Performance Simulation and Analysis of a CMOS/Nano Hybrid Nanoprocessor System

Adam C Cabe and Shamik Das

Nanosystems Group, The MITRE Corporation, McLean, VA 22102 USA

E-mail: sdas@mitre.org

Abstract.

This paper provides detailed simulation results and analysis of the prospective performance of hybrid CMOS/nano electronic processor systems based upon the Field-Programmable Nanowire Interconnect (FPNI) architecture. To evaluate this architecture, a complete design was developed for an FPNI implementation using 90-nm CMOS with 15-nm-wide nanowire interconnects. Detailed simulations of this design illustrate that critical design choices and tradeoffs exist beyond those specified by the architecture. This includes the selection of the types of junction nanodevices, as well as the implementation of low-level circuits. In particular, the simulation results presented here show that only nanodevices with an "on/off" current ratio of 200 or more are suitable to produce correct system-level behavior. Furthermore, the design of the CMOS logic gates in the FPNI system must be customized to accommodate the resistances of both "on"-state and "off"-state nanodevices. Using these customized designs together with models of suitable nanodevices, additional simulations demonstrate that, relative to conventional 90-nm CMOS FPGA systems, performance gains can be obtained of up to 70% greater speed or up to a nine-fold reduction in energy consumption.

This paper appears in *Nanotechnology* , vol. 20, no. 16, 22 Apr. 2009.

1. Introduction

Hybrid micro-nano electronics systems [1–9] seek to combine the very best of industrial microelectronics–complementary metal oxide semiconductor (CMOS) technology–with nanoelectronics, whose chief advantage over CMOS is its capacity for ultra-dense integration of devices and interconnects. In so doing, such hybrid systems purport to offer performance that exceeds that of either CMOS or nanoelectronics alone. Specifically, hybrid systems promise greater computational speed, plus lower power and energy consumption, all within a smaller system form factor due to the increased density of integration.

Two system proposals, in particular, have garnered much recent attention [10]. These systems are CMOL ("CMOS+nanowires+MOLecules"), developed by Likharev *et al.* [5], and its close relative FPNI ("Field-Programmable Nanowire Interconnect"), devised by the Hewlett-Packard corporation [8]. The CMOL and FPNI architectures combine CMOS logic elements with nanowire crossbar arrays to form programmable interconnect fabrics akin to Field-Programmable Gate Arrays (FPGAs) [11, 12].

Designs for both CMOL and FPNI systems have been specified very thoroughly at the architectural level by their respective designers. Initial, high-level analyses conducted by these designers indicate that both systems offer significant promise when measured according to metrics such as circuit speed and system area. Furthermore, the design of the FPNI architecture contains specific enhancements [8] to CMOL that are intended to make the manufacturing of such systems feasible using established nanofabrication technologies, such as nanoimprint lithography [13–16]. A clear next step would be laboratory experimentation to fabricate and test physical prototypes of these systems.

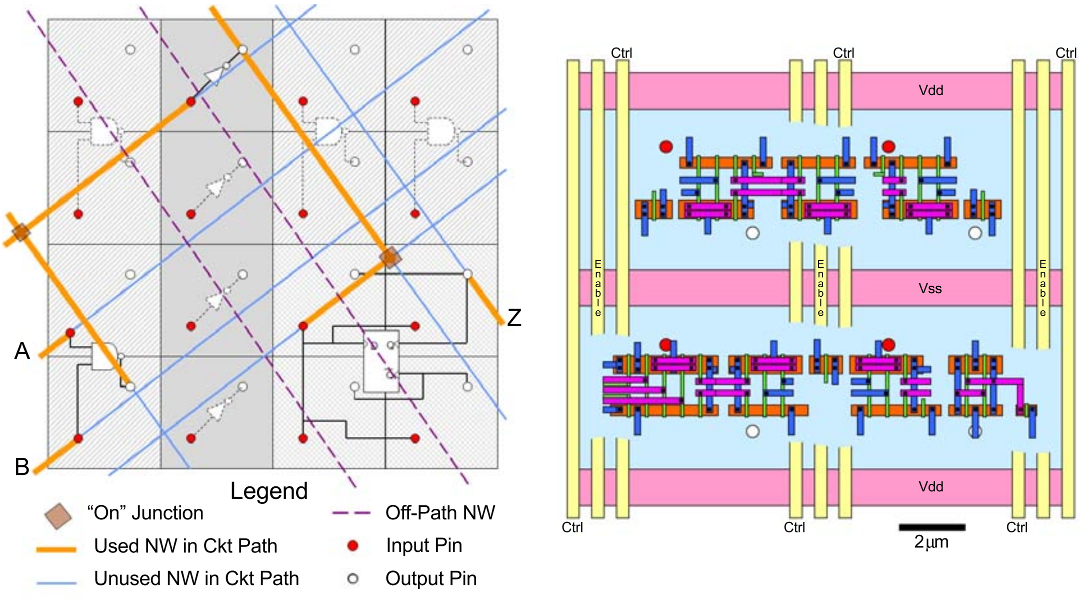
In support of that objective, this paper presents detailed simulation results that demonstrate the critical design challenges and tradeoffs for FPNI that exist beyond the architectural specification of Snider and Williams [8]. In particular, it is shown here that any nanoelectronic switches to be used in FPNI systems must provide an "on/off" current ratio of 200 or more. This restricts options for near-term experiments to a small set of demonstrated nanodevices. Further simulations illustrate that by using such nanodevices, in conjunction with CMOS circuits that are customized to interface with them, performance gains may be achieved of up to 70% greater circuit speed or up to a nine-fold reduction in energy consumption, relative to conventional CMOS FPGAs.

To begin to explain the approach that led to these findings, a detailed design is presented in section 2 for an FPNI system that implements a simple logic circuit. This design is based upon 90-nm CMOS technology, combined with an FPNI-style nanowire crossbar [8] composed of 15-nm-wide nanowires. These dimensions were chosen since they can be achieved with technology that presently is accessible to the research community. Following the discussion of this design, simulation results are presented in section 3. These results elucidate the design choices that enable FPNI systems to function correctly, as well as those that permit functioning FPNI systems to be optimized. Section 4 provides a summary and conclusions.

2. Detailed Design for an FPNI System

Hybrid systems such as CMOL and FPNI consist of two interdependent components: an array of CMOS cells and a homogeneous, switchable array of crossed nanowires that resides atop these cells. The design of such systems is constrained by relative size scales of these components. For example, a complete CMOL design is determined almost entirely by the size ratio of the logic cell to the unit nanowire crossbar. This is because CMOL utilizes just one type of logic cell, the inverter. In contrast, the FPNI architecture permits multiple types of CMOS logic, such as NAND gates and flip-flops. These logic gates may vary widely in size. In order to pack these disparate gates into a homogeneous fabric, the FPNI fabric is partitioned into uniform, rectangular "hypercells" [8]. Each hypercell consists of a small number of unit cells, where a single unit cell corresponds to the smallest logic element, typically a buffer or inverter.

The design of the hypercell must be customized prior to fabrication in order to optimize the FPNI system for its intended applications. For example, Snider and Williams present two hypercell variations in their original work [8]: one variation consists of four two-cell NAND gates and eight single-cell buffer



(a) (b)

Figure 1. Schematic and layout for a 4 × 4 FPNI hypercell. This hypercell can be tiled to make larger FPNI fabrics. Figure (a) shows that this hypercell consists of 4 NAND gates, 4 inverters, and a flip-flop. Figure (b) provides a cutaway view of a partial CMOS layout for the FPNI flip-flop hypercell.

gates in a 4 × 4 hypercell, and the other variation is a 6 × 7 hypercell that provides a flip-flop element in addition to other, simpler logic gates.

Figure 1(a) shows an alternative 4 × 4 hypercell design developed for this analysis. It provides four two-input NAND gates, four inverters, and a flip-flop. In comparison to the hypercells described by Snider and Williams [8], this design provides a greater density of flip-flops per unit cell, as is desirable for the pipelined arithmetic operations considered in section 3.

Also, the FPNI design developed and considered here goes a step beyond the work of Snider and Williams in that it is specified all the way down to the CMOS layout and takes into account precise dimensions for all the CMOS components. A sample CMOS layout for this design is shown in figure 1(b). As is shown in this layout, there can be a small area penalty to be paid in the form of empty space in some unit cells. This is due to the fact that the areas of the complex logic gates are not integer multiples of that of the inverter or buffer, as would be desirable to take full advantage of the density of the interconnect layer above. Thus, in comparison to a custom CMOS-only design, where there is no need to align to a uniform nanowire interconnect structure, the mapping of FPNI gates to integer unit cells can be inefficient. However, it should be noted that in designing the layout shown in figure 1(b), circuit function was given priority over mapping efficiency.

As with the design of the underlying CMOS logic gates, design choices also arise in the FPNI interconnect layer. This interconnect consists of the nanowire crossbar array and the programmable nanodevices that exist at each nanowire junction. The nanowire crossbar consists of two layers of parallel nanowires, one laid orthogonally over the other, creating a 2-D interconnect grid [2]. Nanowire crossbar arrays of the required scale have been demonstrated with pitches as low as 14 nm [13,14]. The nanowire pitch, together with the CMOS unit cell dimensions, determines the location and number of programmable connections between adjacent logic gates. Thus, the first design choice for the FPNI interconnect layer is to decide the nanowire pitch. For this design, a nanowire pitch of 30 nm was selected because it is aggressive, yet accessible with present technology.

The second choice to be made is to decide upon an appropriate junction nanodevice. The configurable nanodevice at each junction must have both a high-conductivity ("on") state and a low-conductivity ("off") state, making it bi-stable. Applying a large positive or negative voltage across the device causes it to switch states. Although many device technologies provide this functionality, in order to select an appropriate technology, one must understand first how these devices are intended to work within FPNI circuits.

The schematic in figure 1(a) shows how these nanodevices are employed to create functional circuits in the FPNI fabric. In this figure, a two-input NAND gate, with inputs 'A' and 'B', is connected to an inverter. The output of this inverter is connected to the flip-flop, whose output is marked 'Z'. The figure shows the nanowires and junctions that are employed to create this circuit, as well as some nanowires in the vicinity that are unused, both in the circuit path and off the path. For clarity, some nanowires are omitted from the figure.

This figure illustrates a design challenge that must be resolved through simulation. Specifically, the circuit path is dictated by programming the required junction nanodevices into their "on" conductive state. Thus, ideally, it is desired that the "off" state of these devices conduct no current, i.e., that the "on/off" ratio be infinite. In practice, such ratios occupy a wide variety of non-ideal, finite values that depend on the device composition.

For example, self-assembled monolayers of molecules, such as rotaxanes and pseudo-rotaxanes [17], yield "on/off" current ratios from two to 11 [18, 19]. Other molecular devices, such as an oligo(phenylene-ethynylene) (OPE) molecule with a nitro sidegroup [20], produce similar ratios of approximately 10. Inorganic nanodevices, such as those based upon metal oxides, also have been shown to exhibit useful switching characteristics. Examples include Cu_2O, Al_2O_3, NiO, and TiO_2. For such devices, "on/off" ratios of 100 or more have been demonstrated [21], with those of Cu_2O [22] and TiO_2 [23] as high as 1000. In particular, the latter material is central to the "memristive" nanodevice [24] proposed by the Hewlett-Packard team that invented the FPNI architecture. In addition to metal oxides, other nanowire-based inorganic junction nanodevices also have been demonstrated to achieve device "on/off" ratios on the order of 1000 or more [25, 26].

Since practical "off"-state nanodevices necessarily will conduct some current, the various CMOS cells and hypercells will not be isolated completely. Thus, the design of circuits in FPNI fabrics must account not only for the CMOS logic and nanodevices to be used in the circuits, but also those that are unused, yet adjacent to the circuits. A simplified example is shown in figure 2. This figure depicts two CMOS logic gates, shown at the upper left and lower right, connected through a nanowire crossbar array. Other gates are shown to share this nanowire array (through connections that are not shown in the figure). Here, a logic '1' is intended as the voltage signal transmitted via the topmost horizontal nanowire. However, the presence of the '0' signals pulls down the output, denoted '?', through the resistive bridge that is formed from the "on"-state junction and the parallel collection of "off"-state junctions, which have finite resistance.

In this simplified example, there is one vertical nanowire to consider with, say, N "off"-state resistors. If the maximum tolerable error in the '1' voltage is ϵ (as a fraction of the total voltage), then

$$\frac{R_{ON}}{R_{ON} + R_{OFF}/N} < \epsilon,$$

i.e., the "on/off" current ratio, equal to $\frac{R_{OFF}}{R_{ON}}$, must exceed $N(\frac{1}{\epsilon} - 1)$. For the complete design presented in this paper, $N = 7$, and assuming $\epsilon = 0.1$, this provides a theoretical requirement that the "on/off" ratio exceed 63.

However, this simplified analysis does not consider the impact of the other nanowire junctions implied in the figure, nor does it consider the other possible configurations of the additional logic gates. Also, importantly, it models the nanowire junction nanodevices as linear resistors and omits the nonlinearities present in experimentally demonstrated nanodevices. Furthermore, this analysis neglects the impacts of parasitic components, such as the nanowire resistances and the capacitances that couple the nanowires. In conjunction with the junction and nanowire resistances, these capacitances can affect the propagation of signals through the crossbar array as these signals change in value.

The most effective way to evaluate these nanodevice and interconnect issues accurately and exhaustively is via the use of detailed system simulation, which takes into account the behavior of individual devices and parasitic components, as well as their behavior in aggregate. Such simulations are discussed in the next section.

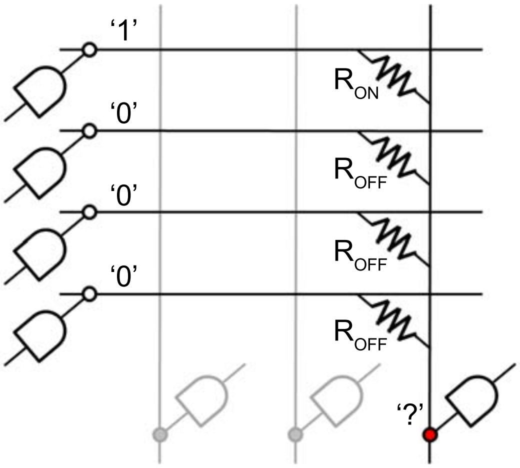


Figure 2. Simplified schematic of the nanowire crossbar interconnecting a set of CMOS logic gates. For the configuration depicted here, the upper-left logic gate is modeled as providing the input to the lower-right gate through a linear resistor. As seen here, the other gates may produce conflicting signals that corrupt the output nanowire, denoted '?', via the finite-resistance "off"-state junctions that connect them.

3. Simulations of the FPNI System

3.1. Simulations of System Functionality

The inventors of CMOL and of FPNI evaluated their respective systems by mapping the Toronto 20 benchmark circuits [27] into their fabrics and examining the overall performance [5, 8]. In doing this, they focused on three primary metrics: circuit area, critical path delay, and dynamic power consumption (i.e., the portion of the total power that primarily is capacitive and is consumed during transitions in the digital state of a circuit). The circuit area was calculated directly from the mapping. The other two metrics were estimated using high-level analytical techniques, such as Elmore delay modeling [8].

However, the nonidealities of the nanodevice behaviors are likely to result in unexpected system-level performance issues in CMOL- and FPNI-based systems. Such nonidealities are not amenable to simple, high-level analytical modeling. Instead, detailed computer-based simulation is required to evaluate the impacts of these behaviors fully. This is well known to designers of deep-submicron and nanometer CMOS, where accounting for the nonidealities of interconnect behavior is a key factor in the characterization and optimization of system performance. No CMOS system design can be completed without simulation at the layout level of the system or its subsystems. This is certain to be true to an even greater extent for nanoelectronic and hybrid CMOS/nano designs, in which ultra-miniaturization exacerbates the parasitic behaviors of interconnects relative to those of the underlying devices.

To study the impacts of such parasitics in a nanoprocessing system, a full adder circuit was designed and mapped into a simulated FPNI fabric. This adder circuit takes three single-bit inputs and produces a two-bit output that is the binary sum of the inputs. This circuit is ubiquitous in digital logic, and therefore, its performance is indicative of that of larger systems, including nanoprocessors. Thus, detailed simulation of this circuit is conducted in lieu of the detailed simulation of an entire nanoprocessor, which would be computationally intractable, just as would be the detailed simulation of an entire commercial microprocessor.

The full adder design is intended especially to provide insight into how the FPNI architecture might scale to larger circuit sizes. The design uses 11 NAND gates and two flip-flops. It requires four of the 4×4 hypercells described in section 2. (These hypercells were designed with the full-adder circuit in mind; the same circuit would require two 6×7 hypercells from Snider and Williams [8], who did not optimize their design for this application.)

Detailed simulations of this hybrid full-adder circuit were performed using a methodology developed originally for simulating nanomemory and nanoprocessor systems. This methodology, together with

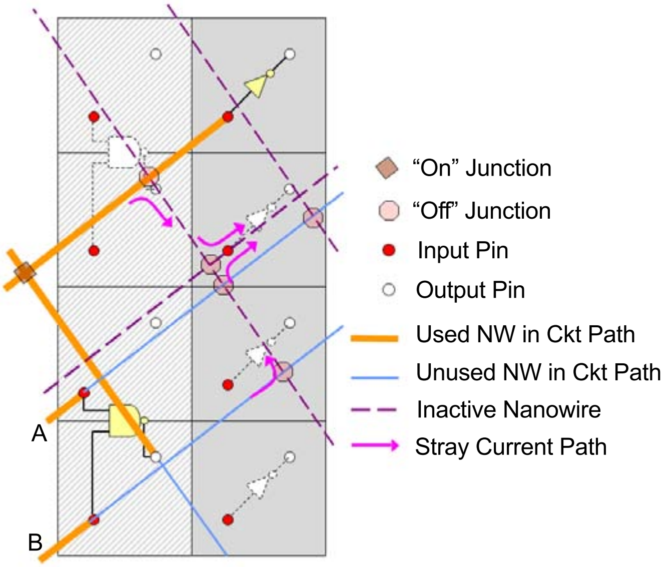


Figure 3. A portion of an FPNI circuit highlighting the current leakage paths through the system. Given inputs A and B, the arrows denote the stray currents flowing through "off" devices that amass at one particular unused inverter (second from top).

the CAD environment and nanodevice models, is discussed in detail in prior publications [28–30]. Four main steps are involved. First, empirical data are obtained for the desired nanodevices and interconnect structures. Second, these data are encapsulated in models written in the Verilog-A language [28–30]. Third, a system-level schematic is assembled within the Cadence Virtuoso modeling software [31], using models for each CMOS device and each nanodevice. Finally, the electrical behavior of the circuit is simulated using the Cadence Spectre simulator [31].

The empirical data used for modeling the nanodevices were obtained from published experimental results on rotaxane-based nanodevices [18, 19, 32]. These nanodevices exhibit exponential current-voltage (I-V) behaviors that are characteristic of many of the resistive nanodevices demonstrated to date [20–22, 25, 26]. Using this experimental data, parameterized Verilog-A models were developed, through which characteristics such as nanodevice resistance could be varied by adjusting the parameters. For example, in initial simulations, the nanodevice "on" resistance was assumed to be 2.5 MΩ, which is consistent with experimental data [17–22, 25, 26, 32].

Models for the nanowire interconnects were based upon resistor-capacitor networks. These interconnect models were constructed using the method of Steinhögl *et al.* [33] that also was employed by Snider and Williams [8]. As stated in section 2, the nanowires were assumed to be 15 nm wide with a pitch of 30 nm. The wire resistivity was set at 8.88 $\mu\Omega$·cm and the substrate and coupling capacitances were 2 pF/cm and 1 pF/cm, respectively.

System schematics were assembled as follows. First, a detailed CMOS layout was designed for the FPNI hypercell shown in figure 1(a). This layout was used to determine the physical dimensions of the nanowire interconnect network. Given the physical dimensions, the schematics were completed by combining the aforementioned nanodevice and nanowire models with Cadence Spectre models of 90-nm CMOS transistor devices.

Using these models and schematics, analyses were carried out in the Cadence Spectre simulator to establish the functionality of FPNI circuits. This was done by simulating the behavior of individual logic gates within the aforementioned schematics. These simulations revealed that there can exist undesired "sneak leakage" current paths flowing throughout the nanowire interconnect array. This issue is depicted in figure 3. Since the nanowire interconnect is based on resistive nanodevice connections that have finite "on/off" ratios, current flows through both the desired "on"-state nanodevices and the unselected "off"-state nanodevices. In figure 3, the bold wires highlight the intended circuit path using diamonds to denote the "on"-state nanodevices and large circles to indicate the "off"-state nanodevices.

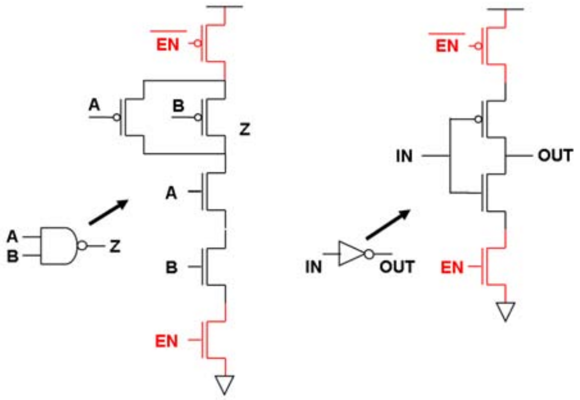


Figure 4. Transistor-level designs for the NAND and inverter gates used in the FPNI schematic. These designs are modified from standard CMOS and FPNI implementations through the addition of the uppermost and lowermost transistors. The "EN" signals that drive these transistors permit the disconnection of unused CMOS logic cells from the power supply, thus reducing the leakage power consumption of these cells.

The arrows represent one example of stray currents in the nanowires.

Detailed circuit simulations demonstrate that these currents can be large enough to disturb the voltage states of internal nodes of the CMOS logic network. This can partially turn on CMOS transistors that are intended to be unused (and therefore off). As a result, there can be short-circuit current paths within the CMOS circuitry itself. Such CMOS leakage current can result in significant power consumption.

Thus, before carrying out further simulations, the CMOS logic cells were redesigned to prevent the CMOS sneak leakage paths. Each CMOS logic gate in the revised design is implemented with "sleep" transistors that allow for power to be disconnected from the unused circuits. Example modified CMOS circuits are shown in figure 4. In these examples, the sleep transistors are inserted next to each power supply line.

Using these design refinements, further simulations were conducted to assess the functionality of FPNI systems. In particular, the "on/off" ratio was expected to have significant impact on the currents that flow through both the intended and undesired interconnect paths. To verify this expectation, simulations were performed by varying the "on/off" ratio, keeping the "on" resistance fixed at 2.5 MΩ.

Figure 5 shows the results of this simulation. The waveforms depicted in this figure confirm the existence of a minimum threshold "on/off" ratio in order to guarantee correct logic operation. At low "on/off" ratios, such as in curve (a), almost no correct output values are attained. However, as the "on/off" ratio is increased, the adder begins to work as intended. The simulation shows that the circuit functions at a ratio of 200, albeit with some voltage waveform degradation still visible during the transitions between '0' and '1' states. Further contrast between "on" and "off" resistances yields the correct, full '0'-to-'1' output. Because it takes many more details into account, this simulation improves upon and gives a somewhat higher, more accurate estimate of the "on/off" ratio requirement than does the illustrative, algebraic analysis that was conducted above in section 2.

These simulations show that once fabricated, the FPNI architecture can be made to work using experimentally demonstrated nanodevices. However, the simulations illustrate that only nanodevices with an "on/off" ratio of 200 or more are suitable for this architecture. Of the devices that are suitable, there exist a variety of options for the "on" resistance, "off" resistance, and "on/off" ratio. Thus, it is important to examine how these parameters may be tuned to optimize the performance of a functioning FPNI design. Simulations to address such questions are described in the next subsection.

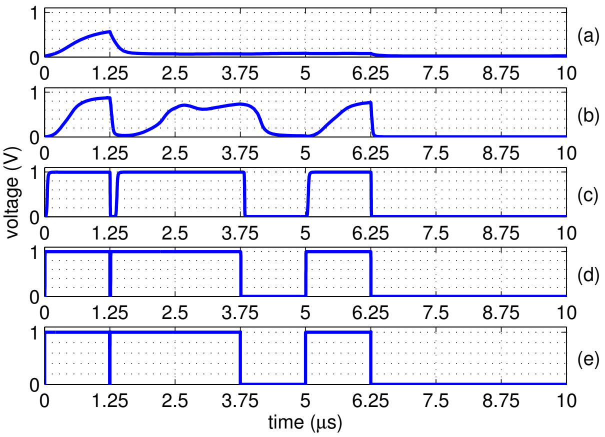


Figure 5. Simulations of the FPNI full-adder circuit. The waveforms shown here depict the voltage of the carry output bit for various values of the junction nanodevice "on/off" ratio. This ratio was set via simulation to (a) 2, (b) 20, (c) 200, (d) 2,000, and (e) 20,000, respectively. The simulated waveforms show that correct behavior is not obtained for the two lowest values of the "on/off" ratio, and also that the waveform voltages achieve full '0'-to-'1' swing only for the two highest values.

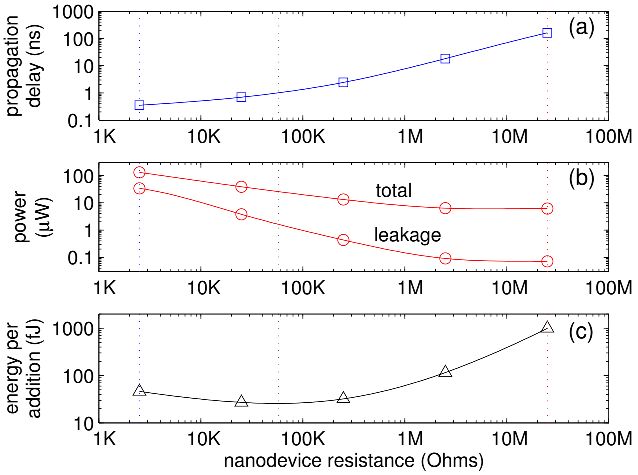


Figure 6. Three-part plot showing the impact of nanodevice "on"-state resistance on circuit delay and energy consumption for the FPNI full adder circuit. This plot provides (a) circuit delay, (b) average power, and (c) energy per addition operation, all as a function of the nanodevice "on" resistance. The optimum resistances for circuit delay, power, and energy per addition are denoted by vertical dotted lines at 2.5 KΩ, 25 MΩ, and 57 KΩ, respectively. The "on/off" ratio is fixed at 2000 in all cases.

3.2. Simulations to Optimize System Performance

Simulations were carried out to determine the impact of nanodevice resistances on circuit performance for a functioning FPNI adder circuit. In these simulations, the circuit delay, power, and energy consumption were evaluated for various nanodevice "on" resistances. The "off" resistances also were varied so that the "on/off" ratio was fixed at 2000 in all cases (which ensures correct functionality). This ratio is a reasonable basis for further simulations since, as discussed above in section 2, several appropriate devices have been demonstrated with "on/off" ratios exceeding 1000.

The results of these simulations are shown in figure 6. Figure 6(a) details the impact on signal propagation delay. This simulation shows a monotonic increase in delay with the nanodevice "on" resistance, assuming a fixed "on/off" ratio. At 24 KΩ, the minimum nanodevice "on" resistance

	CMOS Full Adder	Xilinx Spartan-3	FPNI Full Adder	
			Fastest	Least Energy
Circuit Delay (ps)	148	610	354	698
Leakage Power	10.17 nW	36 nW	33.89 μW	3.79 μW
Dynamic Energy (nW/MHz)	4.30	240	34.24	24.45
Energy per Addition (fJ)	4.30	240	46.25	27.09

Table 1. Comparison of an FPNI full-adder circuit with conventional CMOS implementations. The FPNI full-adder data are the best-case data exhibited in figure 6. This FPNI full adder is compared with a custom, optimized CMOS adder and a Xilinx Spartan-3 [34–36] single-logic-slice FPGA implementation. For each circuit, the circuit delay is given, together with the average leakage power, average dynamic energy, and average total energy per addition.

proposed by Snider and Williams [8], the propagation delay is approximately 0.7 ns, supporting a clock speed of up to 1.4 GHz. In contrast, figure 6(b) shows that power consumption for this circuit decreases as a function of "on" resistance. Here the minimum point is observed at 25 MΩ "on" resistance.

It is clear from figure 6 that there is an optimization tradeoff between delay and power. Furthermore, the product of these two metrics is a single metric that measures the energy per addition operation. This common metric strikes a useful balance between the optimization of speed and power consumption. Figure 6(c) provides this data. Here, it is seen that at low nanodevice resistances, total power dominates due to leakage, while delay remains relatively flat. Conversely, at high nanodevice resistances, delay dominates strongly, while power flattens out. As a result, in the power-delay product, which is the energy per addition, a minimum exists at approximately 57 KΩ. This presents system designers with a middle-ground option between optimizing for circuit speed and optimizing for power efficiency.

To place the results shown in figure 6 in context, it is valuable to compare these results against the performance of a conventional reference circuit. The ideal reference circuit would be one designed for a reconfigurable CMOS technology such as one of several commercially available FPGAs based upon 90-nm CMOS [34, 35]. Alternatively, a custom CMOS full-adder circuit could be used as a reference. Such a circuit would be tailored specifically to compute additions and would not be reconfigurable. This circuit would provide an upper bound for 90-nm full adder performance in terms of speed and energy efficiency.

Table 1 provides a comparison of two FPNI full adder versions to these reference circuits. The two FPNI versions, denoted "fastest" and "least energy" in the table, are the designs using nanodevice "on" resistances of 2.5 KΩ and 57 KΩ, respectively, as determined via the simulation data shown in figure 6. As expected, the fully customized CMOS implementation outperforms FPNI both in delay and in energy consumption. However, when compared against a state-of-the-art reconfigurable CMOS FPGA, the FPNI versions perform better in simulation. As table 1 shows, the FPNI full adder can be made up to 70% faster than the FPGA and simultaneously five times as energy efficient. Alternatively, the FPNI version can be made nine times as energy efficient with only a slight cost to speed. Overall, the FPNI full adders perform more closely overall to the custom CMOS version than to the reconfigurable one.

In particular, it is seen that the dynamic energy consumption of the programmable hybrid circuit is much closer to the custom CMOS than to the programmable CMOS FPGA. This is due to the reduced interconnect capacitance provided by the nanowire interconnect. In contrast, the leakage power is much higher in the FPNI circuit than in either CMOS reference circuit. As discussed above in section 3.1, this is due to the CMOS gate voltage offsets generated by the highly resistive nanodevice network. Nevertheless, due to the low dynamic energy consumption of the FPNI circuit, the overall energy consumption for this circuit is seen to be lower than that of its closest conventional kin, the CMOS FPGA.

Thus, the detailed system simulation results provided here show that by using existing, experimentally demonstrated nanoelectronic devices [23,25,26], a range of system performance options superior to conventional CMOS is available to designers of prospective nano-enabled reconfigurable logic systems such as FPNI. This is the case even though the present state of junction nanodevice research has produced devices that have relatively high resistance values or that are otherwise less suitable than their conventional counterparts. As a result, near-term opportunities exist to improve performance over conventional CMOS by pursuing the fabrication and demonstration of entire systems that hybridize CMOS with presently available nanodevices.

4. Summary and Conclusions

In their paper introducing the FPNI architecture [8], Snider and Williams showed that FPNI performance could by optimized by exploiting design flexibility at the architectural level. For example, by changing the number and/or type of gates within a hypercell, as well as the number of inputs to these gates, various area, delay, and power characteristics could be obtained for a number of different benchmark circuits.

This paper goes beyond that work to show that nanodevice and circuit customizations play an even more fundamental role in the design and optimization of functioning FPNI systems. Such customizations determine whether or not the system will function correctly. Also, even in a correctly functioning system, the CMOS subsystem must be designed to compensate for the non-ideal behavior of the nanodevices. Integration of the design of the nanodevices with that of the CMOS circuits, where each is customized to function with the other, is an essential predicate to further optimization at the architectural level.

The impact of nanodevice "on/off" ratio on overall circuit functionality is apparent from the simulation results presented here. Nanodevices with "on/off" ratios of only two to five yield nanocircuits that produce essentially no correct outputs. In contrast, nanodevices with ratios as low as 200 lead to correct logic values, although the voltage waveforms still are degraded somewhat. With even further contrast between "on" and "off" states, full-swing outputs can be obtained with little to no signal degradation. To achieve these results in practice, near-term physical prototype demonstrations probably will need to employ resistive nanodevices such as those fabricated using metal oxides [23, 25, 26].

Furthermore, even with the highest "on/off" ratios demonstrated to date, nanowire junction devices will conduct a non-trivial amount of current in their "off" states. Simulation results show that FPNI systems will require CMOS subsystems that are designed to drive (and be driven by) this resistive nanodevice network. The modified designs presented here accomplish this by gating the power connections of unused logic cells via "sleep" transistors. Other isolation approaches also may prove effective.

Thus, as shown in this paper, the design of FPNI systems requires careful consideration of hierarchical details, including the choice of nanodevices and the design of basic circuits incorporating these devices. Nevertheless, despite the issues identified here, it is clear from simulations that FPNI systems are capable of achieving the substantial gains in performance predicted by their designers. Simulations show the FPNI architecture to be up to 70% faster or as much as nine times as energy efficient, compared with a commercial state-of-the-art FPGA using the same underlying CMOS technology.

Results such as these make a strong case for the pursuit of hybrid CMOS/nano electronics technologies, and especially, for the prototyping and testing of hybrid systems. Also, they illuminate the critical role performed by detailed system simulation in the design and development of such systems. Thus, it is through close collaboration between design, simulation, and fabrication efforts that the benefits of hybrid CMOS/nano technologies can be made available for electronic system applications.

Acknowledgments

The authors thank R. S. Williams, P. Kuekes, D. Stewart, D. Strukov, G. Snider, and W. Robinett of the Hewlett-Packard Corporation, plus Prof. K. K. Likharev of Stony Brook University and Prof. M. R. Stan of the University of Virginia, for their many generous discussions. Thanks also are due to J. C. Ellenbogen and C. A. Picconatto of the MITRE Nanosystems Group for their helpful comments on this paper. This research was supported by the U.S. Government Nano-Enabled Technology Initiative (NETI).

References

[1] S. C. Goldstein and M. Budiu. Nanofabrics: Spatial computing using molecular electronics. In *Proc. Intl. Symp. Comp. Arch.*, June 2001.

[2] M. M. Ziegler and M. R. Stan. A case for CMOS/nano co-design. In *Proc. IEEE/ACM Intl. Conf. Comp. Aid. Des.*, pages 348–352, 2002.

[3] M. M. Ziegler and M. R. Stan. CMOS/nano co-design for crossbar-based molecular electronic systems. *IEEE Trans. Nano.*, 2(4), December 2003.

[4] R. Weerasekera, J. Liu, L. Zheng, and H. Tenhunen. A nanocore/CMOS hybrid system-on-package (SoP) architecture for future nanoelectronic systems. In *Proc. on High Dens. Microsys. Design and Pack. and Comp. Failure Analysis*, pages 1–4, June 2005.

[5] D. B. Strukov and K. K. Likharev. CMOL FPGA: A reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices. *Nanotechnology*, 16:888–900, 2005.

[6] D. B. Strukov and K. K. Likharev. A reconfigurable architecture for hybrid CMOS/nanodevice circuits. In *Proc. ACM/SIGDA FPGA*, Monterey, CA, 2006. ACM Press.

[7] R. M. P. Rad and M. Tehranipoor. A new hybrid FPGA with nanoscale clusters and CMOS routing. In *Proc. DAC*, pages 727–730, 2006.

[8] G. S. Snider and R. S. Williams. Nano/CMOS architectures using a field-programmable nanowire interconnect. *Nanotechnology*, 18(035204), 2007.

[9] K. K. Likharev. Hybrid CMOS/nanoelectronic circuits: Opportunities and challenges. *J. Nanoelectron. Optoelectron.*, 3(3):203–230, 2008.

[10] See, e.g., (1) J. Markoff, "H.P. to Report an Advance in Adaptable Circuitry," *The New York Times*, 16 Jan. 2007; (2) L. Wirbel, "Silicon, nanoelectronics combo explored," *EE Times*, 11 June 2007.

[11] V. Betz, J. Rose, and A. Marquardt. *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic, New York, 1999.

[12] B. Zeidman. *Designing with FPGAs and CPLDs*. CMP Books, Lawrence, KS, 2002.

[13] N. A. Melosh, A. Boukai, F. Diana, B. Gerardot, A. Badolato, P. M. Petroff, and J. R. Heath. Ultrahigh-density nanowire lattices and circuits. *Science*, 300(5616):112–115, 2003.

[14] M. D. Austin, H. Ge, W. Wu, M. Li, Z. Yu, D. Wasserman, S. A. Lyon, and S. Y. Chou. Fabrication of 5 nm linewidth and 14 nm pitch features by nanoimprint lithography. *Appl. Phys. Lett.*, 84(26):5299–5301, June 2004.

[15] G. Y. Jung, S. Ganapathiappan, D. A. A. Ohlberg, D. L. Olynick, Y. Chen, W. M. Tong, and R. S. Williams. Fabrication of a 34 × 34 crossbar structure at 50 nm half-pitch by UV-based nanoimprint lithography. *Nano Lett.*, 4(7):1225–1229, 2004.

[16] G. Y. Jung, E. Johnston-Halperin, W. Wu, Z. Yu, S.-Y. Wang, W. M. Tong, Z. Li, J. E. Green, B. A. Sheriff, A. Boukai, Y. Bunimovich, J. R. Heath, and R. S. Williams. Circuit fabrication at 17 nm half-pitch by nanoimprint lithography. *Nano Lett.*, 6(3):351–354, 2006.

[17] C. P. Collier, E. W. Wong, M. Belohradsky, F. M. Raymo, J. F. Stoddart, P. J. Kuekes, R. S. Williams, and J. R. Heath. Electronically configurable molecular-based logic gates. *Science*, 285:391–394, 1999.

[18] A. R. Pease, J. O. Jeppesen, J. F. Stoddart, Y. Luo, C. P. Collier, and J. R. Heath. Switching devices based on interlocked molecules. *Acc. Chem. Res.*, 34(6):433–444, 2001.

[19] Y. Luo, C. P. Collier, J. O. Jeppesen, K. A. Nielson, E. Delonno, G. Ho, J. Perkins, H. Tseng, T. Yamamoto, J. F. Stoddart, and J. R. Heath. Two-dimensional molecular electronics circuits. *ChemPhysChem*, 3:519–525, 2002.

[20] N. Majumdar, N. Gergel, D. Routenberg, J. C. Bean, L. R. Harriott, B. Li, and J. M. Tour. Nanowell device for the electrical characterization of metal-molecule-metal junctions. *Vac. Sci. Tech. B*, 23(4):1417–1421, 2005.

[21] I. G. Baek, M. S. Lee, S. Seo, M. J. Lee, D. H. Seo, D. S. Suh, J. C. Park, S. O. Park, H. S. Kim, I. K. Yoo, U-In Chung, and J. T. Moon. Highly scalable non-volatile resistive memory using simple binary oxide driven by asymmetric unipolar voltage pulses. In *Proc. Intl. Electron Devices Mtg.*, pages 587–590, Dec. 2004.

[22] A. Chen, S. Haddad, Y-C. Wu, T-N. Fang, Z. Lan, S. Avanzino, S. Pangrle, M. Buynoski, M. Rathor, W. Cai, N. Tripsas, C. Bill, M. VanBuskirk, and M. Taguchi. Non-volatile resistive switching for advanced memory applications. In *Proc. Intl. Electron Devices Mtg.*, pages 746–749, Dec. 2005.

[23] J. J. Yang, M. D. Pickett, X. Li, D. A. A. Ohlberg, D. R. Stewart, and R. S. Williams. Memristive switching mechanism for metal/oxide/metal nanodevices. *Nature Nano.*, 3:429–433, 15 June 2008.

[24] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams. The missing memristor found. *Nature*, 453:80–83, 1 May 2008.

[25] K. Galatsis, K. Wang, Y. Botros, Y. Yang, Ya-H. Xie, J. F. Stoddart, R. B. Kaner, C. Ozkan, J. Liu, M. Ozkan, C. Zhou, and K. W. Kim. Emerging memory devices: Nontraditional possibilities based on nanomaterials and nanostructures. *IEEE Circ. and Dev. Mag.*, 22(3):12–21, 2006.

[26] R. J. Tseng, J. Huang, J. Ouyang, R. B. Kaner, and Y. Yang. Polyaniline nanofiber/gold nanoparticle nonvolatile memory. *Nano Lett.*, 5(6):1077–1080, 2005.

[27] V. Betz. The FPGA place-and-route challenge. Available online at `http://www.eecg.toronto.edu/~vaughn/challenge/challenge.html`.

[28] M. M. Ziegler, C. A. Picconatto, J. C. Ellenbogen, A. DeHon, D. Wang, Z. H. Zhong, and C. M. Lieber. Scalability simulations for nanomemory systems integrated on the molecular scale. In *Molecular Electronics III*, volume 1006 of *Ann. N.Y. Acad. Sci.*, pages 312–330. Blackwell Publ., 2003.

[29] S. Das, G. S. Rose, M. M. Ziegler, C. A. Picconatto, and J. C. Ellenbogen. Architectures and simulations for nanoprocessor systems integrated on the molecular scale. *Lect. Notes Phys.*, 680:479–512, 2005.

[30] S. Das, C. A. Picconatto, G. S. Rose, M. M. Ziegler, and J. C. Ellenbogen. System-level design and simulation of nanomemories and nanoprocessors. In S. E. Lyshevski, editor, *CRC Nano and Molecular Electronics Handbook*.

CRC Press, 2007.

[31] Cadence Design Framework II, Version IC 5.0.33. Cadence Design Systems, Inc., San Jose, CA, 2004.

[32] D. R. Stewart, D. A. A. Ohlberg, P. A. Beck, Y. Chen, R. Stanley Williams, J. O. Jeppesen, K. A. Nielsen, and J. Fraser Stoddart. Molecule-independent electrical switching in Pt/organic monolayer/Ti devices. *Nano Lett.*, 4(1):133–136, 2004.

[33] W. Steinhögl, G. Schindler, G. Steinlesberger, and M. Engelhardt. Size-dependent resistivity of metallic wires in the mesoscopic range. *Phys. Rev. B*, 66(7), 2002.

[34] V. Degalahal and T. Tuan. Methodology for high level estimation of FPGA power consumption. In *Proc. ASP-DAC*, pages 657–660, 2005.

[35] T. Tuan and B. Lai. Leakage power analysis of a 90-nm FPGA. In *Proc. Custom Int. Circuits Conf.*, pages 57–60, 2003.

[36] S. Sharp. Spartan-3 vs. Cyclone II performance analysis. White Paper WP226, Xilinx, Inc., 2005.