Decoding Nanowire Arrays Fabricated with the Multi-Spacer Patterning Technique

M. Haykel Ben Jamaa, Yusuf Leblebici and Giovanni De Micheli
Swiss Federal Institute of Technology (EPFL)
1015 Lausanne, Switzerland
haykel.benjamaa@epfl.ch

ABSTRACT

Silicon nanowires are a promising solution to address the increasing challenges of fabrication and design at the future nodes of the *Complementary Metal-Oxide-Semiconductor (CMOS)* Technology roadmap. Despite the attractive opportunity that offers their organization onto regular crossbars, the problem of designing the nanowire decoder is still challenging and highly dependent on the nanowire fabrication technology. In this paper, we introduce a novel design style and encoding scheme for decoding nanowires fabricated with the *Multi-Spacer-Patterning Technique (MSPT)*; and we present a method based on Gray codes that reduces the fabrication cost and improves the decoder reliability. We show that by arranging the code in a Gray code fashion, we decrease the fabrication complexity by 17% and the variability by 18% on average. By optimizing the decoder parameters, the simulations showed an improvement of the crossbar yield by 40% and a reduction of the effective bit area by 51% to 169 nm^2.

Categories and Subject Descriptors

B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault-Tolerance

General Terms

Design

Keywords

Emerging Technologies, Nanowires, Decoder, Crossbar, Spacer Technique, MSPT

1. INTRODUCTION

Despite the large efforts spent to develop novel solutions in order to reach the future nodes of the technology roadmap, several hurdles are still challenging the evolution of the semiconductor industry. A major issue is the high variability of nanoscale devices, which needs innovative defect tolerance methods at all design levels. In order to go beyond the 22 nm milestone, alternative devices to bulk *Metal-Oxide-Semiconductor Field Effect Transistor (MOSFET)* are currently being investigated, and their design methodologies are being optimized. Such devices include quasi one-dimensional silicon nanowires [9, 15]. A promising approach to integrate them is the crossbar architecture [10].

Crossbar circuits are formed by two perpendicular layers of parallel nanowires. At the crossing areas, specific materials are deposited or grafted in order to functionalize the circuit [10]. A major issue has been reported with the integration of crossbar circuits, consisting in interfacing them with a decoder to the outer *Complementary Metal-Oxide-Semiconductor (CMOS)* circuit, which is defined on a much larger pitch [6]. Several types of decoders have been suggested [6, 13, 1, 8]. They are strongly dependent on the nanowire fabrication technology, and some of them bridge the sublitho- to litho-scales in a stochastic way. One of the nanowire fabrication techniques is the *Multi-Spacer Patterning Technique (MSPT)*, which has many advantages in terms of density and compatibility with standard CMOS processing [4]. However, no specific design style for the decoder of MSPT-based nanowires has been proposed so far. And the impact on yield of the encoding scheme, *i.e.*, the code used to identify the nanowires, is not known.

In this paper, we propose a novel design style for decoders of MSPT-based nanowire arrays. The first novelty of this design style is that it is the first specific decoder for this fabrication technology that uniquely addresses every nanowire. The second novelty is that it assigns a deterministic address to every nanowire, unlike other decoders [6, 8]. Furthermore, we suggest an innovative design method based on Gray codes, which reduces the fabrication complexity and improves the fault-tolerance of the decoder.

This paper is organized as follows. The next section surveys different nanowire fabrication technologies and circuits, the decoder design and the nanowire codes. Section 3 presents the MSPT that is currently being studied through experimentation, and the decoder design style that we are suggesting. Section 4 formulates the problem of the decoder design in an abstract way. In Section 5 we derive the optimal code types with respect to different optimization criteria. Section 6 presents the simulation platform and results for nanowire decoders with different encoding schemes. We finally conclude the paper in Section 7.

2. BACKGROUND

This section surveys nanowire fabrication techniques reported in literature. Then, it explains the baseline architecture of nanowire crossbars and highlights the decoder part. Finally, it introduces the code types that will be used in the rest of the paper.

2.1 Fabrication Technologies

Silicon nanowires are quasi one-dimensional long structures fabricated on a sub-lithographic scale; the *lithographic* (or *meso*-) scale being the one defined by the photolithography used to convert the circuit design into physical masks. Many techniques were recently shown to be promising ways to fabricate silicon nanowires. We can distinguish between bottom-up and top-down techniques. Bottom-up techniques are based on the growth of nanowires on a silicon substrate from catalyst seeds. The as-grown nanowires are then collected in a solution and dispersed on the substrate to be functionalized [9, 15]. Top-down approaches define the nanowires

[1]This work is supported by the CCMX/MMNS project, the CSI Center and the Swiss FNS Research Grant 200021-109450/1

DAC 2009, July 26 - 31, 2009, San Francisco, California, USA.

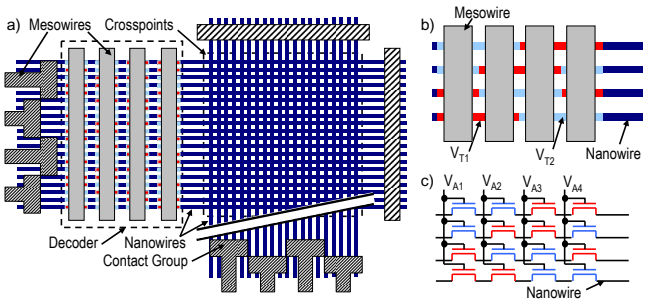

Figure 1: Baseline architecture of a crossbar circuit (a) and highlights of the decoder layout (b) and circuit (c)

directly on the functional substrate by accurately controlling the deposition, oxidation and etching rates [12, 4], or by using nanometer-scale molds [11].

Because of the small dimensions of these devices and the difficulty of placing them accurately, a high variability is expected with respect to the electrical properties of the fabricated circuits. Consequently, a general tendency is in favor of a regular arrangement of these devices in a crossbar architecture where the device placement is easier and the redundancy is higher. Since the nanowires cannot perform computation or information storage, the use of other emerging materials has been proposed to enhance the device functionality. For instance, molecular switches [10] or phase change materials [16] whose on- and off-states can be electrically controlled may be inserted between two layers of crossing nanowires in order to functionalize the circuit.

2.2 Nanowire Circuits and Decoders

The baseline organization of a nanowire crossbar circuit is depicted in Fig 1.a. 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 with a phase change material or molecular switches at the crosspoints. Information storage, interconnection or computation can be performed with these crosspoints [10, 5]. 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 can be contacted by the lithographically defined lines (mesowires).

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_{T} [2] (Fig. 1.b). The distributions of V_{T}'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. 1.c). In this case, this nanowire is said to be addressed by the applied voltage pattern.

Many decoders have been suggested for nanowire arrays. Their design strongly depends on the nanowire fabrication technology. Axial and radial decoders were suggested for in-situ patterned nanowires [6, 13]. Mask-based decoder [1] and random-contact decoders [8] were proposed for nanowires fabricated with nanoimprint. For other bottom-up fabrication techniques, a gate-all-around decoder was suggested in [12].

2.3 Code Types

We will consider in this paper *hot codes (HC)*, *tree codes (TC)*, *Gray codes (GC)* and *balanced Gray codes (BGC)*. More details about these codes can be found in [2, 7, 3]. In the following, we remind the reader of their basic concepts. A code space means the set of all code words of a given code.

HCs are defined for the multi-valued logic n with the parameters $(M, k) \in \mathbb{N}^2$ such that $M = k \cdot n$. Every code word has M digits, such that every digit takes the value $\in \{0, \ldots, n-1\}$ and every value exists in the code word exactly k times. For instance, the code words 001122 and 012120 belong to the same HC space with $(M, k) = (6, 2)$ and $n = 3$; but 000121 does not belong to this code space because the values 0 and 2 are not repeated exactly k= 2 times.

TCs are defined by the parameter $M \in \mathbb{N}$ for the multi-valued logic n. Every code is represented by M digits such that every digit takes the value in $\in \{0, \ldots, n-1\}$. For instance, for $n = 3$ and $M = 4$, we obtain the codes: 0000, 0001, 0002, 0010,. . . ,2222. In order to address nanowires with TCs, it was shown that the codes have to be *reflected* [2], *i.e.*, every code word gets its complementary code word appended to it. The complement is obtained by subtracting the code word from the largest code word in the same code space. For instance, the complement of 0010 in this example is $2222 - 0010 = 2212$. Then, 2212 is appended to 0010, giving the reflected code word 00102212. In a similar way, 0000 and 0001 give 00002222 and 00012221 respectively. In the rest of the paper, all TCs are *implicitly* considered to be reflected.

GCs are an arrangement of TCs where successive code words differ in only one digit. For instance, the sequence of code words $0000 \Rightarrow 0001 \Rightarrow 0002 \Rightarrow 0010$ does not belong to the GC because the last two code words differ in two digits. But the sequence $0000 \Rightarrow 0001 \Rightarrow 0002 \Rightarrow 0012$ is eligible.

BGCs are a type of GCs with an additional constraint that every digit is allowed to change at most a given number of times. In the BGCs considered in this paper, the limit on number of changes is set to 2 [3]. In the previously presented GC sequence $0000 \Rightarrow 0001 \Rightarrow 0002 \Rightarrow 0012$, the first and second digits (from the left) do not change, the third digit changes once and the last digit changes twice. Thus, this sequence fulfills the conditions of a BGCs with a limit set to 2.

GCs and BGCs are special arrangements of code words in TC spaces. So, GCs and BGCs will be used *implicitly* in the reflected form by appending their complement, as explained previously for reflected TCs.

3. FABRICATION AND DESIGN OPPORTUNITIES OF NANOWIRE DECODERS

In this section we present the nanowire array fabrication technology that we are currently experimentally investigating. Then, we introduce the decoder fabrication method and we focus on the decoder design opportunities and challenges.

3.1 The Multi-Spacer Patterning Technique

In order to fabricate nanowire arrays with a pitch independent on the photolithography limit, we are currently using the *Multi-Spacer Patterning Technique (MSPT)*, which is a CMOS-compatible fabrication technique using only standard CMOS steps. It yields large arrays of parallel *poly-crystalline Silicon (poly-Si)* spacers separated by *silicon dioxide (SiO_2)* as insulator. The essential fabrication steps are illustrated in Fig. 2 and were described in detail elsewhere [4]. The processing starts with the definition of caves limited by a sacrificial layer. Then, by a series of conformal deposition of thin poly-Si and SiO_2 layers and their anisotropic etching iteratively, it is possible to obtain parallel poly-Si spacers separated by SiO_2 spacers. The technique yields a symmetrical structure with a symmetry axis parallel to the edge of the sacrificial layer.

With the 0.8 μm photolithography available for academic research purposes in our fabrication facilities, we could obtain dense arrays of poly-Si spacers with a sub-lithographic pitch. These spacers represent nanowires having a pitch of a few tens of nanometer, a height of $\sim$ 300 nm and a length of tens of microns. Since the height does not influence the pitch, it was kept as it is; but it can be shrunk by applying the standard chemical and mechanical planarization technique if needed. A *Scanning Electron Microscopy (SEM)* of a cross-section of the obtained nanowire arrays is presented in Fig. 3. The nanowire pitch exclusively depends on the thickness of deposited poly-Si and on the etch, but not on the lithography resolution. The technique yields reliable and reproducible results. It is not intended to compete with bottom-up nanowire fabrication techniques, but it represents a CMOS-compatible and

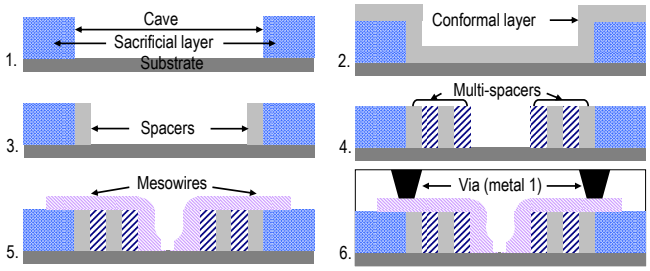

Figure 2: Cave cross-section illustrating the multi-spacer patterning technique. Cave definition (1), conformal poly-Si deposition (2), anisotropic poly-Si etch (3), iteration of 2-3 with poly-Si and SiO_2 (4), gate patterning (5), metallization (6)

cost-efficient technique (possible parallelization of spacer definition steps) with an opportunity to investigate the combined design and fabrication aspects of nanowire decoders.

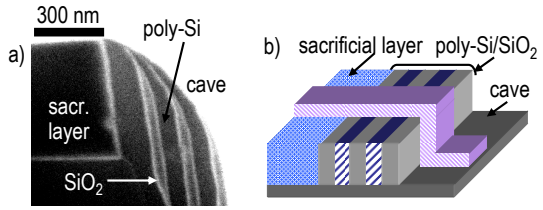

Figure 3: Scanning Electron Microscopy (SEM) of a nanowire array cross-section (a), and a 3-dimensional representation of the full structure (b).

3.2 The Decoder Fabrication Technique

While a pattern can be easily defined during the growth of nanowires in a bottom-up process (Sec. 2.1), it is more difficult to define it with top-down processes. For instance, the MSPT, yields a regular array of undifferentiated nanowires if the bare procedure depicted in Fig. 2 is applied. Once the array is defined on a sub-lithographic scale, it is difficult to pattern it with standard photolithographic means, unless expensive high-resolution and time-costly methods, such as electron-beam lithography, are applied. Consequently, it is desirable to pattern the nanowires while they are defined: *i.e.*, whenever a new spacer is defined, it has to be patterned before the next spacer is defined.

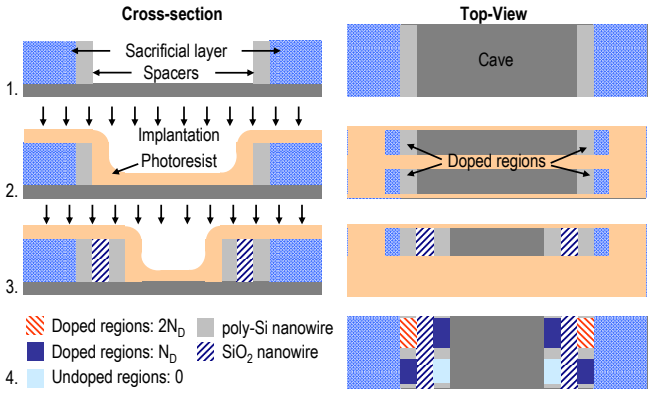

Figure 4: Decoder-aware enhanced fabrication flow: Definition of first poly-Si nanowire (1), photolithography and doping of first poly-Si nanowire (2), photolithography and doping of second poly-Si nanowire (3), final doping patterns (4).

The fabrication flow that includes the decoder is illustrated in Fig. 4 and should be understood as an extension inserted between steps 3 and 4 in Fig. 2. The other steps remain unchanged. The additional steps are lithography patterning and doping after every spacer definition step, using p-type (n-type) doping to increase (decrease) the total doping level. Specific regions from every poly-Si nanowire are defined and doped in this way. Nanowires are fragile and thin structures, and they should be doped carefully with light doses. The total doping level of a specific region is the sum of all (positive and negative) doping levels accumulated in this region throughout the definition of the whole array, as illustrated in step 4 of Fig. 4. An optimized choice of the lithography/doping sequences and the doping doses may result in the desired nanowire pattern.

3.3 Design of the Decoder

The decoder fabrication technique introduced in the previous section yields a decoder operation identical to the description in Sec. 2.2. However, the layout differs in the fact the the nanowires lie within parallel caves having a symmetry axis going through their central axis (Sec. 3.1). The unique addressing of every nanowire in a half cave insures the unique addressing of every nanowire in the whole array. We will therefore consider only half caves in the rest of the paper.

Every half cave contains N nanowires having M doping regions each. Recall that the pattern is the sequence of threshold voltages and the doping profile is the sequence of dopant concentrations along the doping regions of the nanowire. Let $\mathbf{P}_i = \left[P_i^0 \dots P_i^{M-1}\right]$ and $\mathbf{D}_i = \left[D_i^0 \dots D_i^{M-1}\right]$ be the pattern and doping profile of the nanowire i respectively. For the considered technique, whenever a nanowire i is patterned by receiving a doping dose, all nanowires $k = 0, \dots, i-1$ receive the same doping dose simultaneously. Consequently, the doping profile of a nanowire i depends not only on its own doping dose but also on all doping doses received by the nanowires $k = i+1, \dots, M-1$. We therefore need to determine the analytical multivariable application that links P_i^j and D_i^j for $i = 0, \dots, N-1$ and $j = 0, \dots, M-1$, in order to specify whether we can find a set of doping profiles that results in a given set of patterns.

Assuming that a set of doping profiles exists for any set of patterns, it is possible to optimize the choice of patterns according to different cost functions. We consider first the impact of this decoding technique on the fabrication cost. The nanowire profile implies a certain number of lithography/doping steps per nanowire, ϕ_i for $i = 0, \dots, N-1$. From the fabrication point of view, it is of the highest importance to reduce the total number of lithography/doping steps, *i.e.* $\sum \phi_i$. We therefore need to establish the link between P_i^j and ϕ_i in order to minimize $\sum \phi_i$.

Then, we consider the impact of this decoding technique on the circuit yield by analyzing the variability of the decoder. Every doping region j of the nanowire i, referred to as region (i, j), receives successive doping doses bit by bit. With every additional doping dose, the variability of region (i, j), quantified as the standard deviation of the threshold voltage of this region Σ_i^j, accordingly increases. It is therefore desirable to establish the link between P_i^j and Σ_i^j and to optimize the choice of the patterns in order to minimize the variability.

In the following section, we will derive the analytical mapping between patterns and doping profiles. Then, we will define the cost functions related to the fabrication complexity and to the threshold voltage variability. These cost functions will be minimized by choosing the best code type for the decoder.

4. PROBLEM FORMULATION OF MSPT-BASED NANOWIRE DECODER

In this section we provide an abstract description of the decoder part of the nanowire array in a single half cave. The defined matrices describe the most relevant design and fabrication aspects. Using these definitions, we will derive the cost functions of the fabrication complexity and the circuit variability. Further, we assume a multi-valued logic addressing with n values.

Definition 1. The pattern matrix $\mathbf{P}$ is an N × M matrix in $\{0,\ldots,n-1\}^{\mathrm{N}\times\mathrm{M}}$ representing the patterns of N nanowires within a half cave, where every nanowire has M doping regions.

We assume that the N nanowires within every half cave are patterned and have M doping regions each. The pattern corresponds to a set of M V_{T}'s having any one of the n possible values $V_{\mathrm{T}}(0),\ldots,V_{\mathrm{T}}(n-1)$. The patterns are represented by the discrete values $0,\ldots,n-1$, which correspond to the ordered discretization of the threshold voltages $V_{\mathrm{T}}(0),\ldots,V_{\mathrm{T}}(n-1)$. Consequently, the set of patterns on the N nanowires forming one half cave can be represented by an N × M-matrix in $\{0,\ldots,n-1\}^{\mathrm{N}\times\mathrm{M}}$.

Definition 2. The final doping matrix $\mathbf{D}$ is an N × M matrix in $\mathbb{R}^{\mathrm{N}\times\mathrm{M}}$ representing the doping level distribution along the N nanowires within a half cave after the definition of the whole array.

Every V_{T} needs a unique doping level N_{D} fixed by the device physics and geometry [14]. Consequently, the pattern matrix, that is uniquely mapped onto a set of V_{T}'s, defines a unique final doping matrix.

Example 1. For $n=3$, $N=3$ and $M=4$, we assume that V_{T} can have the values 0.1 V, 0.3 V and 0.5 V corresponding to the digits 0, 1 and 2 and to the doping levels 2, 4 and 9 $\times 10^{18}$ cm^{-3}. The patterns are represented with the first N code words of the n-ary tree code. With $\mathbf{V}$ the matrix covering all V_{T}'s, we obtain:

$$\mathbf{P}=\begin{bmatrix}0&1&2&1\\0&2&2&0\\1&0&1&2\end{bmatrix}\quad \mathbf{V}=\begin{bmatrix}1&3&5&3\\1&5&5&1\\3&1&3&5\end{bmatrix}\cdot 0.1\,\mathrm{V}\quad \mathbf{D}=\begin{bmatrix}2&4&9&4\\2&9&9&2\\4&2&4&9\end{bmatrix}\cdot\frac{10^{18}}{\mathrm{cm}^3}$$

PROPOSITION 1. *A non-linear bijective application* h *maps* $\mathbf{P}$ *onto* $\mathbf{D}$ *as follows:* $D_i^j=\mathrm{h}(P_i^j)\ \forall\, i,j$

PROOF. The mapping between digits of the patterns and V_{T} is a discrete ordering, which is a bijective application g. The mapping between V_{T}'s and N_{D}'s is a monotonic non-linear function f, which is also a a bijection. The interested reader is invited to look into [14] to obtain the exact expression of f. Since h is a composition of f and g, it is bijective as well. □

Definition 3. The step doping matrix $\mathbf{S}$ is an N × M matrix in $\mathbb{R}^{\mathrm{N}\times\mathrm{M}}$ representing the additional doping levels after every lithography/doping step.

There is a lithography/doping procedure that follows the definition of every one of the N nanowires. Every procedure $i=0,\ldots,n-1$ is characterized by M doping levels $\left[N_{\mathrm{D},i}^0,\ldots,N_{\mathrm{D},i}^{M-1}\right]$ along the M doping regions of the nanowires. The set of M doping levels in the N steps can be represented by the matrix $\mathbf{S}$ in $\mathbb{R}^{\mathrm{N}\times\mathrm{M}}$.

PROPOSITION 2. *A multi-linear application maps the elements of* $\mathbf{S}$ *onto those of* $\mathbf{D}$ *as follows:* $D_i^j=\sum_{k=i}^{N-1}S_k^j$

PROOF. Every nanowire j that is defined, is subsequently patterned by means of doping doses $\left[S_j^0,\ldots,S_j^{M-1}\right]$. Any nanowire i defined before the nanowire j ($i<j$) receives the same dose simultaneously. Thus, the doping level of the nanowire i is the sum of all the levels defined in the steps $i,\ldots,N-1$ following the definition of the nanowire i. □

Example 2. The following step and final doping matrices verify the property stated in Proposition 2. Negative and positive doping levels correspond to the doses with n- and p-type dopants respectively:

$$\mathbf{D}=\begin{bmatrix}2&4&9&4\\2&9&9&2\\4&2&4&9\end{bmatrix}\cdot\frac{10^{18}}{\mathrm{cm}^3}\quad \mathbf{S}=\begin{bmatrix}0&-5&0&2\\-2&7&5&-7\\4&2&4&9\end{bmatrix}\cdot\frac{10^{18}}{\mathrm{cm}^3}$$

Definition 4. The technology complexity is quantified by Φ representing the total number of additional lithography/doping steps needed to pattern the nanowires.

Every row in $\mathbf{S}$ ($\mathbf{S}_i=\left[S_i^0\ldots S_i^{M-1}\right], i=0,\ldots,N-1$) represents the doping doses used in a single step doping procedure. The number of unequal non-zero elements in $\mathbf{S}_i$ represents the number of different doses used at this doping step. The more doping doses, the more lithography steps are needed and the more complex is the fabrication. Let ϕ_i ($i=0,\ldots,N-1$) be the number of unequal non-zero elements in $\mathbf{S}_i$, then the total number of lithography/doping steps is $\Phi=\sum_i\phi_i$.

Example 3. For $\mathbf{S}$ given in Example 2, we have: $\phi_1=2$, $\phi_2=4$ and $\phi_3=3$. Then, $\Phi=9$ holds.

Definition 5. The decoder variability is quantified by a N × M matrix Σ, describing the standard deviation of the threshold voltages in every doping region in the decoder of a half cave.

Every doping operation yields a V_{T} with a given variability σ_{T}, measured as the standard deviation. In the proposed technique every doping region is doped at most N times (Proposition 2). We expect the variability to increase with increasing number of doping operations. The number of times a doping regions (i,j) receives a doping dose decreases with increasing i and increasing number of zero-elements in the column j of $\mathbf{S}$. Let ν_i^j be this number, then $\nu_i^j=\sum_{k=i\ldots N-1}(1-\delta(S_k^j))$, where $\delta(x)$ is the *Kronecker delta function*: $\delta(x)=1\Leftrightarrow x=0$, otherwise $\delta(x)=0$. Doping operations are assumed to be stochastically independent. The addition of two independent stochastic variables with standard deviations σ_1 and σ_2 respectively yields a stochastic variable with the standard deviation $\sigma_0=\sqrt{\sigma_1^2+\sigma_2^2}$. Therefore, if we define Σ as the N × M-matrix describing the variability of the decoder by setting Σ_i^j to the square of the standard deviation of the doping region (i,j), we obtain: $\Sigma_i^j=\sigma_T^2\cdot\nu_i^j$.

Example 4. For $\mathbf{S}$ given in Example 2, we have:

$$\mathbf{S}=\begin{bmatrix}0&-5&0&2\\-2&7&5&-7\\4&2&4&9\end{bmatrix}\cdot\frac{10^{18}}{\mathrm{cm}^3}\quad \mathbf{\Sigma}=\begin{bmatrix}2&3&2&3\\2&2&2&2\\1&1&1&1\end{bmatrix}\cdot\sigma_{\mathrm{T}}^2$$

PROPOSITION 3. *Optimizing the decoder fabrication complexity consists in finding the best pattern* $\mathbf{P}$ *that minimizes* Φ. *Optimizing the decoder reliability consists in finding the best pattern* $\mathbf{P}$ *that minimizes* $\|\Sigma\|_1$, *where* $\|\Sigma\|_1$ *is the sum of all elements of* Σ, *known as its entrywise 1-norm.*

PROOF. This follows directly from Def. 4 and 5. □

5. OPTIMIZING NANOWIRE CODES

In Sec. 2.3 we reviewed the code types that we are considering to uniquely address the nanowires in any logic with n values: hot codes, tree codes and the arrangement of the tree code words called Gray codes and arranged Gray codes. The last 3 code types are used implicitly in a reflected form. The length of the whole code word–including the reflected part–is M.

In the following, we will prove that the Gray code is the optimal arrangement of the tree code with respect to the defined cost functions. Then, we will investigate the opportunity of arranging the hot codes in a similar fashion to Gray codes in order to optimize the costs of the decoders designed with hot codes.

5.1 The Gray Code

PROPOSITION 4. *Among all arrangements of tree codes, the Gray code minimizes the decoder cost in terms of variability* $\|\Sigma\|_1$.

PROOF. For $i=N-1$, $S_{N-1}^j=D_{N-1}^j=h^{-1}(P_{N-1}^j)$ is fixed by the pattern of the last nanowire, *i.e.* by P_{N-1}^j. Thus, $\nu_{N-1}^j=1-\delta(S_{N-1}^j)=1$, because $S_{N-1}^j\neq 0\ \forall\, j$, since every region receives a doping dose in order to define the pattern of V_{T}'s of the last nanowire P_{N-1}^j. For $i\neq N-1$, $\nu_i^j-\nu_{i+1}^j=1-\delta(S_i^j)=1-\delta(D_i^j-D_{i+1}^j)$. This difference is 1 if $D_i^j\neq D_{i+1}^j$, *i.e.* $P_i^j\neq$

P^j_{i+1}, and 0 if $P^j_i = P^j_{i+1}$. Then, ν^j_i can only increase by steps of 1 or remain constant with decreasing i for a fixed j. It remains unchanged if and only if the pattern P^j_i remains unchanged.

Consequently, $\|\Sigma\|_1$ monotonically increases with increasing transitions in the pattern matrix $\mathbf{P}$ between every two successive rows. Given that the rows of $\mathbf{P}$ are the code words in the chosen code space, it is desirable to use the code that minimizes the number of transitions between successive code words in order to minimize $\|\Sigma\|_1$. This condition is fulfilled by the Gray code. □

Example 5. Instead of $\mathbf{P}$ given in Example 1, which includes a tree code sequence with the cost $\|\Sigma\|_1 = 22 \cdot \sigma^2_{\mathrm{T}}$ (from Example 4), we use a sequence from the Gray code that avoids the forbidden transition in $\mathbf{P}$ 0220⇒1012. Then we obtain $\|\Sigma\|_1 = 18 \cdot \sigma^2_{\mathrm{T}}$:

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 2 & 1 \\ 0 & 2 & 2 & 0 \\ 1 & 2 & 1 & 0 \end{bmatrix} \mathbf{S} = \begin{bmatrix} 0 & -5 & 0 & 2 \\ -2 & 0 & 5 & 0 \\ 4 & 9 & 4 & 2 \end{bmatrix} \cdot \frac{10^{18}}{\mathrm{cm}^3} \quad \mathbf{\Sigma} = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \sigma^2_{\mathrm{T}}$$

PROPOSITION 5. *Among all arrangements of tree codes, the Gray code minimizes the fabrication cost* Φ.

PROOF. In a similar way to the proof of Proposition 4, we notice that the value of S^j_i is unequal to zero if there is a transition $P^j_i \Rightarrow P^j_{i+1}$ between two successive code words in $\mathbf{P}$ at the digit j. Then, any ϕ_i, and consequently Φ, increase with the number of transitions in $\mathbf{P}$. Since the Gray code minimizes the number of transitions, then it is optimal with respect to Φ. □

Example 6. The Gray code in Example 1 has a fabrication cost $\Phi = 9$ (Example 3). By using the Gray code in Example 5, the fabrication cost was reduced to $\Phi = 7$ ($\phi_1 = 2$, $\phi_2 = 2$ and $\phi_3 = 3$).

5.2 Arranged Hot Codes

The previous section demonstrated that the arrangement of the TC words into a sequence that minimizes the number of transitions between every successive code words, defined a the GC, which minimizes the decoder variability and the fabrication cost. We therefore considered the question whether the code words of HCs can be arranged in a similar way to the GC, such that the number of transitions is minimized, and to assess the possible benefits of such codes, that we called *Arranged Hot Code (AHC)*.

Since the number of values in every HC word (M, k) is fixed (Sec. 2.3), then the minimum number of transitions is 2. We used an exhaustive algorithm for most of the hot codes with a reasonable code space size ($\lesssim 100$) for nanowire arrays, and we found that the arrangement in a *Gray-code-fashion* always exists.

It is possible to show in a very similar way to Proposition 4 and 5 that, when an arrangement of a given hot code exists, in such a way that the number of transitions between every successive code words is minimized, then this arrangement is the optimal hot code with respect to $\|\Sigma\|_1$ and Φ compared to all possible arrangements of the same hot code. In the next section we will therefore assess the performance of the optimized versions of both tree and hot codes in terms of fabrication complexity and circuit costs.

6. SIMULATIONS OF THE DECODER

In this section, the yield and area of crossbar circuits are simulated for different code types and the improvement resulting from the optimized decoder design (defined as its size and the code type) is demonstrated.

6.1 Simulation Platform

In order to assess the impact of the decoder design (meaning the choice of the code type) on the fabrication complexity and the circuit features, we performed a statistical analysis of a crossbar circuit (Fig. 1.a). The function of the crossbar circuit was assumed to be a memory. The defects affecting the molecular switches or the phase change layer were not simulated. Only the defects happening at the decoder part due to the variability of the V_{T}'s in the doping regions were considered. We neglected the probability that nanowires can be broken; we actually noticed that the fabricated nanowires (Fig. 3.a) had a yield close to unit.

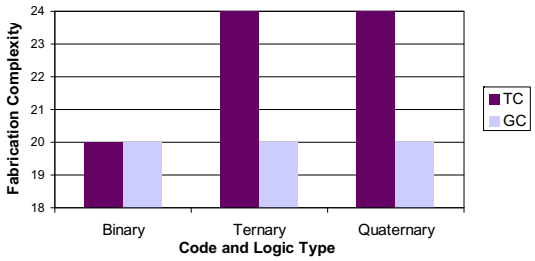

Figure 5: Fabrication complexity in terms of number of additional steps for different code types and logic levels

Both layers forming the crossbar have the same number of caves for a square-shaped crossbar. The cave count and the number of nanowires in every half cave N was fixed according to the raw crosspoint density set to D_{RAW}=16 kB. The number of contact groups per half cave was minimized with respect to the code type (code space size Ω and code length M) and geometry (lithography pitch P_{L} and nanowire pitch P_{N}). While Ω and M were used as simulation parameters, P_{L} was set to 32 nm and P_{N} to 10 nm. According to standard layout rules, the minimum width of every contact group had to be set to $1.5 \times P_{\mathrm{L}}$. The maximum width of every contact group was limited by the width of Ω nanowires at most that can fit in every contact group.

The threshold voltages V_{T}'s are distributed within the range 0 to 1 V, in order to account for a maximum supply voltage of 1 V. The doping levels were estimated from V_{T}'s by using the assumptions in [14] for the most common materials used in standard CMOS processes. The variability σ_{T} of V_{T} was set to 50 mV. A nanowire is addressable if V_{T} at every doping region varies within a small range as specified in [2]. In this way, the probability that a nanowire is addressed was calculated from the Gaussian distributions of V_{T}'s with the standard deviations given by Σ. We accounted for nanowires that may be addressed by two adjacent contact groups, as explained in [6], and we removed them from the set of addressable nanowires. This gives the estimate for the yield of every cave Y. Consequently, the effective array density that denotes the number of working crosspoints can be estimated as: $D_{\mathrm{EFF}} = D_{\mathrm{RAW}} \cdot Y^2$.

6.2 Simulation Results

We calculated the technology complexity Φ for different code and logic types. The results, plotted in Fig. 5 for $N = 10$ show that Φ is constant for all binary codes and equal to the double of the number of nanowires in a half cave. Higher logic level was suggested as a way to reduce the area overhead of the decoder [2]. However, Fig. 5 shows that the higher logic level comes with some fabrication cost: 20% more steps for the tree code. For ternary and quaternary logic, the Gray code performs better than the tree code (17%) by completely canceling the fabrication complexity overhead.

The variability matrix was calculated for various types of binary codes. N was set to 20 and the plots in Fig. 6 show the variability level at every digit in the $N \times M$-matrix Σ, as square roots of elements of Σ normalized to σ_{T}. By comparing Fig. 6.a, c and e, we see that the Gray code and its balanced version reduce the variability level at every digit in comparison to the tree code. The balanced Gray code distributes the variability more evenly than the other codes. In this way, the average variability $\|\Sigma\|_1/(N \cdot M)$ could be reduced by 18%. Similar results were obtained for these codes with a higher logic level, as well as for hot codes and their arranged version. Next, we compared the distribution of the elements of Σ for a fixed code type and different code lengths (Fig. 6.a, c and e vs. Fig. 6.b, d and f). We noticed that longer codes have less digit transitions and help reduce the average variability.

The elements of Σ provide the inputs to estimate the crossbar yield, that we quantified as the effective crossbar density normalized by the raw crossbar density of 16 kB. The crossbar yield

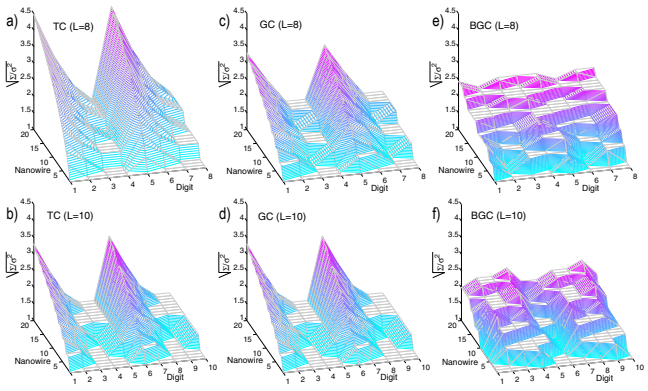

Figure 6: Square root of elements of variability matrix Σ for different binary code types and lengths

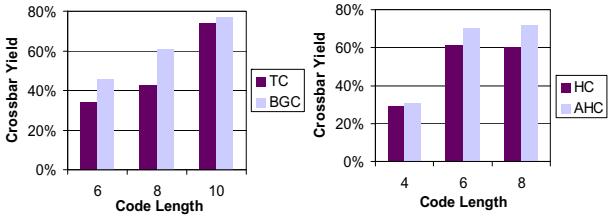

Figure 7: Crossbar yield in terms of percentage of addressable crosspoints for different binary code types and lengths

is plotted in Fig. 7 for various binary code types and lengths. The yield generally increases with increasing code length, until it reaches the maximum ($M \sim 10$ for TC/BGC and $M \sim 6$ for HC/AHC). The yield improvement of the tree code and the arranged hot codes, by increasing the code length from 6 to 10 and 4 to 8 respectively, is $\sim 40\%$. For a fixed code length, the optimized codes (*i.e.* BGC and AHC) perform better than their non-optimized versions (*i.e.* TC and HC respectively). For instance, the balanced Gray code yields 42% more than the tree code, and the arranged hot code 19% better than the hot code with the same length $M = 8$.

The dependency of the yield on the code length is explained by two factors: *i)* the variation with M of the percentage of nanowires at adjacent contact groups, which have to be removed from the set of addressable nanowires; and *ii)* the dependency of the variability on M. On one hand, by increasing M and keeping the code type fixed, the code space size increases and less contact groups are needed, so less nanowires are removed at adjacent contact groups and the yield increases. This effect saturates when the code space size is large enough to neglect the number of nanowires at adjacent contact groups. On the other hand, the average variability $\|\Sigma\|_1/(N \cdot M)$ decreases with increasing M, because longer codes have less digit transitions, as we showed previously. So, for a fixed code type, the yield, first, increases with increasing M, then it starts decreasing for larger M because the increasing number of digits cancels the benefits of decreasing variability of each digit taken separately. This decrease is just slightly seen for the hot code when M increases beyond 6; and for other code types, it starts appearing from $M \sim 10$.

From the geometrical data, we estimated the crossbar area; then, by considering the effective density, we estimated the bit area for different code types and lengths (Fig. 8). For the tree code and its optimized versions (Gray and balanced Gray codes), the bit area decreases with increasing code length mainly due to the vanishing effect of adjacent contact regions, as explained previously. An area saving by 51% can be achieved by setting M to 10 instead of 6 for the tree code. The balanced Gray code yields a denser crossbar than the Gray code, which in turns yields better than the tree code; for instance crossbars with the balanced Gray code are 30% denser than those with the tree code (for $M = 8$). The hot and arranged hot codes yield the most dense crossbars with $M = 6$.

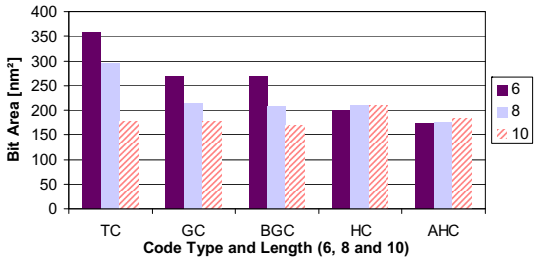

Figure 8: Average area per functional bit for different binary code types and lengths

For larger M, the bit area slightly start to increase, as the yield starts to decrease because of higher variability of longer codes, as explained previously. The arranged hot code performs better than the hot code with 13% less bit area for $M = 6$. Among all these codes, the smallest bit area is 169 nm^2 for the balanced Gray code, followed by the arranged hot code with 175 nm^2.

7. CONCLUSIONS

Silicon nanowires are promising devices that may help tackle many design challenges by means of their organization into regular crossbar circuits. In this paper we presented our validated fabrication technique, the MSPT, that yields very dense parallel arrays of silicon nanowires and we introduced a design style for the nanowire decoder using the same fabrication technique. We identified the design challenges for the MSPT-decoder that result from the interdependence of the nanowire patterns. We quantified these challenges in terms of fabrication cost and decoder variability. Then we proved that the arrangement of the code space in a Gray code fashion minimizes these costs. We performed simulations with different decoder design parameters and code types, including the Gray code, the balanced Gray code and the arranged hot codes. The results showed that the technology cost can be reduced by 17% by using the Gray code. The Gray code and its balanced version help reduce and balance the variability throughout the whole nanowire decoder, resulting in 18% less variability. The decoder design covers not only the code type but also its length: the yield improves by 40% by adding redundancy to the code length and by 19 to 42% by using the optimized code types. The smallest effective area per bit was found to be around 170 nm^2 for the optimized codes: the balanced Gray code and the arranged hot code.

8. REFERENCES

[1] R. Beckman et al. Bridging dimensions: Demultiplexing ultrahigh density nanowire circuits. *Science*, 310(5747):465–468, 2005.

[2] M. H. Ben Jamaa et al. Variability-aware design of multi-level logic decoders for nanoscale crossbar memories. *Trans. CAD*, 27(11):2053–2067, Nov. 2008.

[3] G. S. Bhat and C. D. Savage. Balanced Gray codes. *The Electronic Journal of Combinatorics*, 3(1):R25, 1996.

[4] G. Cerofolini. Realistic limits to computation. II. The technological side. *Applied Physics A*, 86(1):31–42, 2007.

[5] A. DeHon. Design of programmable interconnect for sublithographic programmable logic arrays. In *Proc. Int. Symp. on FPGA*, pages 127–137, 2005.

[6] A. DeHon et al. Stochastic assembly of sublithographic nanoscale interfaces. *IEEE Trans. on Nanotechnology*, 2(3):165–174, 2003.

[7] F. Gray. Pulse code communication. *US Patent No. 2632058*, 1953.

[8] T. Hogg et al. Assembling nanoscale circuits with randomized connections. *Trans. on Nanotechnology*, 5(2):110–122, 2006.

[9] J. D. Holmes et al. Control of thickness and orientation of solution-grown silicon nanowires. *Science*, 287(5457):1471–1473, 2000.

[10] Y. Luo et al. Two-dimensional molecular electronics circuits. *ChemPhysChem*, 3:519–525, 2002.

[11] N. A. Melosh et al. Ultrahigh-density nanowire lattices and circuits. *Science*, 300(5616):112–115, 2003.

[12] K. E. Moselund et al. Prospects for logic-on-a-wire. *Microelectronic Engineering*, (85):1406–1409, 2008.

[13] J. E. Savage et al. Radial addressing of nanowires. *Journal on Emerging Technologies in Computing Systems*, 2(2):129–154, 2006.

[14] S. M. Sze and K. K. Ng. *Physics of Semiconductor Devices*. 2007.

[15] D. Whang et al. Large-scale hierarchical organization of nanowire arrays for integrated nanosystems. *Nano Letters*, 3(9):1255–1259, 2003.

[16] Y. Zhang et al. An integrated phase change memory cell with ge nanowire diode for cross-point memory. *VLSI Technology*, pages 98–99, June 2007.