# Four-dimensional address topology for circuits with stacked multilayer crossbar arrays

Dmitri B. Strukov[1] and R. Stanley Williams

Hewlett-Packard Laboratories, 1501 Page Mill Road, MS1123, Palo Alto, CA 94304

Edited by Konstantin Likharev, Stony Brook University, Stony Brook, NY, and accepted by the Editorial Board September 22, 2009 (received for review June 22, 2009)

**We present a topological framework that provides a simple yet powerful electronic circuit architecture for constructing and using multilayer crossbar arrays, allowing a significantly increased integration density of memristive crosspoint devices beyond the scaling limits of lateral feature sizes. The truly remarkable feature of such circuits, which is an extension of the CMOL (Cmos + MOLecular-scale devices) concept for an area-like interface to a three-dimensional system, is that a large-feature-size complimentary metal-oxide-semiconductor (CMOS) substrate can provide high-density interconnects to multiple crossbar layers through a single set of vertical vias. The physical locations of the memristive devices are mapped to a four-dimensional logical address space such that unique access from the CMOS substrate is provided to every device in a stacked array of crossbars. This hybrid architecture is compatible with digital memories, field-programmable gate arrays, and biologically inspired adaptive networks and with state-of-the-art integrated circuit foundries.**

digital memory | hybrid circuits | three-dimensional integrated circuits

Building three-dimensional circuits is a natural but so far limited way of increasing integration density to circumvent the inevitable stall in lateral device scaling (1, 2). One of the major challenges for any such system is to maintain a sufficiently high density of vertical interconnect to provide high-bandwidth and low-latency communication to and from each layer in the stack without sacrificing so much area within each layer for vias to negate the advantages of stacking. In this article, we show how this problem can be avoided by using a hybrid complimentary metal-oxide-semiconductor (CMOS)/crossbar circuit to implement an "area interface" that utilizes a four-element logical address to specify each physical memory device. Previous approaches to three-dimensional circuitry have been limited by the requirement to integrate active components up the vertical stack (3, 4). However, multilayer CMOS circuits have many obstacles—thin-film transistor technologies have poor performance characteristics for memory and logic applications, whereas three-dimensional wafer bonding suffers from low interconnect density because of limitations to alignment between wafers (i.e., as compared with that of photolithographic masks defining features on a single wafer for multiple metallization layers) and from poor cost efficiency (5).

Here, we describe an architecture, based on hybrid circuits composed of a conventional CMOS layer connected to multiple crossbar layers that contain memristive devices. A memristor is a two-terminal electrical circuit element that changes its resistance depending on the total amount of charge that flows through the device (6, 7). A memristance arises naturally in thin-film semiconductors for which electronic and dopant equations of motion are coupled in the presence of an applied electric field (8). This property is actually common for nanoscale films and has been observed in a variety of material systems (e.g., transition metal oxides and perovskites, various superionic conductors composed of chalcogenides and metal electrodes, and organic polymer films) (9, 10). Although memristance was observed experimentally for at least 50 years before it was recognized as such, it now has become interesting for a variety of digital and analog applications, especially because a true memristor does not lose its state when the electrical power is turned off.

Other key advantages of memristive devices are their small footprints—on the order of $4F^2$, where $F$ is the lithographic feature size (or half-pitch)—and relatively simple structures that are easily fabricated and integrated with conventional CMOS processes. However, memristive devices are not active components (e.g., the equivalent of the CMOS transistor), because they cannot supply energy to a circuit. The solution to that problem is to complement crossbar arrays of memristive devices with a conventional CMOS layer that provides signal restoration and gain but is much less dense. Such a concept, named CMOL (Cmos + MOLecular-scale devices), was proposed originally in the context of nonphotolithographic techniques (11) for a single crossbar layer (i.e., to make higher-density two-dimensional circuits). We show here how to construct hybrid circuits with multiple layers of crossbars that are all addressed with a single set of vertical vias to amplify the density advantage of the memristive switches. Each memory bit requires four labels to specify its location (the four-dimensional address) instead of the usual two required for two-dimensional arrays. The additional complexity in dealing with the larger logical address is more than compensated by the fact that the number of memristive devices that can be addressed scales as the fourth power of the number of transistors in the CMOS circuitry instead of the second power, which dramatically increases the number of devices that can be addressed within a fixed area. Various applications, such as digital memories, field-programmable gate arrays (FPGAs), and even some exotic applications, including synaptic networks, should benefit from this hybrid architecture.

**From Two-Dimensional to Three-Dimensional Hybrid Circuits.** Geometrically regular circuits, such as digital memories, even can be optimized efficiently without the use of design automation tools. For instance, Fig. 1*A* shows a typical array topology used for various memories that renders a simple and compact circuit layout. The memory device, represented by a green dot, depends on the type of technology (i.e., is it a capacitor, variable capacitor, floating gate transistor, four transistor feedback loop circuit, or magnetic tunnel junction in dynamic random access memory (RAM), ferroelectric RAM, flash, static RAM, or magnetoresistive RAM memories, respectively). The read/write operations may be unique for each memory technology, but in general a read operation involves sensing a physical quantity, such as charge, used to store state in a particular device. Accessing a particular memory device in a square array requires the selection of one word line and one bit line (out of $N$ total each) to establish electrical connections between the desired memory cell and the peripheral input/output circuitry.

Author contributions: D.B.S. designed research; D.B.S. and R.S.W. performed research; and D.B.S. and R.S.W. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission. K.L. is a guest editor invited by the Editorial Board.

Freely available online through the PNAS open access option.

[1] To whom correspondence should be addressed. E-mail: strukov@gmail.com.

www.pnas.org/cgi/doi/10.1073/pnas.0906949106

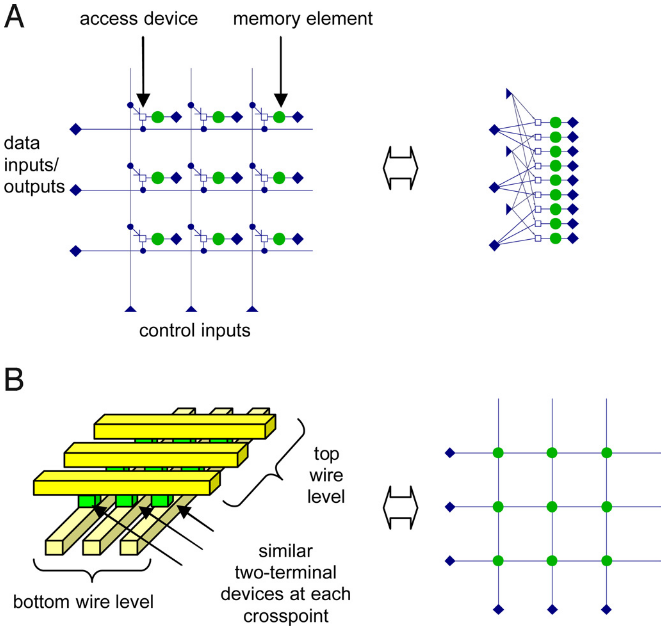

**Fig. 1.** Typical structures for (*A*) arrays with each cell having a dedicated access element (transistor) and (*B*) crossbar arrays with equivalent circuit representations used in the following discussion. The specific case $n = 3$ is used for illustration, but practical arrays are much larger (e.g., to reduce peripheral overhead in memory applications).

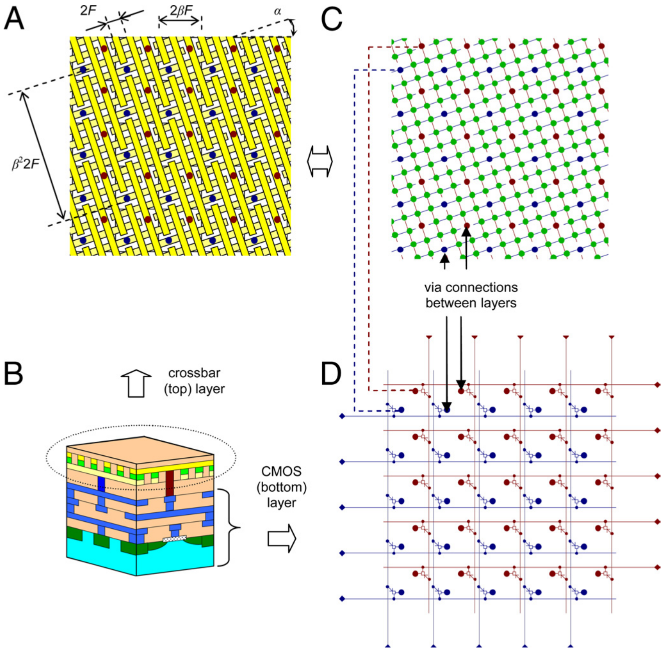

**Fig. 2.** Two-dimensional circuits with an area distributed interface. (*A*) Top view of the crossbar structure showing $\alpha$ for $r = 3$. (*B*) Cut-away illustration showing the two types of vias connecting the CMOS control circuitry to the lower (blue) and upper (red) wire levels of the crossbar. (*C*, *D*) Corresponding equivalent circuit diagram for the $n = 5$ primitive cell array using the notations from Fig. 1.

Thus, the demultiplexing and multiplexing functions are performed with an "edge interface" that utilizes $N$ channels on each of two sides of the array (for a total of $2N$ channels) to access $N^2$ memory cells using a two-label address. By integrating the access function into the crosspoint memory device, one can implement crossbar memory circuits (Fig. 1*B*). In the case considered here, the crosspoint device is a memristive element with a highly nonlinear current–voltage characteristic such that current flow can be detected by applying a full voltage bias across a specified junction while biasing the rest of the lines at half of that voltage to suppress the leakage currents (12).

Crossbar circuits have attracted a great deal of attention, because the device integration density is $(2F)^{-2}$, where $F$ potentially can be scaled down to a few nanometers (12–17), whereas other types of memory devices that incorporate a transistor into each memory element require a larger area per device. In addition, even if $F$ is set by optical photolithography (a few tens of nanometers), the bit density with $M$ crossbar layers is $M(2F)^{-2}$ (18). The greatest challenge is how to approach the maximum density that can be fabricated given the limited functionality of memristive devices and the overhead required for a CMOS addressing circuit and vias to connect the layers. An interesting solution to this problem, called CMOL (11, 19), was developed in the context of nanoscale crossbar circuits (Fig. 2). The key features in this hybrid solution are (*i*) an "area interface" between the CMOS and nanosubsystems, (*ii*) the crossbar array rotated by an angle $\alpha$ with respect to the mesh of CMOS-controlled vias (20), and (*iii*) a double decoding scheme that provides unique access to each crosspoint device (11, 19).

More specifically, as Fig. 2 shows, two types of vias,* one connecting to the lower (shown with blue dots) and the other to the upper (red dots) wire level in the crossbar, are arranged into a square array with side $2\beta F$ (which is also equal to the side length of the "cells" grouping two vias of each kind). Here, $\beta$ is a dimensionless number >1 that depends on the cell size (i.e., complexity) in the CMOS subsystem. The crossbar is rotated by an angle $\alpha = \arcsin(1/\beta)$ relative to the via array such that vias naturally subdivide the wires into fragments of length $\beta^2 2F$. The factor $\beta$ is not arbitrary but is chosen from the spectrum of possible values $\beta = (r^2 + 1)^{1/2}$, where $r$ is an integer so that the precise number of devices on the wire fragment is $r^2 - 1 \approx \beta^2$.

The decoding scheme in CMOL is based on two separate address arrays (one for each level of wire in the crossbar) with access devices similar to those shown in Fig. 1*A* meshed together. Fig. 2*D* shows that there are a total of $4N$ edge channels (illustrated schematically with one edge channel on each of the four sides of the array) to provide access to two different via controllers (one blue and one red) in each of $N^2$ addressing cells in the CMOS plane. In contrast to standard memory arrays, in CMOL each control and data line pair electrically connects the peripheral input/outputs to a via instead of a single memory element. In turn, each via is connected to a wire fragment in the crossbar. The two perpendicular sets of wire fragments provide unique access to any crosspoint device in a fashion similar to Fig. 1*B*, even for large values of $\beta$. The total number of crosspoint devices that can be accessed by the $N \times N$ array of CMOS addressing cells is $\approx N^2\beta^2$, which can provide a significant multiplicative factor when comparing CMOS to crossbar implementations, especially if the lithographic feature size of the crossbar is smaller than that of the CMOS. An alternate way of viewing this is that one can use complex CMOS circuitry built with a significantly larger feature size to address regular crossbars built at a much finer lithographic scale.

From the discussion so far, that the CMOS area interface of the CMOL architecture actually can address a much larger number of crosspoint devices than are present in the single crossbar may be obvious. The major contribution of this article is to show how large the address space actually is and how to use

*Here, we specify typical metal via plugs (i.e., those used for connecting wires in adjacent metallization layers of the interconnect stack in conventional CMOS technology).

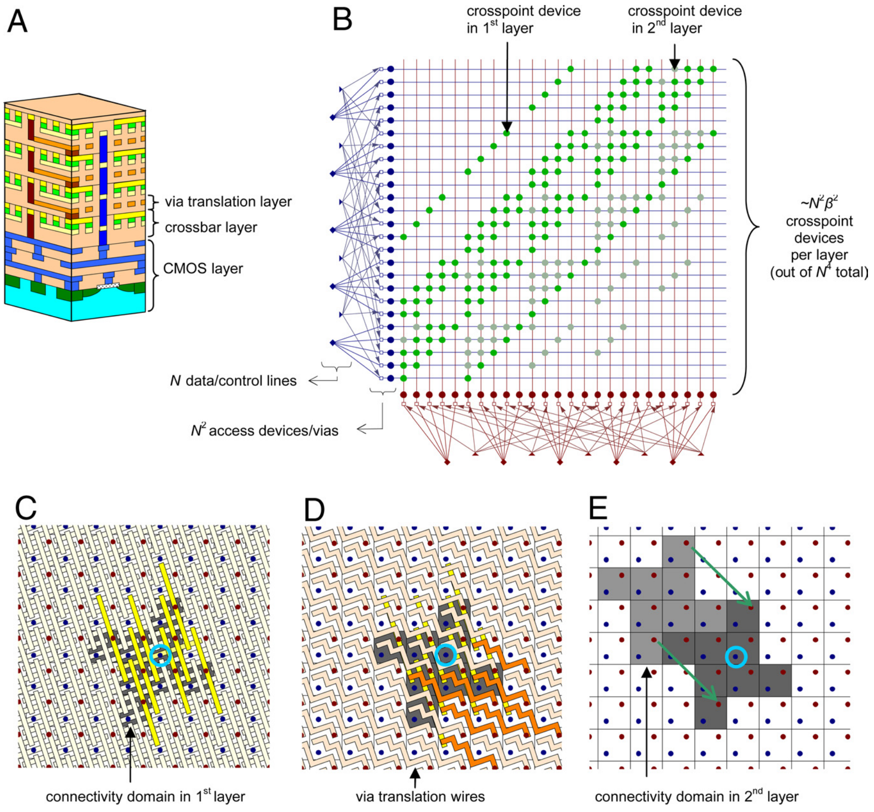
A
B
crosspoint device in 1st layer
crosspoint device in 2nd layer
via translation layer
crossbar layer
CMOS layer
~N²β² crosspoint devices per layer (out of N⁴ total)
N data/control lines
N² access devices/vias
C
D
E
connectivity domain in 1st layer
via translation wires
connectivity domain in 2nd layer

**Fig. 3.** Three-dimensional hybrid CMOS/crossbar circuit with an area distributed interface. (*A*) Cut-away illustration of the circuit showing four crossbar layers ($M = 4$), (*B*) equivalent circuit diagram of the virtual crossbar array for the case $N = 5$, $M = 2$, and $r = 3$, and (*C*–*E*) examples of the connection pattern between two layers. Light blue circles are guides for the eye highlighting the specific via used for the explanation in the text. In *C*, the cells forming a connectivity domain and the corresponding wires are highlighted with gray and yellow colors. Similarly, *D* highlights the wires implementing the translation of red vias within the considered domain, whereas *E* shows the corresponding connectivity domains of a given blue via for the first and second layers.

it. To see the total size of the addressing space, the decoding is shown in Fig. 3*B*, which was constructed by merging Fig. 2 *C* and *D* using the corresponding graph representation of Fig. 1*A* to create a virtual two-dimensional crossbar. From Fig. 3*B*, that the first level of decoding selects from $2N^2$ vias with $4N$ edge channels using a four-label address is clear. The second level of decoding should, in principle, enable the selection of $N^4$ crosspoint devices using the $2N^2$ internal lines (vias) of the area interface, but there are only $N^2\beta^2$ in a single array to be selected. Thus, most of the addressing space available in CMOL is not used, and the virtual array is populated very sparsely. The solution to this problem is to stack multiple crossbar arrays on top of each other using just one set of vias to connect all of the arrays to the cells, as shown in Fig. 3*A*.[†] The theoretical maximum number of layers with the same pitch that can be stacked and still uniquely access all of the crosspoint devices is $M_{\max} = N^2/\beta^2$.

The problem of stacking multiple layers becomes one of geometry to ensure that only one crosspoint device in all of the arrays can be addressed by any allowed set of four address labels (or pair of vias). For example, one algorithm (out of many different possibilities) to place the next crossbar in a sequence is to translate it with respect to the fixed locations of one kind of via (e.g., red vias in Fig. 3) by a distance such that the contacted wire fragments in the new layer are connected to a set of cells that is different from any preceding layer. Such a set of cells is called a "connectivity domain" and is illustrated in Fig. 3*C*. In particular, the number of cells (the gray cells in Fig. 3*C*) that can be reached from the particular via (highlighted with the blue circle) of the given cell with a direct wire–memristive device–wire link is $\approx\beta^2$. To check that the shapes of the connectivity domains for the red and blue vias are the same for every cell and approach square shapes ($\beta \times \beta$ cells) for large $\beta$ is straightforward. Fig. 3 *D* and *E* shows how a crossbar can be translated with respect to red vias by $\approx\beta$ (2 for $r = 3$) cells to the left and down with respect to blue vias (translation indicated with green arrows) using the via-translation wiring layer placed between crossbar arrays. Clearly, the connectivity domains in the first and second layers in Fig. 3*E* do not overlap, and unique access to each memristive device is possible. For instance, the shift of the red vias ensures that they have wire–memristive device–wire connections to the highlighted blue via only in one (the first) crossbar layer, whereas there is no such connection in the second layer.

The total number of crosspoint devices that can be addressed within a single crossbar above the first layer is somewhat smaller than $N^2\beta^2$ because some vias at the edges of the arrays will not be connected to the CMOS substrate. Indeed, Fig. 3 *C*–*E* shows that in a new layer $\approx\beta N$ vias cannot be addressed after the shift operation (i.e., in a $\beta$-wide strip of cells at the top and left side of the array), so the $m$th layer has a fraction ($\approx m\beta/N$) of the crosspoint devices that are not addressable. However, $N$ can be very large and only limited by the size of a chip, so the number of unaddressable crosspoint devices is negligible. This is because

[†] Recently, two interesting suggestions to construct three-dimensional crossbar array circuits were published (21, 22). The first article describes a scheme for interfacing each crossbar wire at the periphery, and therefore, from the authors' point of view, such circuits would have inferior properties (see, for example, the discussion of this point in ref. 11). The main idea of the second article (22) is to alleviate the problems created by the special pin requirement of the original CMOL structure by face-to-face wafer bonding of two dice. The extension of this idea to three-dimensional structures would incur the problems of wafer bonding described above.

the wire fragment size is independent of $N$ and only defined by the parameter $\beta$. For example, assume that all of the crossbar wires are made with optical lithography and the cell consists of two access transistors serving the two vias. In this case, $\beta \approx 5$ (12) and $N \geq 10{,}000$ for a 1-cm$^2$ chip with $F = 100$ nm, so even with an aggressive $M = 100$, the total number of wasted crosspoint devices is $<5\%$.

## Discussion

Four labels (i.e., the row and column addresses for each of the two via types in the CMOS area interface) are required to specify the address of each crosspoint device, thus mapping the three-dimensional device location to a four-dimensional address space. The physical device location is specified by the spatial position of the line fragment in the Euclidean space and its relative position inside the fragment.

In Fig. 3, all of the wire and via patterns are identical in all of the crossbar layers. Consequently, the area density of the vias is kept constant through all of the stacked layers, and adding new layers does not require any changes in the layers below. Thus, this scheme for stacking crossbar layers can be very cost efficient, requiring only a few unique sets of patterning masks or molds if features are defined by optical photolithography or nanoimprint technology, respectively. At the same time, the via density of the area interface can be high enough such that the communication throughput is also very high.

Unlike the original CMOL concept (11), stacking of multiple crossbars would require layer-to-layer alignment in positioning wires and vias. For state-of-the-art photolithography, the overlay accuracy (i.e., $3\sigma$, where $\sigma$ is the standard deviation) is typically at least five times smaller than the minimum feature size $F$ (2).[‡] Moreover, only the relative alignment between the adjacent patterned layers is important, so the overlay alignment error does not accumulate with stacking. For example, Fig. 4 illustrates one plausible way of sustaining the minimum feature size of patterning technology as the number of layers grows by replicating the alignment mark at each step. The features at every patterning step are produced only at that level (i.e., there is no requirement to etch through to lower levels). For many contemporary CMOS circuits, the size of the metal wires typically becomes larger with each succeeding layer, but this is not a fundamental requirement. Rather, it is a convenient feature resulting from the fact that wires in higher layers of an integrated circuit interconnect stack usually are required to conduct larger currents than those in lower layers and the use of a lower-resolution lithography tool for the higher metal layers to lower production costs. For a sufficiently valuable chip, the same quality of surface polish and photolithography can be applied to multiple levels.

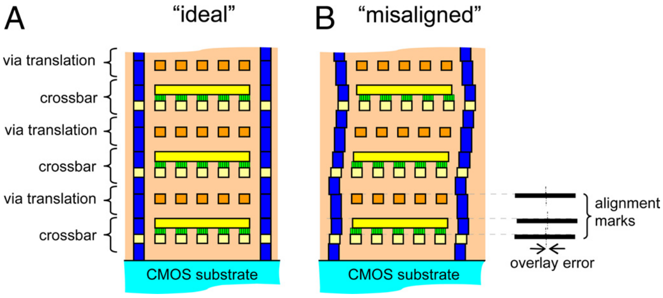

**Fig. 4.** Cross-section of three-dimensional circuit illustrating (*A*) "ideal" alignment between layers and (*B*) more realistic scenario with overlay error.

The combination of high bandwidth and density can be used effectively for various applications besides digital stand-alone memories. For example, memristive devices can act as programmable connections in a hybrid FPGA circuit (19, 24, 25) and as electronic versions of synapses in bio-inspired adaptive networks (20, 26, 27). For these applications, the most important characteristics of the circuit architecture are the programmable complexity (or crosspoint device density) and connectivity of the cells. With the area interface, any CMOS cell can be connected directly to $M\beta^2$ other cells through a single memristive device. Over the long term, we see that this stacking technology offers the possibility of continued scaling of memory density (e.g., for $M = 100$ and $F = 10$ nm the theoretical density is as high as 100 terabits per square centimeter). This would be equivalent to the result of another 15 years of Moore's Law memory scaling for which lateral shrinkage is replaced by stacking.

[‡] The overlay accuracy for nanoimprint technology is not as good as that of the state-of-the-art photolithography yet. For example, the best commercially available tool for nanoimprint, Imprio300 (23), offers patterning of sub-32-nm features with ≈10-nm overlay accuracy across a single wafer. Nevertheless, expectations that overlay accuracy for this relatively immature technology will improve rapidly are realistic.

**ACKNOWLEDGMENTS.** The authors thank Dr. Wei Wu for helpful discussions on the topic of lithography design rules for integrated circuit fabrication.

1. Frank DJ, et al. (2001) Device scaling limits of Si MOSFETs and their application dependencies. *Proc IEEE* 89:259–288.
2. International Technology Roadmap for Semiconductors (2007).
3. Benkart P, et al. (2005) 3D chip stack technology using through-chip interconnects. *IEEE Design Test Computers* 22:512–518.
4. Sakuma K, et al. (2008) 3D chip-stacking technology with through-silicon vias and low-volume lead-free interconnections. *IBM J Res Dev* 52:611–622.
5. Kim E-K, Sung J (2008) Yield challenges in wafer stacking technology. *Microelectron Reliab* 48:1102–1105.
6. Chua LO (1971) Memristor—The missing circuit element. *IEEE Trans Circuit Theory* 18:507–519.
7. Chua LO, Kang SM (1976) Memristive devices and systems. *Proc IEEE* 64:209–223.
8. Strukov DB, Snider GS, Stewart DR, Williams RS (2008) The missing memristor found. *Nature* 453:80–83.
9. Dearnaley G, Stoneham AM, Morgan DV (1970) Electrical phenomena in amorphous oxide films. *Rep Prog Phys* 33:1129–1192.
10. Waser R, Aono M (2007) Nanoionics-based resistive switching memories. *Nat Mater* 6:833–840.
11. Likharev KK, Strukov DB (2005) CMOL: Devices, circuits, and architectures. *Introducing Molecular Electronics*, eds Cuniberti G, Fagas G, Richter K (Springer, Berlin), pp 447–478.
12. Strukov DB, Likharev KK (2007) Defect-tolerant architectures for nanoelectronic crossbar memories. *J Nanosci Nanotechnol* 7:151–167.
13. Chen Y, et al. (2003) Nanoscale molecular-switch crossbar circuits. *Nanotechnology* 14:462–468.
14. DeHon A, Goldstein SC, Kuekes PJ, Linkoln P (2005) Nonphotolithographic memory density prospects. *IEEE Trans Nanotechnol* 4:215–228.
15. Green JE, et al. (2007) A 160-kilobit molecular electronic memory patterned at $10^{11}$ bits per square centimetre. *Nature* 445:414–417.
16. Jo SH, Kim K-H, Lu W (2009) High-density crossbar arrays based on a Si memristive system. *Nano Lett* 9:870–874.
17. Meier M, et al. (2008) Nanoimprint for future non-volatile memory and logic devices. *Microelectron Eng* 85:870–872.
18. Lee M-J, et al. (2008) Stack friendly all-oxide 3D RRAM using GaInZnO peripheral TFT realized over glass substrates. *Proceedings of Electron Devices Meeting* (San Francisco), pp 85–88.
19. Strukov DB, Likharev KK (2005) CMOL FPGA: A reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices. *Nanotechnology* 16:888–900.
20. Likharev K, Mayr A, Muckra I, Türel Ö (2003) CrossNets: High-performance neuromorphic architectures for CMOL circuits. *Ann NY Acad Sci* 1006:146–163.
21. Gojman B, Rubin R, Pilotto C, DeHon A, Tanamoto T (2006) 3D nanowire-based programmable logic. *Proceedings of NanoNet Conference* (IEEE, Lausanne, Switzerland), pp 1–6.
22. Tu D, Liu M, Wang W, Haruehanroengra S (2007) 3D CMOL: A 3D FPGA using CMOS/nanomaterial hybrid digital circuits. *IET Micro Nano Lett* 2:40–45.
23. Lloyd LC, Matt M (2009) SEMATECH's nanoImprint program: A key enabler for nanoimprint introduction. *Proc SPIE Int Soc Opt Eng* 7271:72711Q.
24. Snider GS, Williams RS (2007) Nano/CMOS architectures using a field-programmable nanowire interconnect. *Nanotechnology* 18:035204.
25. Xia Q, et al. (2009) Memristor–CMOS hybrid integrated circuits for reconfigurable logic. *Nano Lett* 9:3640–3645.
26. Mead C (1989) *Analog VLSI and Neural Systems* (Addison–Wesley, Reading, MA).
27. Snider GS (2007) Self-organized computation with unreliable, memristive nanodevices. *Nanotechnology* 18:365202.