Address Generation for Nanowire Decoders

Jia Wang Ming-Yang Kao Hai Zhou
Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60208, USA

ABSTRACT

Nanoscale crossbars built from nanowires can form high density memories and programmable logic devices. To integrate such nanoscale devices with other circuits, nanowire decoders are needed. Due to the stochastic output of the nanoscale fabrication, the decoder addresses to select the nanowires must be generated after fabrication. In this paper, we develop a mathematical model of the nanowire decoders for the generation of the proper addresses. Assuming a simple testing approach called on-off measurement, we prove that the maximum number of the proper addresses can be generated in finite time. We design the algorithms to generate a required number of the proper addresses. Experimental results confirm the efficiency of our algorithms.

Categories and Subject Descriptors: J.6 [Computer-Aided Engineering]: Computer-Aided Design
General Terms: Algorithms
Keywords: Nanowire, Decoder, Testing

1. INTRODUCTION

Nanoelectronics is one of the emerging technologies to enable the fabrication of ultra high density electronic devices beyond the current CMOS technology. Prototype nano scale devices were created in laboratories [1, 2, 3] and system architectures were explored in literatures [4]. Due to the difficulty of fabricating interconnects precisely in the current nanotechnology and the high defect rate of the nanoelectronic [5], leveraging the mature CMOS technology is promising to build a heterogeneous system with high device density, reliability, and yield.

Nanowire crossbar is a nanoscale structure that provides ultra high density memory functionality and can be connected to CMOS circuits through nanowire decoders. Such structure can be configured to become a programmable logic device with FPGA-like functionality [4]. A nanowire crossbar consists of two perpendicular nanowire arrays. Special molecules are inserted to the the cross-points of the two arrays to form bistable junctions. The nanowires in an array are connected to external CMOS interconnects through a nanowire decoder. Once proper addresses are applied to the decoders, individual nanowires can be selected and then voltages can be applied to individual junctions to change or sense their states.

As summarized by Rachlin et al. [6], there are a few methods to construct nanowire decoders. The common practice is to have a mechanism to allow a single CMOS wire, i.e. mesowire, to control a subset of the nanowires by raising the resistance of them when an electric field is applied on that mesowire. Generally speaking, more nanowires are individually addressable if there are more mesowires. The problem of determining the necessary number of the mesowires such that most likely the required number of the nanowires are individually addressable was studied by many research works, e.g., for randomized–contact decoders, Chen et al. [7] conducted an empirical study and Rachlin et al. [6] provided a theoretical bound. In this paper, we focus on the problem of generating a required number of proper addresses that address the nanowires individually. The motivation is that if we consider the tremendous number of chips and decoders where those proper addresses should be generated after fabrication, to design efficient address generation algorithms is a critical issue for volume production. Similar problems had been studied by Chen et al. [7]. They proposed a heuristic approach for address generation of randomized–contact decoders with a single contact group by performing random testing trails. The heuristic can generate most proper addresses efficiently. However, if all the proper addresses are required to be generated, special configurable junctions are necessary. Without such special junctions, the heuristic cannot guarantee to generate the required number of the proper addresses in finite time when such proper addresses are presented.

Our contribution in this paper is that we design the algorithms that can generate most proper addresses efficiently while guarantee to generate all the proper addresses in finite time if required, without the special junctions. We develop a mathematical model first for the nanowire decoders and then the concept of *measurement* to model the testing approaches that inspect the nanowire decoders. Assuming a simple testing approach called *on-off* measurement, we design an algorithm for the nanowire decoders with a single contact group. Assuming the "Take What You Get" addressing strategy [6] with randomized–contact decoders, we extend the algorithm to handle multiple contact groups. Experimental results show that our algorithms can generate the required number of the proper addresses efficiently in terms of the number of the measurements performed. Note that the problems investigated by Rachlin et al. [6] are different from address generation – the existence of the proper addresses does not mean that they can be generated.

2. REVIEW OF NANOWIRE DECODERS

The detailed review of nanowire decoders can be found in [6]. An example nanowire decoder is shown in Figure 1, which is a part of a nanowire crossbar structure. There are another set of nanowires (not shown in the figure) perpendicular to the shown ones connected to another nanowire decoder. The components and interconnects in the decoder except the nanowires are fabricated via the current CMOS technology. The nanowires shown are grouped into contact groups and connected to ohmic contacts, shown as "oc_0" to "oc_3". The demultiplexer "demux" activate the nanowires

GLSVLSI'07, March 11–13, 2007, Stresa-Lago Maggiore, Italy.

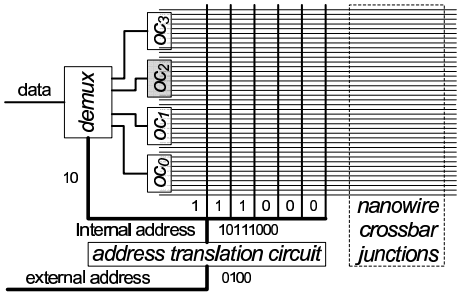

Figure 1: A nanowire decoder. Narrow wires are nanowires. Other wires are mesowires and buses.

in one particular contact group. Each mesowire perpendicular to the nanowires controls a subset of the nanowires: if a "1" is applied to the mesowire, the resistances of the controlled nanowires are raised. We assume an *ideal* control where the resistance of one nanowire is either 0 if there is no "1" applying to the controlling mesowires or $+\infty$ otherwise. The "address translation circuit" (ATC) translates external addresses into internal addresses that select individual contact groups and nanowires. As suggested in [6], the ATC could be the most area consuming component because the addresses are stored in the ATC. We assume a "Take What You Get" addressing strategy so that the area overhead of the ATC is minimal. The internal address bus is separated into two part: one chooses a contact group and the other controls the nanowires. For example, suppose the ATC translates an external address "0100" into an internal address "10111000". The highest two bits "10" activates "oc_2" and the lowest six bits "111000" selects some nanowires from that contact group.

3. PROBLEM FORMULATION

3.1 Modeling Nanowire Decoders

We use the set theory for defining terminologies, formulating problems, and providing proofs. Suppose there are OC contact groups. Let M be the set of the mesowires. For the i'th contact group, where $1 \leq i \leq \text{OC}$, let N_i be the set of the nanowires in the contact group. Let $\mathcal{M}$ and $\mathcal{N}_i$ be the power sets of M and N_i respectively, i.e.,

$$\mathcal{M} \triangleq \{w : w \subseteq M\} \text{ and } \mathcal{N}_i \triangleq \{u : u \subseteq N_i\}, \forall 1 \leq i \leq \text{OC}.$$

A *nano address* $w \in \mathcal{M}$, abbreviated as *address* when there is no ambiguity, is a pattern of activated mesowires, i.e. "1" is applied to the mesowire m if $m \in w$ or "0" is applied otherwise. The function $K_i : \mathcal{M} \to \mathcal{N}_i$ defines the *on-set*s of the addresses in the i'th contact group: $K_i(w)$ is the set of the nanowires in the i'th contact group whose resistance is 0 when the address w is applied. Assume $K_i(\emptyset) = N_i$. It is straightforward that

$$K_i(x \cup y) = K_i(x) \cap K_i(y). \tag{1}$$

We can use a set of addresses to control the nanowires in a contact group if every nanowire belongs to at most one on-set of those addresses. Formally, such sets of addresses are defined as *proper* sets of addresses, or abbreviated as proper addresses. For the i'th contact group, a set of addresses W is proper iff: first, $\forall w \in W$, $K_i(w) \neq \emptyset$; second, $\forall x, y \in W$, $K_i(x) \cap K_i(y) = \emptyset$.

According to Eq. (1), $K_i(w)$ for any w can be computed if $K_i(\{m\})$ is known for every $m \in M$. Then proper addresses can be generated. However, it is not cost effective and practical to obtain all the $K_i(\{m\})$. Instead of requiring the exact K_i, partial information regarding K_i can be obtained by performing *measurement*s. We are interested in a measurement with highly limited ability, called the *on-off* measurement, which can be implement in a similar way as proposed in [7] but requires no special junction. Once an address is given, the on-off measurement determines if all the resistances of the nanowires are $+\infty$ in a particular contact group. Formally, the on-off measurement in the i'th contact group is defined by a function $f_i : \mathcal{M} \to \{0, 1\}$ such that $f_i(w) = 0$ if $K_i(w) \neq \emptyset$ and $f(w) = 1$ if $K_i(w) = \emptyset$. One important property of f_i is that it is *non-decreasing*,

LEMMA 1. *$\forall x, y \in \mathcal{M}$, if $x \subseteq y$, then $f_i(x) \leq f_i(y)$.*

Proper addresses can be redefined with f_i:

THEOREM 1 (PROPER SET OF ADDRESSES). *For the i'th contact group, a set W of addresses is proper iff $f_i(w) = 0, \forall w \in W$ and $f_i(x \cup y) = 1, \forall x, y \in W$.*

3.2 Problem Definition

We formulate the *Nano Address Generation* problem as follows.

PROBLEM 1 (NANO ADDRESS GENERATION). *Suppose M is a finite set representing mesowires and $\mathcal{M}$ is the power set of M. There are OC contact groups. Non-decreasing functions $f_i : \mathcal{M} \to \{0, 1\}$ represent the on-off measurement for the i'th contact group, i.e. $f_i(x) \leq f_i(y), \forall x \subseteq y$ where $x, y \in \mathcal{M}$. Let maxA be the number of the proper addresses to be generated and to be stored in the ATC. For the i'th contact group, generate a set of proper addresses W_i, i.e., $W_i \subseteq \mathcal{M}$ satisfying that $f_i(w) = 0, \forall w \in W_i$ and $f_i(x \cup y) = 1, \forall x, y \in W_i$, such that either $\sum_{i=1}^{OC} |W_i| \geq maxA$ or $\sum_{i=1}^{OC} |W_i|$ is the maximum.*

In the Nano Address Generation problem, the details concerning the nanowires are hidden via the functions f_i. The only requirement is that f_i should be non-decreasing. The advantage of the abstraction is that the model and the approach in this paper may be applied to similar problems while the disadvantage is that more specific but efficient algorithms could be designed with more details. The nanotechnology is in its infancy. We prefer the abstraction because it could be proper for or even could guide the future development of the nanoscale systems.

4. ADDRESS GENERATION ALGORITHMS

4.1 Maximal Addresses

For the ease of presentation, an address w is called *blocking* for the i'th contact group if $f_i(w) = 1$ and *non-blocking* if $f_i(w) = 0$. For a non-blocking address w, if $\forall w \subset x$, $x \in \mathcal{M}$, the address x is blocking, then w is defined as a *maximal* address. Denote the set of all the maximal addresses in the i'th contact group by $W_i^\top$, i.e.,

$$W_i^\top \triangleq \{w : (f_i(w) = 0) \wedge (\forall w \subset x : f_i(x) = 1)\}. \tag{2}$$

For any $x, y \in W_i^\top$ where $x \neq y$, we must have $x \subset x \cup y$ – otherwise $x = x \cup y$ and then $y \subset x$, which contradicts that y is a maximal address. Thus $f(x \cup y) = 1$ from the definition of the maximal addresses. So,

LEMMA 2. *$W_i^\top$ is a proper set of addresses for the i'th contact group.*

It is straightforward that for every non-blocking address, there is a maximal address containing it. If there is a set of more than $|W_i^\top|$ non-blocking addresses for the i'th contact group, according to the pigeonhole principle, there must be

two of them, say x and y, and a maximal address a such that $x \subseteq a$ and $y \subseteq a$. Thus $x \cup y \subseteq a$. Then $f_i(x \cup y) \leq f_i(a) = 0$ and therefore $f_i(x \cup y) = 0$. This set of addresses is not proper as stated in Lemma 3.

LEMMA 3. *A proper set of addresses for the i'th contact group contains no more than $|W_i^\top|$ elements.*

According to Lemma 2 and 3, all the $W_i^\top$ are the solution of the Nano Address Generation problem.

THEOREM 2. *For the i'th contact group, $1 \leq i \leq OC$, the set of all the maximal addresses $W_i^\top$ is a proper set of addresses with the maximum number of elements. For all the contact groups, $\sum_{i=1}^{OC} |W_i^\top|$ is the maximum number of all the proper addresses.*

4.2 Partial Maximal Address Generation

First we design the *MaxExpand* subroutine as shown in Figure 2 that returns a maximal address w' for the i'th contact group such that $w \subseteq w'$. Note that the order to enumerate the $a \in M - w$ on line 2 will not affect the correctness of the subroutine but may result in different w' since w could be the subset of multiple maximal addresses.

Subroutine MaxExpand(w, i)	
1	$w' \leftarrow w$
2	**For** each $a \in M - w$:
3	**If** $f_i(w' \cup \{a\}) = 0$:
4	$w' \leftarrow w' \cup \{a\}$
5	Return w'.

Figure 2: The MaxExpand subroutine.

Then we design the Partial Maximal Address Generation algorithm as shown in Figure 3. For the i'th contact group, the algorithm returns a set of the maximal addresses A'. The key idea is to generate a non-blocking address w such that MaxExpand(w, i) always returns a maximal address that is not generated already. Lemma 4 and Lemma 5 summarize the idea.

LEMMA 4. *In the i'th contact group, for a set of the maximal addresses A', if a non-blocking address w satisfies that $w \not\subseteq a, \forall a \in A'$, then MaxExpand$(w, i)$ returns a maximal address $w' \notin A'$.*

LEMMA 5. *In the i'th contact group, for a set of the maximal addresses A', if $A' \neq W_i^\top$, then there is a non-blocking address w satisfying that $w \not\subseteq a, \forall a \in A'$.*

The condition $a \subset w$ on line 3 is used to save a measurement $f_i(w)$ since we must have $f_i(w) = 1$ according to the definition of the maximal addresses. The correctness of the Partial Maximal Address Generation algorithm is stated in Theorem 3.

THEOREM 3. *For the i'th contact group, the Partial Maximal Address Generation algorithm terminates and when it terminates, A' is a set of the maximal addresses. If maxA' $< |W_i^\top|$, then $|A'| =$ maxA'; otherwise $A' = W_i^\top$.*

The order to enumerate the elements of $\mathcal{M}$ will not affect the correctness of the algorithm. However, it will affect the practical running time in terms of the number of the measurements performed since the measurement of a blocking address could be avoided if there is a maximal address generated being its subset. Thus we enumerate the element w of $\mathcal{M}$ with smaller $|w|$ first in our implementation.

Algorithm Partial Maximal Address Generation **Inputs** i and maxA'. **Outputs** A'.	
1	$A' \leftarrow \emptyset$
2	**For** each $w \in \mathcal{M}$:
3	**If** $\exists a \in A'$ such that $w \subset a$ or $a \subset w$:
4	Continue the **For** loop.
5	**If** $f(w) = 0$:
6	$w' \leftarrow$MaxExpand(w, i)
7	$A' \leftarrow A' \cup \{w'\}$
8	Terminate **If** $\lvert A' \rvert =$ maxA'.

Figure 3: The Partial Maximal Address Generation algorithm.

4.3 Maximal Address Generation

According to Theorem 2, Problem 1 can be solved by applying the Partial Maximal Address Generation algorithm to each contact group separately with the parameter maxA' set to $+\infty$. However,the experimental results in Section 5 show that to generate most of the maximal addresses is efficient but to generate all of them requires significantly more measurements. Intuitively, to guarantee the yield of the nanowire decoders, the required number of the proper addresses to be generated, i.e. maxA, will be smaller than $\sum_{i=1}^{OC} |W_i^\top|$ for most decoders. The number of the measurements performed can be reduced by requiring most but not all the maximal addresses to be generated for most contact groups. Since it is impossible to determine the number of the proper addresses to be generated for each contact group in advance, we propose to interleave the address enumeration and the address generation for all the contact groups in the Maximal Address Generation algorithm as shown in Figure 4. Although for the worse case the number of the measurements performed will be the same as that of applying the Partial Maximal Address Generation algorithm separately, the average number of the measurements performed is reduced.

Algorithm Maximal Address Generation **Inputs** maxA. **Outputs** A.	
1	$A \leftarrow \emptyset$
2	**For** each $w \in \mathcal{M}$:
3	**For** $i = 1$ to OC:
4	**For** each $(i', a) \in A$ such that $i' = i$:
5	**If** $w \subset a$ or $a \subset w$:
6	Continue the **For** loop on line 3.
7	**If** $f_i(w) = 0$:
8	$w' \leftarrow$MaxExpand(w, i)
9	$A \leftarrow A \cup \{(i, w')\}$
10	Terminate **If** $\lvert A \rvert =$ maxA.

Figure 4: The Maximal Address Generation algorithm.

The Maximal Address Generation algorithm solves the Nano Address Generation problem as stated in Theorem 4.

THEOREM 4. *The Maximal Address Generation algorithm terminates and when it terminates, A is a set of the maximal addresses. If maxA $< \sum_{i=1}^{OC} |W_i^\top|$, then $|A| =$ maxA; otherwise $A = \bigcup_{i=1}^{OC} W_i^\top$.*

For the same reason as in the Partial Maximal Address Generation algorithm, we implement the enumeration of the elements of $\mathcal{M}$ by visiting the element w with smaller $|w|$ first. Most proper addresses are generated efficiently in terms of the measurements performed.

Table 1: Simulations for Single Contact Group.

$\|M\|$	$\|N\|$=30			$\|N\|$=20			$\|N\|$=15		
	maxA	# meas.	# fail.	maxA	# meas.	# fail.	maxA	# meas.	# fail.
512	450	33560	0	250	7312	19	80	957	26
	400	14519	0	200	3822	3	60	706	6
	350	9758	0	150	2481	0	40	475	3
256	240	21664	0	160	4933	20	80	1213	36
	200	6244	0	120	2309	0	60	771	4
	160	4059	0	80	1367	0	40	489	1
128	120	5212	0	100	5986	38	60	1028	38
	100	2982	0	80	1614	1	40	526	3
	80	2233	0	60	1082	0	20	265	0

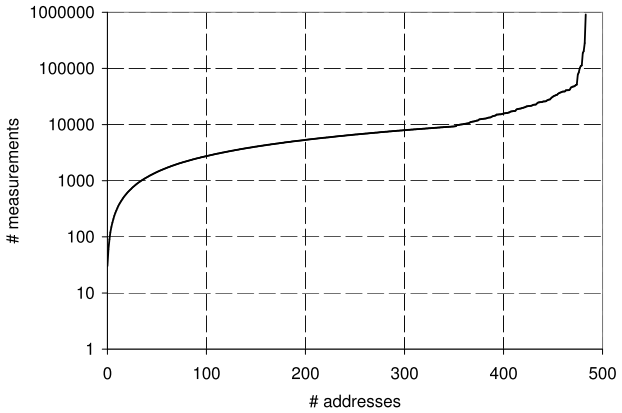

Figure 5: The number of the measurements vs. the number of proper addresses generated for a nanowire decoder with single contact group, 512 **nanowires, and** 30 **mesowires.**

5. EXPERIMENTAL RESULTS

We implement a simulation framework for the randomized–contact decoders and the proposed algorithms in C++. The simulation framework takes OC, $|M|$, and $|N_i|$ where $1 \leq i \leq \text{OC}$ as the inputs and builds the nanowire decoder by assigning $K_i(\{m\}), \forall 1 \leq i \leq \text{OC}, m \in M$ randomly. We assume that mesowires control nanowires independently with a probability p, i.e, the elements of N_i are assigned to each $K_i(\{m\})$ with a probability p. In our simulations, we set $p = 0.5$ and set all the $|N_i|$ to be equal to the same value denoted by $|N|$.

We first experiment with a single contact group. Different combinations of $|M|$, $|N|$, and maxA are simulated 100 times each and the results are reported in Table 1. The columns "# fail." show the number of the simulations where there are less than maxA proper addresses. The columns "# meas." show the average number of the measurements performed among the simulations with at least maxA proper addresses. Figure 5 shows the curve between the number of proper addresses generated and the number of the measurements performed for one of the simulations with $|M| = 30$ and $|N| = 512$. It shows that to generate most of the maximal addresses is efficient but to generate all of them requires significantly more measurements.

We then experiment with multiple contact groups. We use the same example as in [6] where $|M| = 16$, $|N| = 8$, and maxA $= 1024$. Different values of OC are simulated 100 times each and the results are reported in Table 2 in a similar format as in Table 1. Two approaches are compared: for the columns "MAG", the Maximal Address Generation algorithm is applied; for the columns "PMAG", the Partial Maximal Address Generation algorithm is applied to each contact group separately with maxA$'$ set to $+\infty$. It shows that when maxA $< \sum_{i=1}^{\text{OC}} |W_i^{\top}|$, the Maximal Address Generation algorithm is much more efficient than applying the Partial Maximal Address Generation algorithm separately. It is interesting that the number of the measurements decreases with the increase of the number of the contact groups for the Maximal Address Generation algorithm.

Table 2: Simulations for Multiple Contact Groups where $|M| = 16$, $|N| = 8$, and maxA $= 1024$.

OC	MAG		PMAG	
	# meas.	# fail.	# meas.	# fail.
200	16742	0	7836616	0
180	16912	0	7840608	0
160	17250	0	7843691	0
140	31408	0	7843442	0

6. CONCLUSION

In this paper, we developed a mathematical model of the nanowire decoders for the problem of proper address generation. We proved that the maximum number of the proper addresses can be generated via on-off measurement by generating all the maximal addresses. We designed algorithms for such purpose and investigated the running time efficiency in terms of the number of the measurements performed via simulations. The experimental results showed that the proposed algorithms generate the required number of the proper addresses efficiently.

7. REFERENCES

[1] R. S. Williams and P. J. Kuekes. Demultiplexer for a molecular wire crossbar network. US Patent Number 6,256,767, 2001.

[2] Y. Luo, J. E. Green, R. Beckman, E. Johnston-Halperin, and J. R. Heath. Bridging dimensions: Demultiplexing ultrahigh-density nanowire circuits. *Science*, 310:465–468, 2005.

[3] E. W. Wong, Y. Luo, K. Beverly, J. Sampaio, F. Raymo, J. F. Stoddart, C. P. Collier, G. Mattersteig, and J. R. Heath. A [2]catenate-based solid state electronically reconfigurable switch. *Science*, 290:1172–1175, 2000.

[4] A. DeHon and M. J. Wilson. Nanowire-based sublithographic programmable logic arrays. In *Intl. Symp. on FPGA*, pages 123–132, 2004.

[5] M. Mishra and S. C. Goldstein. Defect tolerance at the end of the roadmap. In *International Test Conference*, pages 1201–1210, 2003.

[6] E. Rachlin and J. E. Savage. Nanowire addressing with randomized–contact decoders. In *ICCAD*, pages 735–742, 2006.

[7] Y. Chen, T. Hogg, and P. J. Kuekes. Assembling nanoscale circuits with randomized connections. *IEEE Trans. on Nanotechnology*, 5(2):110–122, 2006.