2.2.1 Crossbars with Bottom-Up Nanowires

Nanowires fabricated with bottom-up processes have the property of generally being grown on a different substrate from the functional one. Consequently, they need to be dispersed into a solution and transferred onto the substrate to be functionalized. The iteration of the transfer operations with different directions may lead to a crossbar structure [20]. It has been demonstrated that the application of an electrical field to the substrate improves the directionality of the assembled nanowires [21, 22]. The approach is, however, limited by the electrostatic interference between nearby electrodes, and the requirement for an extensive lithography to fabricate the electrodes [20].

2.2.2 Nanomold-Based Nanowire Crossbars

NIL was used in [23, 18] in order to define two orthogonal layers of metallic nanowires. First, the nanomold was fabricated by electron-beam lithography and *reactive ion etching (RIE)* of a SiO_2-covered silicon substrate. The mold was then pressed onto a spin-coated polymer to define a lift-off mask for Ti/Pt nanowires. A layer of molecular switches, [2]rotaxane, was deposited over the entire substrate using the *Langmuir–Blodgett (LB)* method [24]. Then, the fabrication of the top Ti/Pt nanowire layer was performed in a similar way as explained for the lower nanowire layer. High-density crossbars were also demonstrated with the SNAP technique that was explained in Section. 2.1 [25], yielding 160-kb molecular memories with a density up to 10^{11} bit/cm^2 [26].

2.2.3 Crossbar Switches

Many attempts have been carried out in the last few decades to design molecules comprising a donor-(σ bridge)-acceptor, which have an asymmetric behavior, allowing the current to flow in a preferential direction [27–29]. Another class of switching molecules is represented by bistable molecules, such as [2]rotaxanes, pseudorotaxanes and [2]catenanes. They consist of two mechanically interlocked, or threaded, components having two stable states and can be switched between them when the appropriate bias voltage is applied [30, 31]. Other research groups have focused on phase change materials as a switching material at the nanowire crosspoints [32] operating as diodes.

3 Architecture of Nanowire Crossbars

Nanowire crossbars are defined on a scale that can be far below the lithographic limit. The ability to hybridize CMOS technology with the previously surveyed nanowire techniques (Section. 2) promotes the organization of the overall system in a regular way, where globally CMOS parts are operational, while locally nanowire crossbars are used. This raises the questions of the way in which crossbars should be connected to the outer CMOS circuit on the one hand, and the type of functions that crossbars can execute on the other hand. This section introduces the crossbar organization, surveying some emerging crossbar architectures and focuses on the design of the decoder.

3.1 Organization of Nanowire Crossbars

The baseline organization of a nanowire crossbar circuit is depicted in Fig. 1a. An arrangement of two orthogonal layers of parallel nanowires defines a regular grid of intersections called crosspoints. The separation between the two layers can be filled

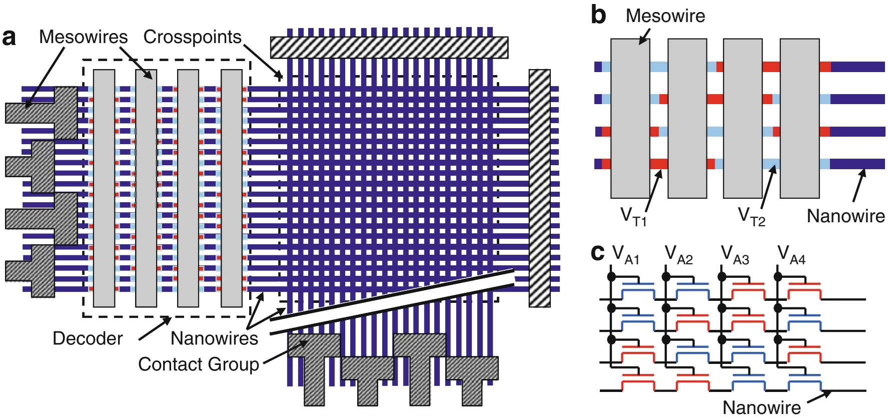


Fig. 1 Baseline organization of a crossbar circuit and its decoder. (**a**) Architecture of a crossbar circuit. (**b**) Decoder layout. (**c**) Decoder circuit design

with phase change material or molecular switches at the crosspoints. Information storage, interconnection or computation can be performed with these crosspoints [30, 33]. A set of contact groups is defined on top of the nanowires. Every contact group makes an ohmic contact to a corresponding distinct set of nanowires, which represents the smallest set of nanowires that allows contacts with lithographically defined lines, called *mesowires (MWs)*. The *mesoscale* corresponds in the context of this chapter to the lithography scale; while the *nanoscale* corresponds to the sublithographic scale.

This configuration bridges every set of nanowires within a contact group to the outer CMOS circuit. In order to *fully bridge the scales* and make every nanowire within this set uniquely addressable by the outer circuit, a decoder is needed. It is formed by a series of transistors along the nanowire body, controlled by the mesowires and having different threshold voltages V_{th} (Fig. 1b). The distributions of V_{th}'s is called the *nanowire pattern*. Depending on this pattern and the pattern of applied voltages in the decoder (V_{A}'s), one single nanowire in the array can be made conductive (Fig. 1c). In this case, this nanowire is said to be addressed by the applied voltage pattern.

It is possible to think of replacing each transistor at the diagonal crosspoints by an ohmic contact and to eliminate all other transistors; thus, mapping each horizontal wire onto a vertical one. However, this method is technologically difficult, because the nanowire pitch is defined below the photolithographic limit, which justifies the proposed decoder design.

3.2 Architectures Based on Nanowire Crossbars

Before the emergence of the crossbar architecture, many experiments were performed with a massively parallel computer built at Hewlett-Packard laboratories, the Teramac [34], in state-of-the-art CMOS technology. Despite the high defect rate affecting single components in the Teramac, the approach seemed to be efficient, resulting in 100× faster robust operation than a high-end single-processor computer in some configurations. The required architectural elements are a large number of computing instances, parallelism of their operation and high bandwidth. Since these elements naturally exist in the crossbar architecture, this architectural paradigm emerged as a possible approach for reliable massively parallel computing with highly defective basic components [35], where molecular devices can perform a logic operation or information storage at the crosspoints.

Some crossbar prototypes were fabricated with different sizes [18, 26, 30], and the basic function that these prototypes implemented is information storage. Crossbars implementing computational units, such as the *nanoBlock* [36, 37], are also conceptually possible. However, they need restoration stages and latches that can be implemented using *resonant tunneling diodes (RTDs)* or by hybridizing crossbars with CMOS. The CMOS part can also provide the necessary gain and input/output interface. It is not excluded that the CMOS part performs more functions than the crossbars in a hybrid architecture, however, the parallelism, reconfigurability and high connectivity will be the main advantages provided by crossbars owing to their matrix form, in addition to their ability to scale down below the limit imposed by photolithography.

The *nanoPLA* architecture is a concept based on semiconducting SiNWs organized in a crossbar fashion with molecular switches at their crosspoints. The switches can be programmed in order to perform either signal routing or wired-OR logic function. The input of the crossbar represents a decoder, which is used in order to uniquely address every nanowire independently of the others. The decoder design assumes that the nanowires are differentiated by a certain doping profile [38]. This will be explained in more detail in Sect. 3.3. The output of the crossbar is routed to a second crossbar, in which the signals can be inverted by gating the nanowires carrying the signals. A cascade of these two planes is equivalent to a NOR plane [39]. Two back-to-back NOR planes can implement any logic function in two-level form.

3.3 Decoding Nanowires

The decoder is the element of the crossbar circuit that bridges the meso- to nanoscale. Even though the structure of the decoder circuit is simple, its reliable fabrication and design are challenging. The need to use different transistors necessitates different doping levels in specific regions on the nanowires whose location cannot be controlled precisely because the nanowire scale is below the lithographic limit. Thus, nanowires that are already doped during the fabrication process may simplify the task. We distinguish, therefore, between fabrication techniques that yield differentiated and those that yield undifferentiated nanowires. Differentiated nanowires have a certain doping profile; they are generally fabricated using a bottom-up approach and the doping profile is defined during nanowire growth. Undifferentiated nanowires have no specific doping profile; they are generally fabricated using a top-down approach.

3.3.1 Decoders for Differentiated Nanowires

Differentiated NWs have an axial or radial doping profile which is defined during the NW growth process. An axial decoder was presented in [40], in which the distribution of the V_{th}'s is fully random. The NWs are dispersed parallel to each other and they are addressable when they have different V_{th} patterns. The probability that their addresses are different may be increased by increasing the number of addressing wires. On the other hand, the radial decoder [41] relies on NWs with several radial doping shells. The remaining shells after a sequence of etchings depends on the etching order in every region. The suite of shells along the NW after all etching steps defines the NW patterns. While both axial and radial decoders require the same estimate of the number of MWs needed to address the available NWs; the radial decoder has the advantage of being less sensitive to misalignment of NWs. To address N NWs, M MWs are needed, $M = [2.2 \times \log_2(N)] + 11$. With these dimensions, the decoders address every nanowire with a probability greater than 99%.

3.3.2 Decoders for Undifferentiated Nanowires

On the other hand, for undifferentiated nanowires, namely those fabricated in a top-down process, a mask-based decoder was presented in [42] and its ability to control undifferentiated NWs was proven. The MWs are separated from the NWs by a nonuniform oxide layer: in some locations, a high-κ dielectric is used, in others, a low-κ dielectric. The high-κ dielectric amplifies the electric field generated by the MWs relatively to the low-κ dielectric. Consequently, the field-effect control by the MWs happens only at the NW regions lying under the high-κ dielectric. The oxide mask is lithographically defined; making the decoder dependent on lithography limits. In order to address N nanowires, the mask-based decoder necessitates the use of $M = 2 \times \log_2(N) + \varepsilon$ mesowires, with ε a small constant ≥ 1, which depends on the fabrication technique and the degree of redundancy to be achieved.

For undifferentiated NWs, a random contact decoder has been presented in [43, 44]. Unlike the other decoders for which the NW codes are among a known set of codes, the connections established between MWs and NWs for this decoder are fully random. It results from a deposition of gold particle onto the NWs, where the only controlled parameter is the density of particles. In order to control each of the N NWs uniquely with a high probability, $M = 4.8 \times \log_2(N) + C$ mesowires are needed, with C being a large constant that depends on the design parameters.

4 Decoder Logic Design

The sequence of V_{th}'s along every NW defining the NW pattern associates a unique code word with the NW that can activate it. Previously explored nanowire encoding schemes, *i.e.,* codes, are binary. The code length impacts the decoder size and the overall crossbar area. It is, therefore, interesting to investigate the benefits of reducing the code length by using multivalued logic (MVL) codes. The generalization of the usual codes to MVL produces novel code families that have not been explored before. In this section, the construction rules for new code families are presented. Defects that can affect them are modeled. Then, the fault tolerance of the considered codes and their impact on the crossbar circuit in terms of reliability and area are investigated.

4.1 Semantic of Multi-valued Logic Addressing

In the following discussion, we generalize the notion of encoding to multiple-valued bits by first defining some basic relations needed to identify possible codes. Some basic concepts used in coding theory are generalized from the binary definitions stated in [45] to multiple-valued logic. The matching of a code word and its pattern corresponds here to conduction. Before introducing the impact of defects, we consider the code (Ω) and pattern (A) spaces to be identical, realizing a one-to-one mapping between each other. Algebraic operations are performed as defined in the ring of integers.

Definition 1. *A multiple-valued pattern* **a**, or simply a pattern **a**, is a suite of M digits a_i, in the n-valued base $\mathbb{B}$; *i.e.,* $\mathbf{a} = (a_0,\ldots, a_{M-1}) \in \mathbb{B}^{\text{M}}$, $\mathbb{B} = \{0,\ldots, n-1\}$. A multiple-valued *code word* **c**, or simply a code word **c**, is defined the same way as a pattern.

A pattern represents a serial connection of M transistors in the silicon nanowire core; each digit a_i of the code word represents a threshold voltage $V_{\text{th},i}$, with the convention $a_i < a_j \Leftrightarrow V_{\text{th},i} < V_{\text{th},j}$, $\forall i, j = 0,\ldots, M-1$. An analogous equivalence holds for $a_i = a_j$ and, Consequently, for $a_i > a_j$. This convention is equivalent to discretizing the M values of V_{th} and ordering them in an increasing order. In Fig. 2a, b, we illustrate the pattern 002120 representing the V_{th} sequence (0.2 V, 0.2 V, 0.6 V, 0.4 V, 0.6 V, 0.2 V).

A code word represents the suite of applied voltages V_{A} at the M mesowires. These are defined such that every $V_{\text{A},i}$ is slightly higher than $V_{\text{th},i}$, and lower than $V_{\text{th},i+1}$ (Fig. 2c, d).

Definition 2. *A complement* of digit x_i in a code word or pattern **x** is defined as: $\text{NOT}(x_i) = \overline{x_i} = (n-1) - x_i$. The operator NOT can be generalized to vector **x**, acting on each component as defined above. Note that $\text{NOT}(\text{NOT}(\mathbf{x})) = \mathbf{x}$.

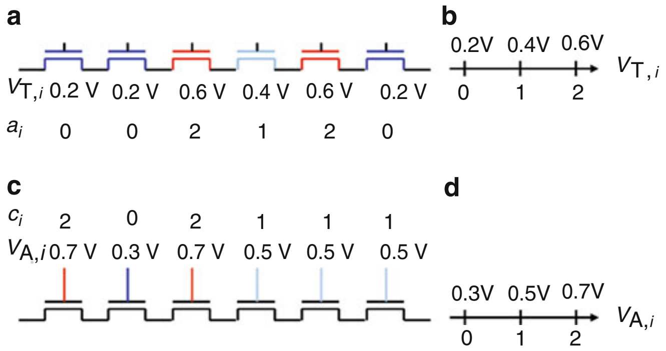


Fig. 2 Mapping of threshold and applied voltages onto discretized values. (**a**) Pattern 002120 and its V_{th} sequence. (**b**) Discretization of V_{th} values. (**c**) Code word 202111 and its V_{A} sequence. (**d**) Discretization of V_{A} values

Definition 3. *A pattern* ***a*** *is covered* by a code word **c** if and only if the following relation holds: $\forall i = 0,\ldots, M - 1, c_i \geq a_i$. By using the sigmoid function:

$$\sigma(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$$

generalized to vectors: $\sigma(\mathbf{x}) = (\sigma(x_0),\ldots, \sigma(x_{M-1}))$, the definition above becomes: **a** is covered by **c** $\Leftrightarrow \|\sigma(\mathbf{a} - \mathbf{c})\| = 0$. Alternatively, we can define the order relations on vectors **c** and **a**:

$$\mathbf{c}{<}\mathbf{a} \Leftrightarrow \forall i, \quad c_i{<}a_i$$
$$\mathbf{c}{>}\mathbf{a} \Leftrightarrow \forall i, \quad c_i{>}a_i.$$

The relation becomes relaxed (*i.e.*, $\leq$ or $\geq$) if there exists i such that $c_i = a_i$. Then, a pattern **a** is covered by a code word **c** if and only if $\mathbf{a} \leq \mathbf{c}$. The same definition for covering can be generalized to two patterns or two code words.

Covering a given pattern with a certain code word is equivalent to applying a suite of gate voltages making every transistor conductive. Then, the nanowire is conducting and we say that it is controlled by the given sequence of gate voltages. Figure 3a illustrates the case in which the code word covers the pattern and the nanowire is conducting, while Fig. 3b illustrates the opposite case.

Definition 4. *A pattern* ***a*** *implies* a pattern **b** if and only if $\|\sigma(\mathbf{b} - \mathbf{a})\| = 0$; *i.e.*, **b** is covered by **a**. We note this as follows: $\mathbf{a} \Rightarrow \mathbf{b}$. Since a one-to-one mapping between the patterns and codes was assumed, we generalize this definition to code words: $(\mathbf{c}^a \Rightarrow \mathbf{c}^b) \Leftrightarrow \|\sigma(\mathbf{c}^b - \mathbf{c}^a)\| = 0$; *i.e.*, $\mathbf{c}^b$ is covered by $\mathbf{c}^a$.

This means that if a nanowire with pattern **a** corresponding to code word $\mathbf{c}^a$ is covered by a code word $\mathbf{c}^*$, then the nanowire with pattern **b** corresponding to code

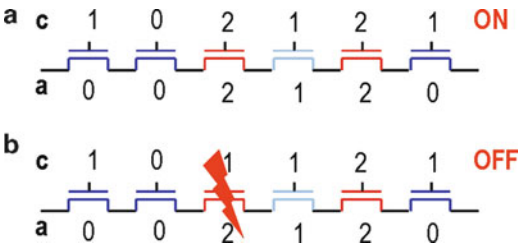


Fig. 3 Example of conducting and nonconducting nanowires. (**a**) Conducting nanowire (code word covers pattern). (**b**) Nonconducting nanowire (code word does not cover pattern)

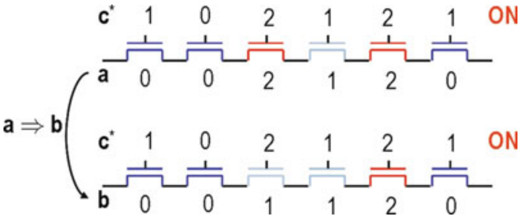


Fig. 4 Example of implication between patterns: $\mathbf{c}^*$ covers **a**; since **a** ⇒ **b**, then $\mathbf{c}^*$ also covers **b**

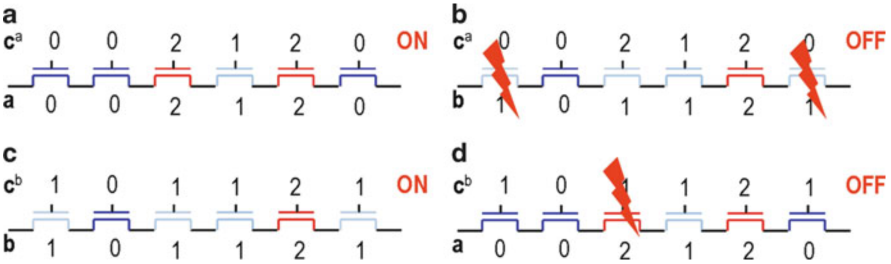


Fig. 5 Example of independent covering: code words $\mathbf{c}^a$ and $\mathbf{c}^b$ are independently covered

word $\mathbf{c}^b$ is also covered by the same code word $\mathbf{c}^*$. Applying the voltage suite $\mathbf{c}^*$ will result in turning on the nanowires with either pattern (see Fig. 4).

Definition 5. *Code words $\mathbf{c}^a$ and $\mathbf{c}^b$ are independently covered* if and only if $\mathbf{c}^a$ does not imply $\mathbf{c}^b$ and $\mathbf{c}^b$ does not imply $\mathbf{c}^a$.

This definition means that there exists a voltage suite that turns on the nanowire with pattern **a** corresponding to $\mathbf{c}^a$, but not with pattern **b** corresponding to $\mathbf{c}^b$ (see Figs. 5a, b). Reciprocally, there exists a second voltage pattern that turns on the nanowire with pattern **b** corresponding to $\mathbf{c}^b$, but not with pattern **a** corresponding to $\mathbf{c}^a$ (see Figs. 5c, d).

Definition 6. *Code word $\mathbf{c}^a$ belonging to set Ω is addressable* if and only if it does not imply any other code word in $\Omega\backslash\{\mathbf{c}^a\}$. We define set Ω to be addressable if and only if every code word in Ω is addressable.

Assuming that there is a one-to-one mapping between code space Ω and pattern space A, then saying that a code word $\mathbf{c}^a$ implies no other code word in $\Omega \backslash \{\mathbf{c}^a\}$ is equivalent to saying that it covers only pattern **a** and no other pattern in $\mathrm{A} \backslash \{\mathbf{a}\}$. Thus, there exists a voltage sequence that activates only the nanowire with pattern **a** and no other nanowire having its pattern in $\mathrm{A} \backslash \{\mathbf{a}\}$.

Proposition 1. *A set Ω of code words is addressable if and only if every code word in Ω is independently covered with respect to any other code word in Ω.*

Proof. This follows directly from Defs. 5 and 6.

Consequently, an admissible set of applied voltages that uniquely addresses each nanowire corresponds to the set of code words Ω that independently covers every pattern in A. This set of patterns can be simply taken as Ω itself, if Ω is addressable.

4.2 Code Construction

4.2.1 Hot Encoding

In binary logic, the (k, M) hot code space is defined as the set of code words with length M having k occurrences of bit '1' and $(M - k)$ occurrences of bit '0' in every code word ($k \leq M$). It is also known as the k-out-of-M code; which was first used as a defect-tolerant encoding scheme [46]. This definition can be generalized to the n-valued logic. We first define **k** as an n-dimensional vector $(k_0, \ldots, k_{n-1})$, such that $\sum_i k_i = M$. Then, the multivalued $(\mathbf{k}, M)$-hot encoding is defined as the set of all code words having length M such that each k_i represents the occurrence of digit i, $i = 0, \ldots, n - 1$. We consider, for instance, the ternary logic ($n = 3$), and we set $\mathbf{k} = (4, 3, 1)$ and $M = 8$. Then, every code word in the considered $(\mathbf{k}, M)$-hot space contains 4× the digit '0', 3× the digit '1' and 1× the digit '2'. The considered code space includes, for instance, code words 00001112 and 00210110. The code space defined by a multi-valued $(\mathbf{k}, M)$-hot encoding is addressable and its size is maximal for $k_i = M/n$, $\forall$ $i = 0, \ldots, n - 1$. The size of the maximal-sized space is asymptotically $\propto n^M / M^{(n-1)/2}$ for a given n. In this chapter, it is implicitly understood that the $(\mathbf{k}, M)$-hot code with the maximal-sized space is used, even when just $(\mathbf{k}, M)$-hot code is mentioned.

4.2.2 N-ary Reflexive Code

The binary tree code with length M is a 2-to-2^M encoder representing the 2^M binary numbers $0 \cdots 0$ to $1 \cdots 1$. Similarly, an n-ary tree code with length M is defined as the set of n^M numbers ranging from $0 \cdots 0$ to $(n - 1) \cdots (n - 1)$. For instance, the ternary ($n = 3$) tree code with length $M = 4$ includes all ternary logic numbers ranging from 0000 to 2222. As one can easily see, some code words imply many others from the same space: for instance, 2222 implies all other code words. It is possible to prevent the inclusive character of the n-ary tree code by attaching the complement of the code word (*i.e.,* 2222 becomes 22220000). The as-constructed code is the *N-ary Reflexive Code (NRC)*. The code space defined by the NRC is addressable and its size is n^M. In a similar fashion, the reflection principle works for any other code (*e.g.,* Hamming code), making the whole code space addressable. However, in return it doubles the code length.

4.3 Defect Models

4.3.1 Basic Error Model

Figure 6 illustrates the main assumptions behind basic error models. We assume that the threshold voltages $V_{\mathrm{th},i}$ are equidistant; *i.e.,* $V_{\mathrm{th},i+1} - V_{\mathrm{th},i} = 2\alpha V_0$, V_0 being a given scaling voltage and α is given by the technology. The applied voltages $V_{\mathrm{A},i}$ are set between every two successive threshold voltages $V_{\mathrm{th},i}$ and $V_{\mathrm{th},i+1}$, not necessarily in the middle, rather shifted by υV_0 towards $V_{\mathrm{th},i}$; where υ is a design parameter.

If the variability of $V_{\mathrm{th},i}$ is high or the spacing between two successive $V_{\mathrm{th},i}$'s is low due to the large number of doping levels, then $V_{\mathrm{th},i}$ may exceed a voltage $V_{\mathrm{X},i}$ given by $V_{\mathrm{A},i} - \delta \cdot V_0$; where δ will be derived later. When V_{th} increases, the sensed current, while a_i is applied to digit c_i, decreases ($a_i = c_i$) and the sensed current,

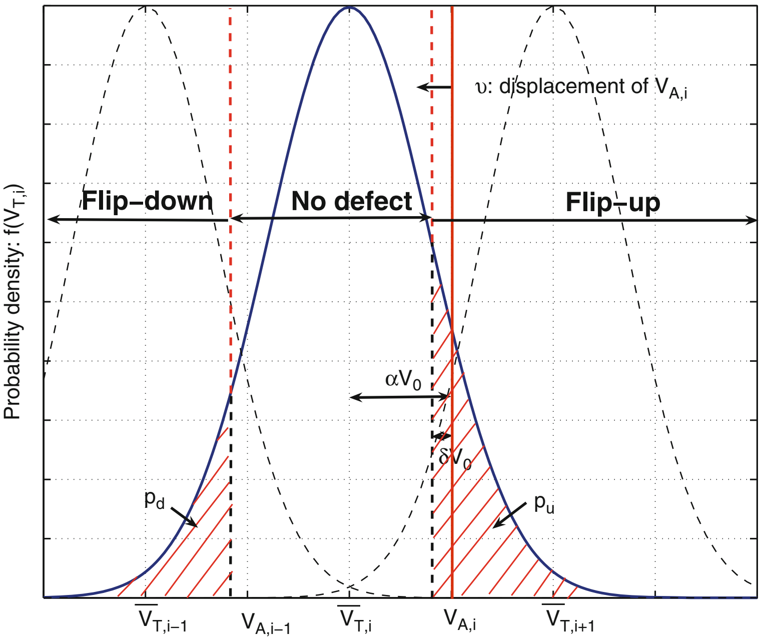


Fig. 6 Coding defects induced by V_{th} variability

while $a_i + 1$ is applied to the same digit, increases. Voltage $V_{X,i}$ is defined as the gate voltage which results in the decrease of the sensed current for a_i by a factor q from its value at $\overline{V}_{th,i}$. Higher the q, more accurate the sensing. Thus, q is also considered a design parameter. Assuming that the transistors are saturated, then the current in the saturation region is proportional to $(V_{A,i} - V_{th})^2$, where V_{th} is the actual threshold voltage. Consequently, the following condition on V_X must hold: $(V_{A,i} - \overline{V}_{th,i})^2/(V_{A,i} - V_{X,i})^2 = q$; which gives: $\delta = (\alpha + \upsilon)/\sqrt{q}$ for long-channel transistors.[2] This fixes the values of $V_{X,i}$; when $V_{th,i}$ exceeds $V_{X,i}$, digit a_i acts as $a_i + 1$; its address becomes $c_i + 1$ and we call this case the *flip-up defect*.

Now, consider the case when $V_{th,i}$ falls below $V_{A,i-1} - \delta \cdot V_0 = V_{X,i-1}$, then the current flowing while $a_i - 1$ is applied is not ~ 0 anymore, and always greater than q times the current flowing while $c_i - 2$ is applied. Then, a_i is implied by c_i and $c_i - 1$ but not by $c_i - 2$; its address is $c_i - 1$ which means that a_i acts as $a_i - 1$; this case is called the *flip-down defect*. The probabilities of flip-ups and flip-downs are given by the following expressions, which are independent of i. Here, f_i is the probability density function of $V_{th,i}$:

$$p_u = \int_{V_{X,i}}^{\infty} f_i(x)\mathrm{d}x \quad p_d = \int_{-\infty}^{V_{X,i-1}} f_i(x)\mathrm{d}x$$

When $V_{th,i}$ falls within the range between the threshold values for flip-up and flip-down defects, the digit is correctly interpreted. We notice that the flip-down error never occurs at digits having the smallest value, 0, since the corresponding $\overline{V}_{th,i}$ is by definition smaller than the smallest $V_{A,i}$ available. For the same reason, the flip-up error never occurs at digits having the largest value, $n - 1$. In order to study the size of the addressable code space, we consider flip-up and flip-down errors in the code space instead of flip-up and flip-down defects at the nanowires, since the two considerations are equivalent.

4.3.2 Overall Impact of Variability

If V_{th} varies within a small range close to its mean value, then the pattern does not change, since the nanowire still conducts under the same conditions. Then, a one-to-one mapping between the code and the pattern space holds, which is shown in Fig. 7 for a ternary hot code with $M = 3$. On the contrary, if the V_{th} variation is large, then some digits may be shifted up or down, as explained above. When a pattern has a sequence of errors, it can be either covered by one or more code words or it can be uncovered. When we consider the code words, some of them cover one or more patterns and some cover no pattern under the error assumptions. The following example explains this conjecture:

[2] If we consider short-channel transistors, then the saturation current is proportional to $(V_{A,i} - V_{th})$ and $\delta = (\alpha + \upsilon)/q$.

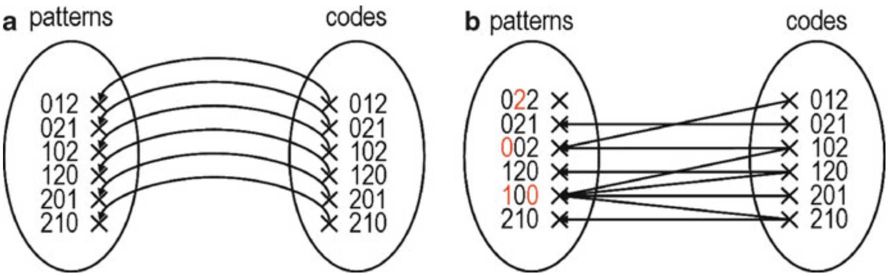


Fig. 7 Mapping of the code space onto the pattern space. (**a**) Mapping in the defect-free case. (**b**) Mapping and in the case of defects

Example 1. *Figure 7b illustrates the digit shift at some patterns. We notice that the first pattern 022 (which underwent a defect) is not covered by any code word anymore. Thus, its nanowire cannot be addressed. All the other patterns are covered at least by one code word. Two categories among these covered patterns can be distinguished. On the one hand, the fourth pattern 120 is covered by the fourth code word, which in turn covers another pattern (the fifth). Thus, by activating the fourth nanowire, the required control voltages activate either the fourth or fifth nanowire. Consequently, the fourth nanowire cannot be addressed uniquely. This case represents the patterns covered only by code words covering more than a single pattern. On the other hand, the complementary case is illustrated by the fifth pattern 100, which is covered by many code words. However, one of these code words (201) covers no other pattern except the considered one. Thus, it is possible to uniquely activate the fifth nanowire by applying the voltage sequence corresponding to code word 201.*

The examples shown in Fig. 7 demonstrate that a pattern undergoing defects can be either (1) not covered by any valid code word, in which case the nanowire cannot be identified as addressable and the pattern is useless; or (2) covered by at least one valid code word. In the second case, if two patterns or more are covered by the same code word, then this code word cannot be used because more than one nanowire would have the same address. Thus, in the second case, the pattern is only useful if at least one code word covering it covers no other pattern, insuring that the covered pattern can be addressed.

Assuming that, on an average, every code word covers v patterns when errors occur, let p_{I} be the probability that a pattern becomes uncovered, and p_{U} the probability that a code word covers a unique pattern ($\bar{p}_{\mathrm{U}} = 1 - p_{\mathrm{U}}$). Let $|\Omega|$ be the original size of the code space and $|\Omega'|$ the size after errors occur. Set Ω' contains useful addresses under defect conditions, *i.e.,* those that address unique nanowires even though the nanowires are defective. The size of Ω' indicates the number of nanowires that remain useful under high variability conditions. Then:

$$|\Omega'| \;=\; |\Omega| \cdot \;(1 - p_{\mathrm{I}})(1 - \bar{p}_{\mathrm{U}}^{v}) \qquad (1)$$

A model for multidigit errors in multivalued logic codes was presented in [47] and gives an estimate of p_{I} and p_{U} for both types of code space. Parameter v is estimated as a fit parameter from Monte Carlo simulations.

4.4 Impact of the Encoding Scheme

In order to assess the variation of the addressable code space under variable V_{th}, we plotted separately the uncovered part $|\Omega|_{\mathrm{un}} = p_{\mathrm{I}} \cdot |\Omega|$, the addressable part $|\Omega'|$, and the immune part $|\Omega|_{\mathrm{im}}$ in which no defects occur. The fit parameter v was estimated with Monte-Carlo simulations. Figure 8 shows the sizes of these subspaces for a ternary (3, 14)-reflexive code depending on the 3σ-value of V_{th}. The Monte-Carlo simulation confirms in the same figure the analytical results and gives the value 2.8 for the fit parameter v. The size of the addressable space $|\Omega'|$ drops quickly when 3σ reaches 0.4 V. At the same time, more patterns become uncovered. Interestingly, there are more addressable than immune patterns, because some defective patterns can be randomly addressed. This tendency increases for unreliable technologies, and around 10% of the original code space size can be randomly addressed under extreme conditions. The simulation of hot codes was not shown, because the result

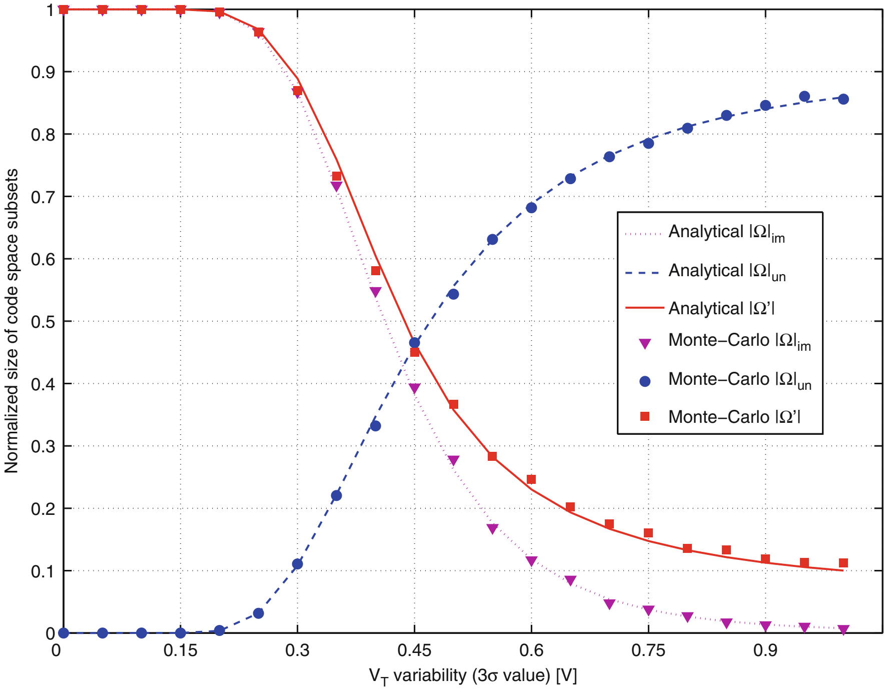


Fig. 8 Dependency of different code space subsets on V_{th} variability

is similar, except for large defect probabilities: under these conditions, the size of the addressable space goes faster towards 0 because the construction of hot codes imposes more constraints than the NRC.

The sizing of memory blocks (*i.e.*, the size of contact groups in Fig. 1) and the number of V_{th}'s are interdependent. As a matter of fact, Fig. 9 shows that increasing the number of V_{th}'s has two opposite effects: on one hand, it enables the addressing of more wires with the same code length; on the other hand, it makes the transistors more vulnerable to defects and increases the number of lost code words. A typical trade-off situation is illustrated in Fig. 9 with the ternary (3,9) and binary (2,12) hot codes (with $(n, k) = (3,3)$ and (2,6), respectively) yielding almost the same number of addressable nanowires for 3σ around 0.4 V. The first one saves area because it has shorter code words, whereas the second one is technologically easier to realize (only two different V_{th}'s). The use of the ternary decoder is recommended for reliable technologies (ensuring less area and more code words), but when the technology becomes more unreliable, there is a trade-off between area savings and easier fabrication process.

The benefits of using multivalued logic to design bottom-up decoders is summarized in Table 1. Among the decoders presented in Sect. 3.3, the radial decoder would need several oxide shell thicknesses and the random contact decoder would

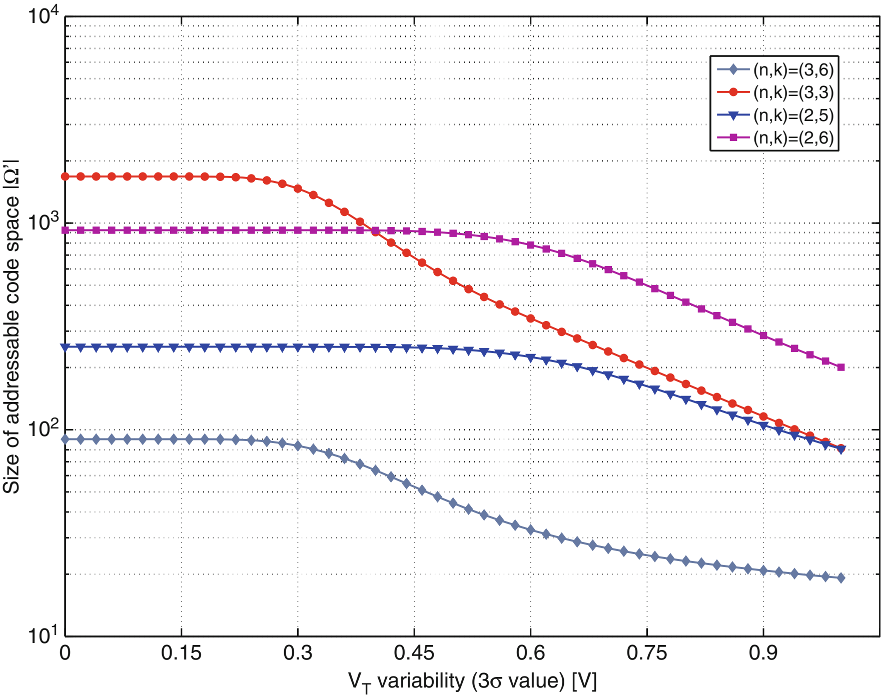


Fig. 9 Number of addressable nanowires for different hot codes

Table 1 Yield of different decoders in terms of area per working bit (nm^2) at the technology node 45 nm

Raw size (kB)	Base	Axial decoder	Mask-based
8	2	1,576	622
8	3	1,196	550
8	Δ	24.1%	11.5%
32	2	846	423
32	3	676	373
32	Δ	20.2%	11.8%

need more than one level of conduction in order to be extended to *n*-ary logic. These features are not inherent to the decoders, as shown in [41] and [44]; thus they cannot be extended to multilevel logic. On the contrary, it is possible to assume more than two levels of doping for the axial decoder and more than one oxide thickness for the mask-based decoder in order to perform MVL addressing without altering the underlying decoding paradigm. Consequently, only these two decoders were extended to MVL addressing. The bottom-up approaches promise a high effective density under technological assumptions that are still to be validated. The use of ternary logic in 32 kB raw area memories saves area up to 20.2% for memories with axial decoder, and up to 11.8% for memories with mask-based decoder.

5 Testing Crossbars

The physical defects affecting the nanowires have been modeled at a high abstraction level as changes in the nanowire addresses. A defect can cause a change of the nanowire address such that the nanowire becomes unaddressable in the considered code space, or it shares the same address with another nanowire. In these cases, it is required that defective nanowire addresses are detected and discarded from the used set of addresses. This task can be performed by testing the decoder circuit.

Testing the decoder, in order to keep only defect-free parts of the code space, highly simplifies the test procedure of the whole crossbar circuit. This section proposes a test method that identifies the defective codes. The method quantifies the test quality, measured as the probability of test error, and it investigates the dependency of the test quality on the decoder design parameters. Without loss of generality, crossbar circuits considered in the following implement a memory function.

5.1 Testing Procedure

This section presents an overview of a test method that can be applied to nanowire arrays. This is an exhaustive method used to illustrate the testing principle. More efficient pseudo-random techniques also exist. However, the focus here is only on the thresholder design and the test quality.

The nanowire testing is performed for every layer separately. Thus, we depicted a single nanowire layer with its additional test circuitry in Fig. 14. Besides the nanowire layer, the system comprises the interfacing circuit (decoder) and a CMOS part formed by a thresholder, a control unit and a *lookup table (LUT)*. The thresholder measures the output current and indicates whether a single nanowire is detected. The control unit regulates the execution of the testing phase and other functions, such as the reading and writing operations. The LUT stores the valid addresses, *i.e.,* those that activate a single nanowire each.

The test can be performed by applying the following exhaustive procedure. First, the two nanowire layers are disconnected by setting the power (V_P) and sense (GND) electrodes of every layer to the same voltage, such that a large voltage drop is created between the two layers. Then, we consider every layer separately. By going through all possible addresses, a voltage V_P is applied; then the address is stored in the LUT if the sensed current indicates the activation of a single nanowire. The same procedure is repeated for the second layer and it is linear with N.

The output of the nanowire layer (I_s) is sensed by the thresholder. We assume that the variability mainly affects the sublithographic part of the memory representing the nanowire array. This part is fabricated using an unreliable technology, unlike the rest of the circuit, defined on the lithography scale and assumed to be more robust. Thus, we consider that the thresholder, the control circuit or the LUT are defect-free. The thresholder senses I_s, it possibly amplifies it, then it compares I_s to two reference values (I_0 and I_1 with $I_0 < I_1$). If the sensed current is smaller than I_0, then no nanowire is addressed. If the sensed current is larger than I_1, then at least two nanowires are activated with the same address. If the sensed current is between the reference current levels, then only one nanowire is activated and the address is considered to be valid. Given the statistical variation of the threshold voltages, the ability to correctly detect addresses can be expressed with the following probabilities (Fig. 10):

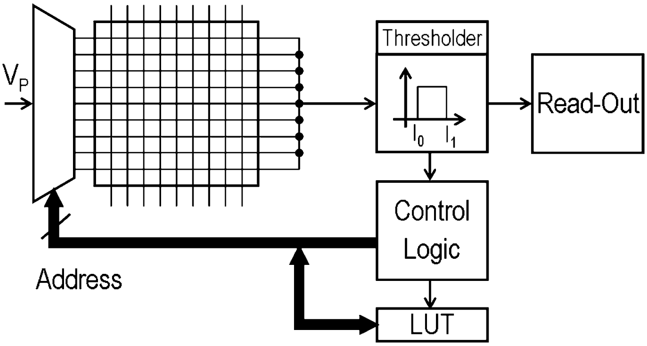


Fig. 10 Crossbar memory and testing unit: besides the memory array and the decoder, the system comprises a CMOS part formed by a thresholder that detects the bit state, a control unit that synchronizes the test operation, and a LUT that saves correct addresses

$$\begin{cases} P_0 = \Pr\{(I_s \le I_0) \text{ given that no nanowire is addressed}\} \\ P_1 = \Pr\{(I_0 < I_s < I_1) \text{ given that 1 nanowire is addressed}\} \\ P_2 = \Pr\{(I_1 \le I_s) \text{ given that } \ge 2 \text{ nanowires are addressed}\} \end{cases} . \quad (2)$$

Then, the probability that all three events happen simultaneously is given by: $P_0 \times P_1 \times P_2$, assuming that the considered events are independent. We can define the error probability of this test procedure as follows:

$$\varepsilon = 1 - P_0 \times P_1 \times P_2 \quad (3)$$

The purpose of the following is to design the thresholder in order to obtain the best test result with the smallest ε. In the next sections, we derive the analytical expressions of P_0, P_1 and P_2, then we optimize I_0 and I_1 in order to minimize ε.

5.2 Perturbative Current Model

During the code testing phase, every nanowire is disconnected from the crossing nanowires. It can be modeled as a wire connecting the power electrode to the sensing electrode and formed by two parts (see Fig. 15): the decoder part that is a series of M pass transistors, and the memory part. Since the memory part is disconnected from the second layer of nanowires, it can be modeled as a resistive load R_M. We model the devices (SiNWFETs) in this section in a general way as a voltage-controlled current sources, *i.e.,*: $I = f(V_{DS}, V_{GS}, V_T)$ where I is the drain-source current, V_{DS}, V_{GS} and V_T are respectively the drain-to-source, gate-to-source and threshold voltages. The decoder design is based on two different V_T's ($V_{T,Ref0}$ and $V_{T,Ref1}$ such that $V_{T,Ref0} < V_{T,Ref1}$, and we define $\Delta V_T = V_{T,Ref1} - V_{T,Ref0}$). When a nanowire is addressed, every variation of V_T results in a variation of the current through the nanowire, which can be noted the following way (Fig. 11):

$$I = I^{OP} + \delta I \quad (4)$$

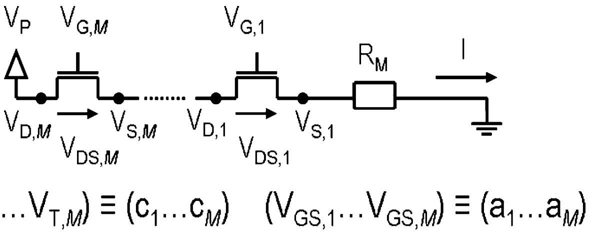

V_P
$V_{G,M}$
$V_{G,1}$
R_M
I
$V_{D,M}$
$V_{S,M}$
$V_{DS,M}$
$V_{D,1}$
$V_{S,1}$
$V_{DS,1}$
$(V_{T,1} \dots V_{T,M}) \equiv (c_1 \dots c_M)$ $(V_{GS,1} \dots V_{GS,M}) \equiv (a_1 \dots a_M)$

Fig. 11 Electrical parameters of a biased nanowire under test: the decoder part is represented by M transistors in series, and the memory part is represented by a resistance R_M. Notice that the perpendicular nanowire layer is disconnected from the nanowire under test

The signal I is linearized around the *operating point (OP)* and divided into a *large* I^{OP} and a *small signal* δI. This approach is widely used in circuit and network theory and in sensitivity analysis [48]. The large signal can be estimated with a SPICE simulator. The small signal can be calculated by linearizing all the equations describing the circuit around the OP:

$$\delta I = -\frac{1}{R_{\mathrm{M}}} \cdot v^{T} \cdot A^{-1} \cdot B \cdot \delta V_{\mathrm{T}} \tag{5}$$

with the variational vector $\delta \mathbf{V}_{\mathrm{T}} = [\delta V_{\mathrm{T},1},\ldots, \delta V_{\mathrm{T},M}]^{\mathbf{T}}$ for the threshold voltages, and the small signal matrices **A** and **B** given by:

$$A = \begin{bmatrix} 1 + r_1 \cdot g_{\mathrm{DS},1} & 1 & \cdots & 1 \\ 1 - r_2 \cdot g_{\mathrm{m},2} & 1 + r_2 \cdot g_{\mathrm{DS},2} & \cdots & 1 \\ \vdots & & & \vdots \\ 1 - r_M \cdot g_{\mathrm{m},M} & 1 - r_M \cdot g_{\mathrm{m},M} & \cdots & 1 + r_M \cdot g_{\mathrm{DS},M} \end{bmatrix},$$

$$B = \begin{bmatrix} -r_1 \cdot g_{\mathrm{T},1} & 0 & \cdots & 0 \\ 0 & -r_2 \cdot g_{\mathrm{T},2} & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & -r_M \cdot g_{\mathrm{T},M} \end{bmatrix}.$$

We used the following notations: $g_{\mathrm{DS},i} = \partial f_i / \partial V_{\mathrm{DS},i}$, $g_{\mathrm{m},i} = \partial f_i / \partial V_{\mathrm{GS},i}$, $g_{\mathrm{T},i} = \partial f_i / \partial V_{\mathrm{T},i}$ and $r_i = R_{\mathrm{M}} || g_{\mathrm{m},i}^{-1}$ (parallel resistance connection). All the components of the matrices **A** and **B** are considered at the operating point.

5.3 Stochastic Current Model

We divide the sensed current into a useful and a noisy part. The useful signal (I_{u}) is the current that flows through a nanowire when the code corresponding to its pattern is applied. On the other hand, the noise can be generated by two different processes: intrinsically (I_{i}), or defect-induced (I_{d}). The intrinsic noise is generated by nanowires that are switched off, which generate subthreshold current. The defect-induced noise is generated by unintentionally addressed nanowires. Their number is denoted by N_{def}, while the number of nanowires generating intrinsic noise is N_{off}. Since the total number of nanowires is N, the following equation must hold $N_{\mathrm{use}} + N_{\mathrm{off}} + N_{\mathrm{def}} = N$, where $N_{\mathrm{use}} = 0$ if no nanowire is activated by the applied code, and $N_{\mathrm{use}} = 1$ otherwise.

5.3.1 Distribution of the Useful Signal

Every V_T is considered as an independent and normally distributed stochastic variable with mean value $\overline{V}_T$ and standard deviation σ_T: $V_T \sim N(\overline{V}_T, \sigma_T^2)$. If the nanowire pattern is correct, then the operating point of V_T coincides with its mean value. If a defect happens so that the bit representing V_T flips, then the operating point of V_T is shifted from the mean value of V_T by $-\Delta V_T$.

We consider a nanowire with a defect-free pattern **a**, which is controlled by its corresponding code $\mathbf{c}^a$, and which generates the useful signal I_u. Then, $V_T^{OP} = \overline{V}_T$ and $\delta \mathbf{V}_T \sim N(0, \sigma_T^2 \cdot v)$ hold. Then, a useful signal follows the distribution resulting from (Eq. 10) to (Eq. 11). The operating point is the on-current of the transistors I_{on}, which is calculated with SPICE simulator; whereas the variable part is given by (Eq. 11). We obtain the following mean value and standard deviation of I_u:

$$\begin{cases} \bar{I}_u = I_{on} \\ \sigma_u = \dfrac{\sigma_T}{R_M} \cdot \parallel v^T A^{-1} B \parallel \end{cases} \tag{6}$$

5.3.2 Distribution of the Defect-Induced Noise

Now we consider a nanowire NW^b with the pattern **b** that undergoes some defects and turns into $\mathbf{b}^*$. This defective nanowire can be activated by the code $\mathbf{c}^a$ of another nanowire NW^a having the pattern **a**. In this case, NW^b generates a defect-induced noise I_d. This defect can be described by a series of shifts at the digits of **b** represented by the vector $\mathbf{s} \in \{0, 1\}^M$, where $\Delta V_T \cdot s_i \in \{0, \Delta V_T\}$ indicates whether a threshold voltage shift happened at the transistor i ($i = 1, \ldots, M$). Assuming that N_{def} nanowires generate a defect-induced noise, then every one of them is characterized by a given threshold voltage shift vector $\mathbf{s}_i$, $i \in \{1, \ldots, N_{def}\}$. By applying the summation rule of independent Gaussian distributions, we obtain the following mean value and standard deviation of I_d:

$$\begin{cases} \bar{I}_d = N_{def} \cdot I_{on} - \dfrac{\Delta V_T}{R_M} \cdot v^T \mathbf{A}^{-1} \mathbf{B} \cdot \sum\limits_{i=1 \ldots N_{def}} \mathbf{s}_i \\ \sigma_d = \dfrac{\sqrt{N_{def}} \cdot \sigma_T}{R_M} \cdot \parallel \mathbf{v}^T \mathbf{A}^{-1} \mathbf{B} \parallel \end{cases} . \tag{7}$$

5.3.3 Distribution of the Intrinsic Noise

The intrinsic noise is generated in the subthreshold regime of the transistors forming the decoder part of the nanowire. If N_{off} nanowires are not conducting, then $I_i = N_{\mathrm{off}} \times I_{\mathrm{off}}$ is the maximum expected intrinsic noise, assumed to be an additive constant to the total sensed current.

5.4 Test-Aware Design Optimization

The model was implemented using the bulk MOSFET model for the considered SiNWFET, as described in [49]. The linearization around the operating point was performed in the linear region, in order to keep $V_{\mathrm{DS},i}$, and consequently V_{P}, as low as possible. It is desirable to obtain a symmetrical device operation, *i.e.,* the same value of the operating point at all transistors, in order to simplify the matrices **A** and **B**. We assumed also the simple case of a binary reflexive code with the length M, and we set $V_{\mathrm{P}} = 0.9\ V$.

The thresholder parameters that we are investigating in this work are I_0 and I_1. The minimal value of I_0 has to be greater than $N \cdot I_{\mathrm{off}}$ in order to insure that $P_0 = 1$. While keeping I_0 larger than this critical value, we plotted I_1 that gives the best test quality (*i.e.,* the minimal error ε). The results are shown in Fig. 16 for different technology and design parameters. Among the considered technology parameters, β has the strongest influence on I_1. However this influence is globally weak: for $R_{\mathrm{M}} = 10\ k\Omega$, increasing β by a factor of 10, adds just 4% to I_1. It is unlikely to have both β and R_{M} large; because β increases with the nanowire width W; while the opposite happens to R_{M}. Increasing the design parameter M from 12 to 18 has less impact than increasing β by a factor of 10×, because $\sigma \sim \beta/\sqrt{M}$, showing that the dependency on M is weaker. Consequently, I_1 has a robust value $\sim 1.2 \times I_{\mathrm{on}}$ with respect to design and technology variation. On the other hand, I_0 should be large enough compared to the intrinsic noise I_{i}, but not too large, in order to separate the useful signal from the intrinsic noise. For a wide range of reasonable technological assumptions and array size, $I_0 \sim 0.66 \times I_{\mathrm{on}}$ holds.

By using these optimized thresholder parameters, we investigated the test quality under different conditions. The test quality is improved by reducing the minimum test error, as plotted in Fig. 17. As expected, the best test quality is obtained for $I_1 \sim 1.2 \times I_{\mathrm{on}}$. For a small-granularity array with $M = 12$, the test error is $\varepsilon \sim 10^{-4}$. Reducing the power level from 0.9 V down to 0.6 V reduces the current level at the operating point without reducing its variable part. Thus, it increases the noise level in the sensed current, and the test quality degrades by a factor of 22×. The variability level is the most critical parameter: increasing σ_{T} to 100 mV degrades the test quality by a factor larger than 50×. Improving the transistor gain factor β by 10× enhances the test quality by a factor of 3×. Our analytical model and results show that a better strategy is to increase the number of addressing wires M by using redundant decoders (Figs. 12 and 13).

The physical defects affecting the nanowires have been modeled at a high abstraction level as changes in nanowire addresses. A defect can cause a change of the nanowire address such that the nanowire becomes unaddressable in the considered code space, or it shares the same address with another nanowire.

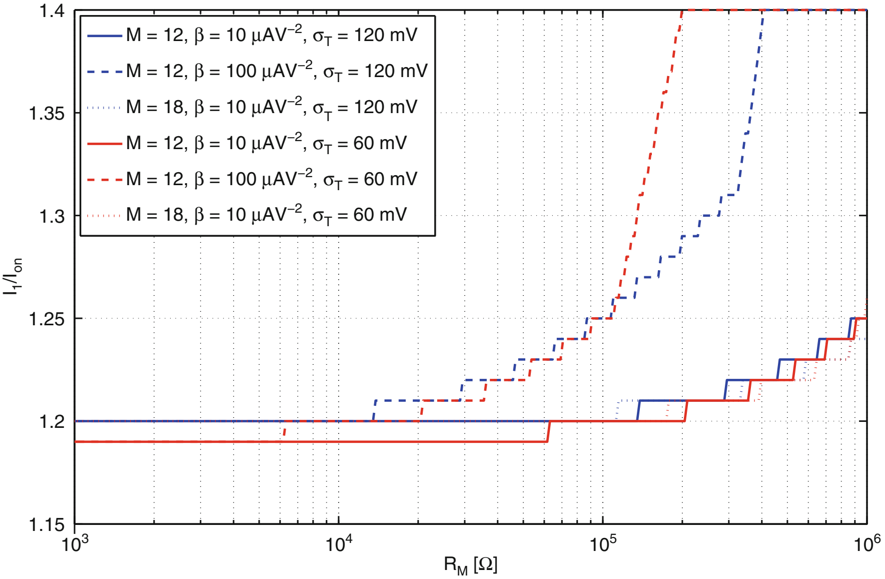


Fig. 12 Optimal value of I_1 vs. design and technology

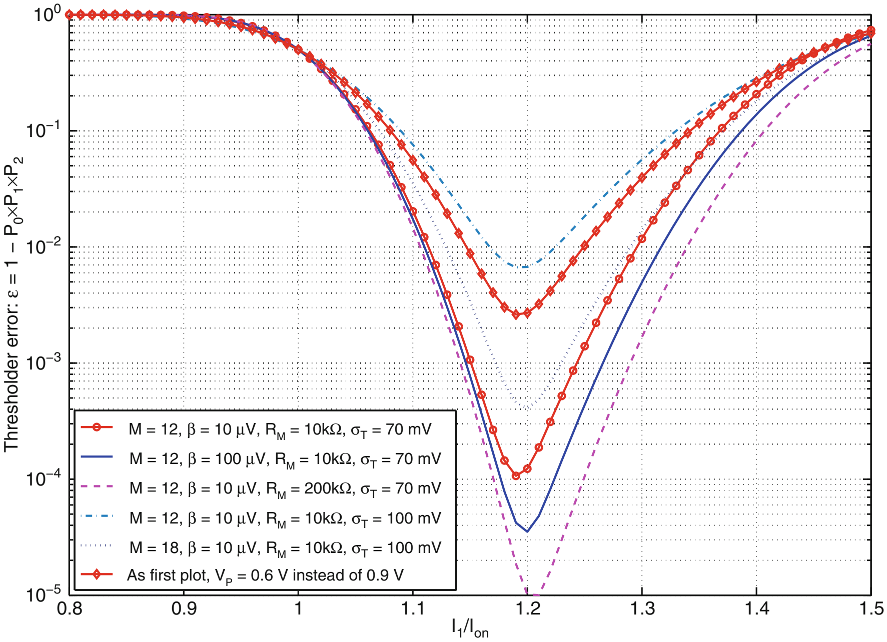


Fig. 13 Test quality vs. thresholder parameter I_1

In these cases, it is required that defective nanowire addresses be detected and discarded from the used set of addresses. This task can be performed by testing the decoder circuit.

Testing the decoder, in order to keep only defect-free parts of the code space, highly simplifies the test procedure of the whole crossbar circuit. This section proposes a test method that identifies the defective code words. The method quantifies the test quality, measured as the probability of test error, and investigates the dependency of the test quality on the decoder design parameters. Without loss of generality, crossbar circuits considered in the following discussion implement a memory function.

5.5 Testing Procedure

This section presents an overview of a test method that can be applied to nanowire arrays. This is an exhaustive method used to illustrate the testing principle. More efficient pseudo-random techniques also exist. However, the focus here is only on the thresholder design and test quality.

Nanowire testing is performed for every layer separately. We depict a single nanowire layer with its additional test circuitry in Fig. 14. Besides the nanowire layer, the system comprises the interfacing circuit (decoder) and a CMOS part formed by a thresholder, control unit and *look-up table (LUT)*. The thresholder measures the output current and indicates whether a single nanowire is detected. The control unit regulates the execution of the testing phase and other functions, such as the reading and writing operations. The LUT stores the valid addresses, *i.e.,* those that activate a single nanowire each.

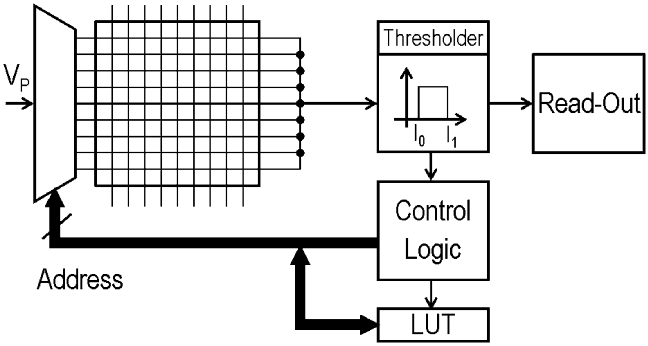


Fig. 14 Crossbar memory and testing unit: besides the memory array and the decoder, the system comprises a CMOS part formed by a thresholder that detects the bit state, a control unit that synchronizes the test operation, and a LUT that saves correct addresses

The test can be performed by applying the following exhaustive procedure. First, the two nanowire layers are disconnected by setting the power (V_P) and sense (GND) electrodes of every layer to the same voltage, such that a large voltage drop is created between the two layers. Then, each layer is considered separately. By going through all possible addresses, a voltage V_P is applied; then the address is stored in the LUT if the sensed current indicates the activation of a single nanowire. The same procedure is repeated for the second layer. The procedure is linear in N.

The output of the nanowire layer (I_s) is sensed by the thresholder. We assume that the variability mainly affects the sub-lithographic part of the memory representing the nanowire array. This part is fabricated using an unreliable technology, unlike the rest of the circuit, defined on the lithography scale and assumed to be more robust. Thus, we consider that the thresholder, the control circuit or the LUT are defect-free. The thresholder senses I_s, possibly amplifies it, then compares I_s to two reference values (I_0 and I_1 with $I_0 < I_1$). If the sensed current is smaller than I_0, then no nanowire is addressed. If the sensed current is larger than I_1, then at least two nanowires are activated with the same address. If the sensed current is between the reference current levels, then only one nanowire is activated and the address is considered to be valid. Given the statistical variation of the threshold voltages, the ability to correctly detect addresses can be expressed using the following probabilities:

$$\begin{cases} P_0 = \Pr\{(I_s \leq I_0) \text{ given that no nanowire is addressed}\} \\ P_1 = \Pr\{(I_0 < I_s < I_1) \text{ given that 1 nanowire is addressed}\} \\ P_2 = \Pr\{(I_1 \leq I_s) \text{ given that } \geq 2 \text{ nanowires are addressed}\} \end{cases} \tag{8}$$

Then, the probability that all three events occur simultaneously is given by $P_0 \times P_1 \times P_2$, assuming that the considered events are independent. We can define the error probability of this test procedure as follows:

$$\varepsilon = 1 - P_0 \times P_1 \times P_2 \tag{9}$$

The purpose of the following discussion is to design the thresholder in order to obtain the best test result with the smallest ε. Next, we derive the analytical expressions of P_0, P_1 and P_2, then we optimize I_0 and I_1 in order to minimize ε.

5.6 *Perturbative Current Model*

During the code testing phase, every nanowire is disconnected from the crossing nanowires. It can be modeled as a wire connecting the power electrode to the sensing electrode and formed by two parts (see Fig. 15): the decoder part that is a series of M pass transistors, and the memory part. Since the memory part is

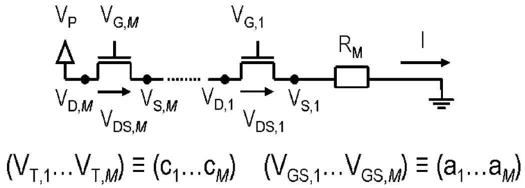

V_P
V_G,M
V_G,1
R_M
I
V_D,M
V_S,M
V_D,1
V_S,1
V_DS,M
V_DS,1
(V_T,1...V_T,M) ≡ (c_1...c_M)
(V_GS,1...V_GS,M) ≡ (a_1...a_M)

Fig. 15 Electrical parameters of a biased nanowire under test: the decoder part is represented by *M* transistors in series, and the memory part is represented by a resistance R_M. Note that the perpendicular nanowire layer is disconnected from the nanowire under test

disconnected from the second layer of nanowires, it can be modeled as a resistive load R_M. We model the devices (SiNWFETs) in this section in a general manner as voltage-controlled current sources, *i.e.*, $I = f(V_\mathrm{DS}, V_\mathrm{GS}, V_\mathrm{th})$ where I is the drain-source current, V_DS, V_GS, and V_th are, respectively, the drain-to-source, gate-to-source and threshold voltages. The decoder design is based on two different V_th's ($V_\mathrm{th,Ref0}$ and $V_\mathrm{th,Ref1}$ such that $V_\mathrm{th,Ref0} < V_\mathrm{th,Ref1}$; we define $\Delta V_\mathrm{th} = V_\mathrm{th,Ref1} - V_\mathrm{th,Ref0}$). When a nanowire is addressed, every variation of V_th results in a variation of the current through the nanowire, which can characterized as:

$$I = I^\mathrm{OP} + \delta I \tag{10}$$

Signal I is linearized around the *operating point (OP)* and divided into a *large* I^OP and a *small signal* δI. This approach is widely used in circuit and network theory and sensitivity analysis [48]. The large signal can be estimated with a SPICE simulator. The small signal can be calculated by linearizing all the equations describing the circuit around OP:

$$\delta I = -\frac{1}{R_\mathrm{M}} \cdot v^T \cdot A^{-1} \cdot B \cdot \delta V_\mathrm{th} \tag{11}$$

where variational vector $\delta\mathbf{V}_\mathrm{th} = [\delta V_\mathrm{th,1}, \cdots, \delta V_{\mathrm{th},M}]^\mathbf{T}$ for the threshold voltages, and the small signal matrices **A** and **B** are given by:

$$A = \begin{bmatrix} 1 + r_1 \cdot g_\mathrm{DS,1} & 1 & \cdots & 1 \\ 1 - r_2 \cdot g_\mathrm{m,2} & 1 + r_2 \cdot g_\mathrm{DS,2} & \cdots & 1 \\ \vdots & & & \vdots \\ 1 - r_M \cdot g_{\mathrm{m},M} & 1 - r_M \cdot g_{\mathrm{m},M} & \cdots & 1 + r_M \cdot g_{\mathrm{DS},M} \end{bmatrix},$$

$$B = \begin{bmatrix} -r_1 \cdot g_\mathrm{th,1} & 0 & \cdots & 0 \\ 0 & -r_2 \cdot g_\mathrm{th,2} & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & -r_M \cdot g_{\mathrm{th},M} \end{bmatrix}.$$

We have used the following notations: $g_{\mathrm{DS},i} = \partial f_i/\partial V_{\mathrm{DS},i}$, $g_{\mathrm{m},i} = \partial f_i/\partial V_{\mathrm{GS},i}$, $g_{\mathrm{th},i} = \partial f_i/\partial V_{\mathrm{th},i}$ and $r_i = R_{\mathrm{M}}||g_{\mathrm{m},i}^{-1}$ (parallel resistance connection). All components of matrices **A** and **B** are computed at the operating point.

5.7 Stochastic Current Model

We divide the sensed current into a useful and noisy part. The useful signal (I_{u}) is the current that flows through a nanowire when the code word corresponding to its pattern is applied. On the other hand, the noise can be generated by two different processes: intrinsically (I_{i}) or defect-induced (I_{d}). The intrinsic noise is generated by nanowires that are switched off, which generate subthreshold current. The defect-induced noise is generated by unintentionally addressed nanowires. Their number is denoted by N_{def}, while the number of nanowires generating intrinsic noise is N_{off}. Since the total number of nanowires is N, the following equation must hold: $N_{\mathrm{use}} + N_{\mathrm{off}} + N_{\mathrm{def}} = N$, where $N_{\mathrm{use}} = 0$ if no nanowire is activated by the applied code word, and 1 otherwise.

5.7.1 Distribution of the Useful Signal

Every V_{th} is considered as an independent and normally distributed stochastic variable with mean value $\overline{V}_{\mathrm{th}}$ and standard deviation σ_{th}: $V_{\mathrm{th}} \sim \mathrm{N}(\overline{V}_{\mathrm{th}}, \sigma_{\mathrm{th}}^2)$. If the nanowire pattern is correct, then the operating point of V_{th} coincides with its mean value. If a defect occurs such that the bit representing V_{th} flips, then the operating point of V_{th} is shifted from the mean value of V_{th} by $-\Delta V_{\mathrm{th}}$.

We consider a nanowire with a defect-free pattern **a**, which is controlled by its corresponding code word $\mathbf{c}^a$, and which generates the useful signal I_{u}. Then, $V_{\mathrm{th}}^{\mathrm{OP}} = \overline{V}_{\mathrm{th}}$ and $\delta\mathbf{V}_{\mathrm{th}} \sim \mathrm{N}(\mathbf{0}, \sigma_{\mathrm{th}}^2 \cdot v)$ hold. Thus, a useful signal follows the distribution resulting from (10) to (11). The operating point is the on-current, I_{on}, of the transistors, which is calculated using the SPICE simulator; whereas the variable part is given by (11). We obtain the following mean value and standard deviation of I_{u}:

$$\begin{cases} \bar{I}_{\mathrm{u}} = I_{\mathrm{on}} \\ \sigma_{\mathrm{u}} = \dfrac{\sigma_{\mathrm{th}}}{R_{\mathrm{M}}} \cdot \parallel v^T A^{-1} B \parallel \end{cases} \tag{12}$$

5.7.2 Distribution of Defect-Induced Noise

Next, we consider a nanowire NW^{b} with pattern **b** that undergoes some defects and the pattern turns into $\mathbf{b}^*$. This defective nanowire can be activated by the code word $\mathbf{c}^a$ of another nanowire NW^{a} with pattern **a**. In this case, NW^{b} generates a defect-induced noise I_{d}. This defect can be described by a series of shifts at the digits of **b** represented by vector $\mathbf{s} \in \{0, 1\}^M$, where $\Delta V_{\mathrm{th}} \cdot s_i \in \{0, \Delta V_{\mathrm{th}}\}$ indicates whether a threshold voltage shift occurred at transistor $i(i = 1, \ldots, M)$. Assuming that N_{def} nanowires generate a defect-induced noise, then every one of them is characterized by a given threshold voltage shift vector $\mathbf{s}_i$, $i \in \{1, \ldots, N_{\mathrm{def}}\}$. By applying the summation rule of independent Gaussian distributions, we obtain the following mean value and standard deviation of I_{d}:

$$\begin{cases} \bar{I}_{\mathrm{d}} = N_{\mathrm{def}} \cdot I_{\mathrm{on}} - \dfrac{\Delta V_{\mathrm{th}}}{R_{\mathrm{M}}} \cdot v^T \mathbf{A}^{-1} \mathbf{B} \cdot \sum\limits_{i=1 \ldots N_{\mathrm{def}}} \mathbf{s}_i \\ \sigma_{\mathrm{d}} = \dfrac{\sqrt{N_{\mathrm{def}}} \cdot \sigma_{\mathrm{th}}}{R_{\mathrm{M}}} \cdot \parallel \mathbf{v}^T \mathbf{A}^{-1} \mathbf{B} \parallel \end{cases} . \tag{13}$$

5.7.3 Distribution of the Intrinsic Noise

The intrinsic noise is generated in the subthreshold regime of the transistors forming the decoder part of the nanowire. If N_{off} nanowires are not conducting, then $I_i = N_{\mathrm{off}} \times I_{\mathrm{off}}$ is the maximum expected intrinsic noise, and assumed to be a an additive constant to the total sensed current.

5.8 *Test-Aware Design Optimization*

The model was implemented using the bulk MOSFET model for the considered SiNWFET, as described in [49]. The linearization around the operating point was performed in the linear region, in order to keep $V_{\mathrm{DS},i}$, and consequently V_{P}, as low as possible. It is desirable to obtain a symmetrical device operation, *i.e.,* the same value of the operating point at all transistors, in order to simplify matrices **A** and **B**. We assume the simple case of a binary reflexive code with length M, and we set $V_{\mathrm{P}} = 0.9\ V$.

The thresholder parameters that we have investigated in this work are I_0 and I_1. The minimal value of I_0 has to be greater than $N \cdot I_{\mathrm{off}}$ in order to ensure that $P_0 = 1$. While keeping I_0 larger than this critical value, we plotted I_1 that gives the best test quality (*i.e.,* minimal error ε). The results are shown in Fig. 16 for different technology and design parameters. Among the considered technology parameters, β (the transistor gain factor) has the strongest influence on I_1. However this influence is globally weak: for $R_{\mathrm{M}} = 10\ k\Omega$, increasing β by a factor of 10 adds just 4% to I_1. It is unlikely to have both β and R_{M} large; because β increases with nanowire width W, whereas the opposite happens to R_{M}. Increasing the design parameter M (the number of addressing wires) from 12 to 18 has less impact than increasing β by a factor of 10×, because $\sigma \sim \beta/\sqrt{M}$, showing that the dependency on M is weaker. Consequently, I_1 has a robust value $\sim 1.2 \times I_{\mathrm{on}}$ with respect to design and technology variation. On the other hand, I_0 should be large enough compared to intrinsic noise I_{i}, but not too large, in order to separate the useful signal

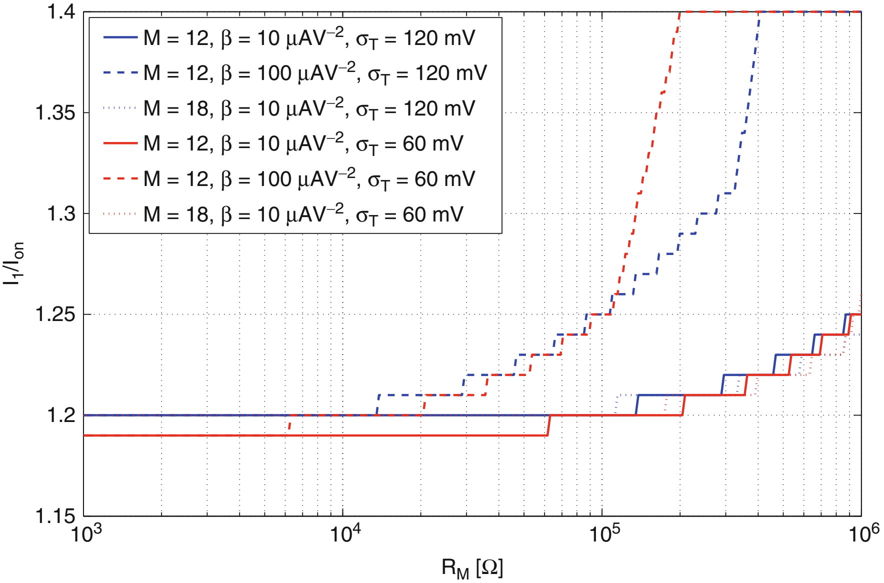


Fig. 16 Optimal value of I_1 vs. design and technology

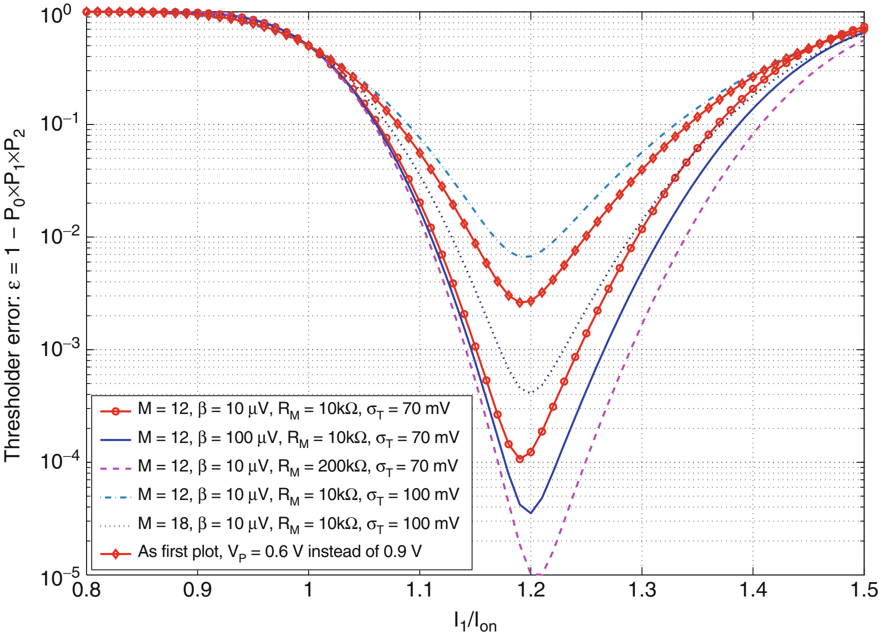


Fig. 17 Test quality vs. thresholder parameter I_1

from the intrinsic noise. For a wide range of reasonable technological assumptions and array size, $I_0 \sim 0.66 \times I_{on}$ holds.

By using these optimized thresholder parameters, we investigated the test quality under different conditions. The test quality is improved by reducing the minimum test error, as plotted in Fig. 17. As expected, the best test quality is obtained for $I_1 \sim 1.2 \times I_{on}$. For a small-granularity array with $M = 12$, the test error is $\varepsilon \sim 10^{-4}$. Reducing the power level from 0.9 V down to 0.6 V reduces the current level at the operating point without reducing its variable part. Thus, it increases the noise level in the sensed current, and the test quality degrades by a factor of 22×. The variability level is the most critical parameter: increasing σ_{th} to 100 mV degrades the test quality by a factor larger than 50×. Improving the transistor gain factor β by 10× enhances the test quality by a factor of 3×. Our analytical model and results show that a better strategy is to increase the number of addressing wires M by using redundant decoders.

6 Conclusions

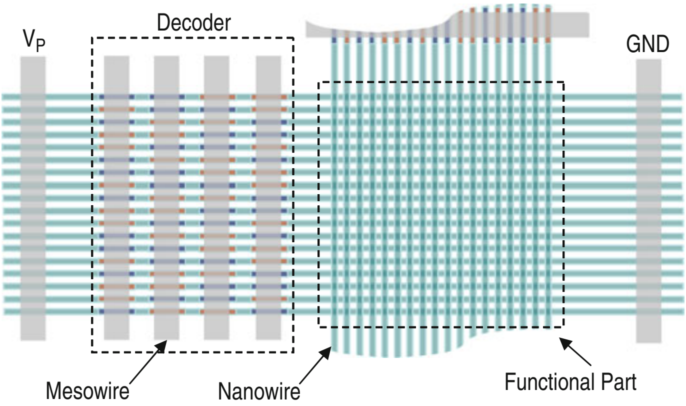
The crossbar architecture is a possible architectural paradigm for high-density integration of SiNWs into CMOS chips, and can be applied to a wide range of NW fabrication technologies. The variability of the nanowires due to their shrinking dimensions has an impact on the operation of the arrays. In this chapter, the focus was on the impact of variability on decoder design, which is the part of the circuit that bridges the array to the CMOS part of the chip. In order to address this problem, an abstract model of nanowires as a set of code words in a code space was introduced and the impact of variability was modeled as a sequence of errors that affect the code words. Based on this model, the decoder design was shown to be made more reliable by optimizing the choice of encoding scheme. Detection of errors can be performed by carrying out a test procedure. It was demonstrated that the decoder design can be optimized with respect to the test procedure in order to minimize the test error probability.

Exercise 1 Delay in a Crossbar

Consider the following *N*x*N* crossbar circuit with *N* nanowires in every plane and $M = \log_2(N)$ access transistors in the decoder of every plane. Determine the delay through the crossbar when the address corrsponding to the crosspoint (*X*,*Y*) is activated (*X* and *Y* are between 1 and *M*). Assume the following parameters:

- Decoder parameters:
  - On-resistance of an access transistor: R_{on} = 10 kΩ
  - Off-resistance of an access transistor: R_{off} = 100 MΩ
  - Drain/source capacitances of an access transistor: $C_{D/S}$ = 1 fF

- Parameters of the functional part of the crossbar
  - Resistance of a nanowire length unit equal to the nanowire pitch: RNW = 100 Ω
  - Capacitance of a molecular switch: C_S = 2 fF
  - Resistance through a molecular switch: R_S = 1 kΩ
  - Parasistic capacitance between crossing nanowires, parallel nanowires and between the nanowires and the substrate: not included



Ex. Figure 1 Baseline crossbar architecture

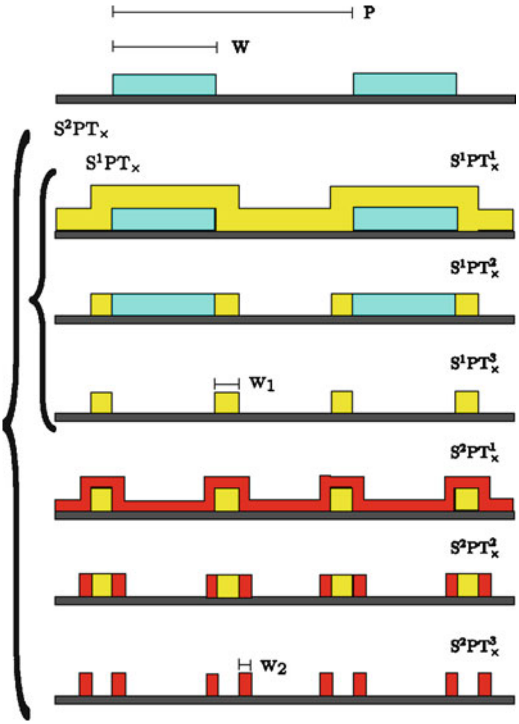
Exercise 2 Process Optimization

In goal of this exercise is to optimize the geometry of mask used in the MSPT process. The MSPT can used iteratively, starting with a given sacrificial layer, in order to define the spacers that may be used as sacrifical layers in the following steps.

This techniques envolves the deposition of a first sacrificial layer (1st step) with the width W and pitch P. Then, a sacrificial layer with the height $W_1 = qW$ is deposited (2nd step) and etched (3rd step) in order the form the sacrifical layer with the width W1 and a smaller pitch than P (4th step). These steps can be repeated with a following deposition of a layer with a height $W_2 = q\ W_1$ (step 5 to 7) in order to decrease the pitch further.

Questions taken from [50]:

1. Calculate the extension of the spacer underneath and beyond the first sacrificial layer after n iterations ($l_{out}(n)$ and $l_{in}(n)$ respectively).
2. For a large number of iterations, calculate the optimal values for q and W/P.



Ex. Figure 2 Multiplicative road of the multi-spacer technique [50]

References

1. M. H. Ben Jamaa, "Fabrication and Design of Nanoscale Regular Circuits," *PhD Thesis at Ecole Polytechnique Fédérale de Lausanne (EPFL)*, Sept. 2010.
2. R. S. Wagner and W. C. Ellis, "Vapor-liquid-solid mechanism for single crystal growth," *Applied Physics Letters*, vol. 4, no. 5, pp. 89–90, 1964.
3. Y. Cui, X. Duan, J. Hu, and C. M. Lieber, "Doping and electrical transport in silicon nanowires," *The Journal of Physical Chemistry B*, vol. 4, no. 22, pp. 5213–5216, 2000.
4. R. He and P. Yang, "Giant piezoresistance effect in silicon nanowires," *Nature Nanotechnology*, vol. 1, no. 1, pp. 42–46, 2006.
5. L. J. Lauhon, M. S. Gudiksen, D. Wang, and C. M. Lieber, "Epitaxial core-shell and core-multishell nanowire heterostructures," *Nature*, vol. 420, pp. 57–61, 2002.
6. K. E. Moselund, D. Bouvet, L. Tschuor, V. Pot, P. Dainesi, C. Eggimann, N. L. Thomas, R. Houdré, and A. M. Ionescu, "Cointegration of gate-all-around MOSFETs and local silicon-on-insulator optical waveguides on bulk silicon," *IEEE Transactions on Nanotechnology*, vol. 6, no. 1, pp. 118–125, 2007.
7. K.-N. Lee, S.-W. Jung, W.-H. Kim, M.-H. Lee, K.-S. Shin, and W.-K. Seong, "Well controlled assembly of silicon nanowires by nanowire transfer method," *Nanotechnology*, vol. 18, no. 44, p. 445302 (7pp), 2007.
8. S. D. Suk, S.-Y. Lee, S.-M. Kim, E.-J. Yoon, M.-S. Kim, M. Li, C. W. Oh, K. H. Yeo, S. H. Kim, D.-S. Shin, K.-H. Lee, H. S. Park, J. N. Han, C. Park, J.-B. Park, D.-W. Kim, D.

Park, and B.-I. Ryu, “High performance 5nm radius twin silicon nanowire MOSFET (TSNWFET): fabrication on bulk Si wafer, characteristics, and reliability,” in *Proc. IEEE Int. Electron Devices Meeting*, pp. 717–720, Dec. 2005.

9. D. Sacchetto, M. H. Ben-Jamaa, G. De Micheli, and Y. Leblebici, “Fabrication and characterization of vertically stacked gate-all-around Si nanowire FET arrays,” in *Proc. European Solid-State Device Research Conference*, 2009.
10. L. Doherty, H. Liu, and V. Milanovic, “Application of MEMS technologies to nanodevices,” *Circuits and Systems, 2003. ISCAS* ’03. Proceedings of the 2003 International Symposium on, vol. 3, pp. III–934–III–937, May 2003.
11. R. M. Y. Ng, T. Wang, and M. Chan, “A new approach to fabricate vertically stacked single-crystalline silicon nanowires,” in *Proc. IEEE Conf. on Electron Devices and Solid-State Circuits*, pp. 133–136, Dec. 2007.
12. S.-M. Koo, A. Fujiwara, J.-P. Han, E. M. Vogel, C. A. Richter, and J. E. Bonevich, “High inversion current in silicon nanowire field effect transistors,” *Nano Letters*, vol. 4, no. 11, pp. 2197–2201, 2004.
13. J. Kedzierski and J. Bokor, “Fabrication of planar silicon nanowires on silicon-on-insulator using stress limited oxidation,” *Journal of Vacuum Science and Technology B*, vol. 15, no. 6, pp. 2825–2828, 1997.
14. V. Auzelyte, H. H. Solak, Y. Ekinci, R. MacKenzie, J. Vrs, S. Olliges, and R. Spolenak, “Large area arrays of metal nanowires,” *Microelectronic Engineering*, vol. 85, no. 5–6, pp. 1131–1134, 2008.
15. O. Vazquez-Mena, G. Villanueva, V. Savu, K. Sidler, M. A. F. van den Boogaart, and J. Brugger, “Metallic nanowires by full wafer stencil lithography,” *Nano Letters*, vol. 8, no. 11, pp. 3675–3682, 2008.
16. J. Hållstedt, P.-E. Hellström, Z. Zhang, B. Malm, J. Edholm, J. Lu, S.-L. Zhang, H. Radamson, and M. Östling, “A robust spacer gate process for deca-nanometer high-frequency MOSFETs,” *Microelectronic Engineering*, vol. 83, no. 3, pp. 434–439, 2006.
17. Y.-K. Choi, J. S. Lee, J. Zhu, G. A. Somorjai, L. P. Lee, and J. Bokor, “Sublithographic nanofabrication technology for nanocatalysts and DNA chips,” *Journal of Vacuum Science Technology B: Microelectronics and Nanometer Structures*, vol. 21, pp. 2951–2955, 2003.
18. W. Wu, G.-Y. Jung, D. L. Olynick, J. Straznicky, Z. Li, X. Li, D. A. A. Ohlberg, Y. Chen, S.-Y. Wang, J. A. Liddle, W. M. Tong, and R. S. Williams, “One-kilobit cross-bar molecular memory circuits at 30-nm half-pitch fabricated by nanoimprint lithography,” *Applied Physics A: Materials Science and Processing*, vol. 80, no. 6, pp. 1173–1178, 2005.
19. G.-Y. Jung, E. Johnston-Halperin, W. Wu, Z. Yu, S.-Y. Wang, W. M. Tong, Z. Li, J. E. Green, B. A. Sheriff, A. Boukai, Y. Bunimovich, J. R. Heath, and R. S. Williams, “Circuit fabrication at 17 nm half-pitch by nanoimprint lithography,” *Nano Letters*, vol. 6, no. 3, pp. 351–354, 2006.
20. Y. Huang, X. Duan, Q. Wei, and C. M. Lieber, “Directed assembly of one-dimensional nanostructures into functional networks,” *Science*, vol. 291, no. 5504, pp. 630–633, 2001.
21. P. A. Smith, C. D. Nordquist, T. N. Jackson, T. S. Mayer, B. R. Martin, J. Mbindyo, and T. E. Mallouk, “Electric-field assisted assembly and alignment of metallic nanowires,” *Applied Physics Letters*, vol. 77, pp. 1399–1401, 2000.
22. X. Duan, Y. Huang, Y. Cui, J. Wang, and C. M. Lieber, “Indium phosphide nanowires as building blocks for nanoscale electronic and optoelectronic devices,” *Nature*, vol. 409, pp. 66–69, 2001.
23. Y. Chen, D. A. A. Ohlberg, X. Li, D. R. Stewart, R. Stanley Williams, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, D. L. Olynick, and E. Anderson, “Nanoscale molecular-switch devices fabricated by imprint lithography,” *Applied Physics Letters*, vol. 82, pp. 1610–1612, 2003.
24. J. A. Zasadzinski, R. Viswanathan, L. Madsen, J. Garnaes, and D. K. Schwartz, “Langmuir-Blodgett films,” *Science*, vol. 263, no. 5154, pp. 1726–1733, 1994.

25. N. A. Melosh, A. Boukai, F. Diana, B. Gerardot, A. Badolato, P. M. Petroff, and J. R. Heath, “Ultrahigh-density nanowire lattices and circuits,” *Science*, vol. 300, no. 5616, pp. 112–115, 2003.
26. J. E. Green, J. W. Choi, A. Boukai, Y. Bunimovich, E. Johnston-Halperin, E. Deionno, Y. Luo, B. A. Sheriff, K. Xu, Y. S. Shin, H.-R. Tseng, J. F. Stoddart, and J. R. Heath, “A 160-kilobit molecular electronic memory patterned at 10^{11} bits per square centimetre,” *Nature*, vol. 445, pp. 414–417, 2007.
27. G. Ho, J. R. Heath, M. Kondratenko, D. F. Perepichka, K. Arseneault, M. Pézolet, and M. R. Bryce, “The first studies of a tetrathiafulvalene-sigma-acceptor molecular rectifier,” *Chemistry – A European Journal*, vol. 11, no. 10, pp. 2914–2922, 2005.
28. R. L. McCreery, “Molecular electronic junctions,” *Chemistry of Materials*, vol. 16, no. 23, pp. 4477–4496, 2004.
29. G. J. Ashwell, B. Urasinska, and W. D. Tyrrell, “Molecules that mimic Schottky diodes,” *Physical Chemistry Chemical Physics (Incorporating Faraday Transactions)*, vol. 8, pp. 3314–3319, 2006.
30. Y. Luo, C. P. Collier, J. O. Jeppesen, K. A. Nielsen, E. DeIonno, G. Ho, J. Perkins, H.-R. Tseng, T. Yamamoto, J. F. Stoddart, and J. R. Heath, “Two-dimensional molecular electronics circuits,” *Journal of Chemical Physics and Physical Chemistry*, vol. 3, pp. 519–525, 2002.
31. C. P. Collier, G. Mattersteig, E. W. Wong, Y. Luo, K. Beverly, J. Sampaio, F. M. Raymo, J. F. Stoddart, and J. R. Heath, “A [2]catenane-based solid state electronically reconfigurable switch,” *Science*, vol. 289, pp. 1172–1175, 2000.
32. Y. Zhang, S. Kim, J. McVittie, H. Jagannathan, J. Ratchford, C. Chidsey, Y. Nishi, and H.-S. Wong, “An integrated phase change memory cell with Ge nanowire diode for cross-point memory,” in *Proc. IEEE Symp. on VLSI Technology*, pp. 98–99, June 2007.
33. A. DeHon, “Design of programmable interconnect for sublithographic programmable logic arrays,” in *Proc. Int. Symp. on Field-Programmable Gate Arrays*, 2005, pp. 127–137.
34. W. Culbertson, R. Amerson, R. Carter, P. Kuekes, and G. Snider, “Defect tolerance on the Teramac custom computer,” in *Proc. IEEE Symp. on FPGAs for Custom Computing Machines*, pp. 116–123, Apr. 1997.
35. J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams, “A defect-tolerant computer architecture: Opportunities for nanotechnology,” *Science*, vol. 280, no. 5370, pp. 1716–1721.
36. S. C. Goldstein and M. Budiu, “NanoFabrics: Spatial computing using molecular electronics,” in *Proc. Int. Symp. on Computer Architecture*, 2001, pp. 178–189.
37. S. Goldstein and D. Rosewater, “Digital logic using molecular electronics,” in *Proc. IEEE Int. Solid-State Circuits Conference*, vol. 1, pp. 204–459, 2002.
38. M. S. Gudiksen, L. J. Lauhon, J. Wang, D. C. Smith, and C. M. Lieber, “Growth of nanowire superlattice structures for nanoscale photonics and electronics,” *Nature*, vol. 415, pp. 617–620, 2002.
39. A. DeHon and K. K. Likharev, “Hybrid CMOS/nanoelectronic digital circuits: devices, architectures, and design automation,” in *Proc. IEEE/ACM Int. Conf. on Computer-aided Design*, 2005, pp. 375–382.
40. A. DeHon, P. Lincoln, and J. Savage, “Stochastic assembly of sublithographic nanoscale interfaces,” *IEEE Trans. on Nanotechnology*, vol. 2, no. 3, pp. 165–174, 2003.
41. J. E. Savage, E. Rachlin, A. DeHon, C. M. Lieber, and Y. Wu, “Radial addressing of nanowires,” *ACM Journal on Emerging Technologies in Computing Systems*, vol. 2, no. 2, pp. 129–154, 2006.
42. R. Beckman, E. Johnston-Halperin, Y. Luo, J. E. Green, and J. R. Heath, “Bridging dimensions: demultiplexing ultrahigh density nanowire circuits,” *Science*, vol. 310, no. 5747, pp. 465–468, 2005.
43. P. J. Kuekes and R. S. Williams, “Demultiplexer for a molecular wire crossbar network (MWCN DEMUX),” US Patent 6,256,767, 2001.

44. T. Hogg, Y. Chen, and P. Kuekes, “Assembling nanoscale circuits with randomized connections,” *IEEE Trans. on Nanotechnology*, vol. 5, no. 2, pp. 110–122, 2006.
45. E. Rachlin, “Robust nanowire decoding,” 2006. [Online]. Available: http://www.cs.brown.edu/publications/theses/masters/2006/eerac.pdf
46. D. A. Anderson and G. Metze, “Design of totally self-checking check circuits for m-out of-n codes,” in *Proc. Int. Symp. on Fault-Tolerant Computing, "Highlights from Twenty-Five Years,"* pp. 244–248, June 1995.
47. M. H. Ben Jamaa, D. Atienza, K. E. Moselund, D. Bouvet, A. M. Ionescu, Y. Leblebici, and G. De Micheli, “Variability-aware design of multilevel logic decoders for nanoscale crossbar memories,” *IEEE Trans. on Computer-Aided Design*, vol. 27, no. 11, pp. 2053–2067, Nov. 2008.
48. R. K. Brayton, *Sensitivity and Optimization*. Elsevier, 1980.
49. J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*. Prentice-Hall International Editions, 1996.
50. C. F. Cerofolini, “The multi-spacer patterning technique: a non-lithographic technique for terascale integration,” *Semiconductor Science and Technology*, 23(2008) 075020.