Design of Programmable Interconnect for Sublithographic Programmable Logic Arrays

André DeHon
Dept. of CS, 256-80
California Institute of Technology
Pasadena, CA 91125
andre@acm.org

ABSTRACT

Sublithographic Programmable Logic Arrays can be interconnected and restored using nanoscale wires. Building on a hybrid of bottom-up assembly techniques supported by conventional lithographic patterning, we show how modest-sized PLA logic blocks, which are efficient for implementing logic, can be organized into a segmented, Manhattan mesh interconnection scheme. The resulting programmable architecture has a macro-scale view which is reminiscent of lithographic FPGA and CPLD designs despite the fact that the low-level, sublithographic fabrication techniques used are much more highly constrained than conventional lithography and are prone to high defect rates. Using the Toronto 20 benchmark set, we begin to explore the design space for these sublithographic architectures and show that they may allow us to exploit nanowire building blocks to reach one to two orders of magnitude greater density than 22nm CMOS lithography.

Categories and Subject Descriptors

B.6.1 [**Logic Design**]: Design Styles—*logic arrays*; B.7.1 [**Integrated Circuits**]: Types and Design Styles—*advanced technologies*

General Terms

Design

Keywords

Sublithographic architecture, nanowires, programmable logic arrays, programmable interconnect, Manhattan mesh

1. INTRODUCTION

Many research efforts are now contemplating nanoscale systems with wires which are a small number (*e.g.* 5–20) of atoms in diameter and switching elements which fit into the space of a single such wire crossing (*e.g.* [10, 29]). The fabrication and operation of key nanowire building blocks [18, 19, 37] and devices [13, 30, 42] have been demonstrated, and techniques for assembling them into larger, interconnected ensembles are under development [31, 44, 45].

Owing to the fabrication regularity demanded by these bottom-up assembly techniques, regular, crossbar-like architectures are emerging as one of the most promising strategies for organizing these atomic-scale building blocks into usable logic [23, 27, 35, 41]. This naturally suggests Programmable Logic Arrays (PLAs) as the key logic building block. From our long experience building PLAs in VLSI, we know that a single, monolithic PLA cannot exploit the structure in most logic functions. Since exploiting this structure is necessary to achieve compact implementations, in conventional silicon this has led us to organizations where modest size PLA blocks are interconnected using a programmable wiring network (*e.g.* [32], Altera's MAX 7000 series [2], Xilinx's XC9500 family [48]).

In this work we introduce a detailed design architecture for interconnecting nanoPLA building blocks. The regular architecture is compatible with emerging techniques for bottom-up assembly and patterning of sublithographic-pitch nanowires. Our designs allow the nanoPLA building blocks to communicate with each other without transitioning to lithographic scale logic, allowing us to exploit the tight pitch of nanowires to provide compact wiring. Nonetheless, the nanoscale logic can be integrated with lithographic scale logic and communicate with it. Driven by the demand for fabrication regularity at the sublithographic scale, our designs merge routing and logic resources more intimately than conventional, lithographic architectures resulting in a minimal number of distinct features which compose the nanoPLA tile.

Novel contributions of this work include:

- Detailed formulation of programmable interconnect for nanoPLAs
- Architecture for I/O between lithographic-scale CMOS and nanoPLA array
- Complete tool flow for mapping from conventional logic netlists to placed and routed nanoPLA designs
- Area and delay models for interconnected nanoPLAs
- Assessment of the impact of stochastic vs. deterministic feature construction
- Mapped logic, net density comparison between these interconnected nanoPLAs and conventional, lithographic FPGAs using the Toronto 20 benchmark suite

FPGA'05, February 20–22, 2005, Monterey, California, USA.

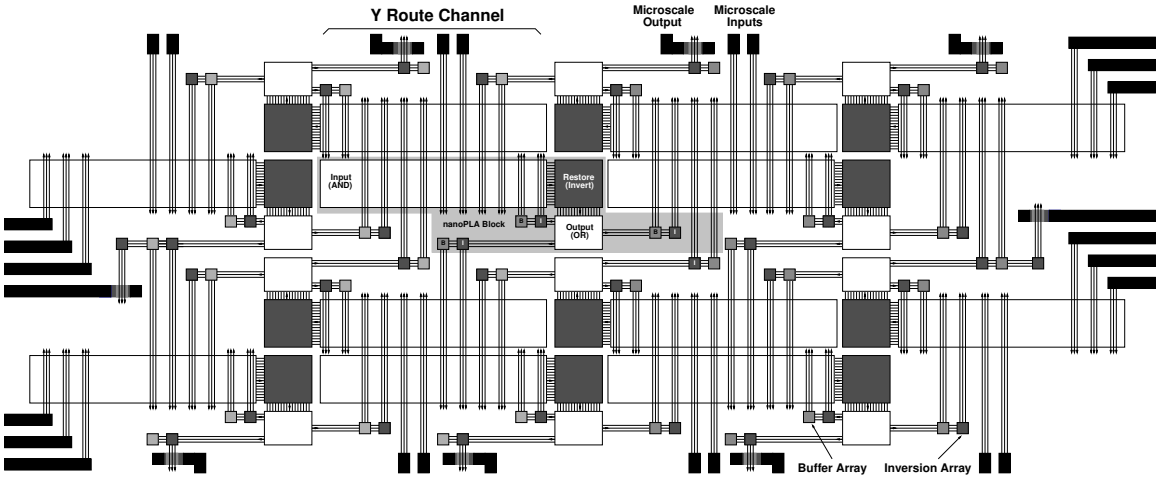


Figure 1: nanoPLA Cluster Tiling with Edge IO to Lithographic Scale

We start by reviewing the basic nanoPLA structure, building blocks, and fabrication techniques (Section 2). Section 3 describes the new interconnect architecture. In Section 4, we examine the impact of technology parameters and array sizing. Section 5 details our CAD flow for mapping designs to these nanoPLAs. We map the Toronto 20 benchmark set for a wide range of parameters and report the density achievable with these sublithographic architectures as compared to conventional FPGAs in Section 6.

2. BACKGROUND

2.1 nanoPLA

The designs in this work build on the nanoPLA design introduced by DeHon and Wilson [26]. The main features from this earlier design remain in the nanoPLA logic block used here (See Figure 3). The nanoPLA is built from a crossed set of N-type and P-type nanowires. Decorated nanowires are grown from seed catalysts that define their diameter [19]; nanowires with diameters down to 3nm have been demonstrated [46]. These nanowires are assembled into crossed arrays using flow alignment and/or Langmuir-Blodgett techniques [45]. Electrically switchable, diode crosspoints provide a programmable wired-OR plane which can be used to configure arbitrary logic into the PLA and avoid defective nanowires. The output of the OR plane becomes the input of a restoration plane which is realized by field-effect control of nanowire conduction [28, 30]. The restoration plane can buffer or invert the input. The resulting logic can be clocked using the precharge and enable field-effect control on the nanowires [26] (See Enable/Precharge gating in Figure 3). Lithographic scale wires along with stochastically coded nanowire addresses allow us to electrically reach into the nanoPLA and program individual crosspoint junctions [25].

2.2 Crosspoint Arrays and Defects

Many technologies have been demonstrated for non-volatile, switched crosspoints. So far, they all seem to have: (1) resistance which changes significantly between "on" and "off" states, (2) the ability to be made rectifying, and (3) the ability to turn the device "on" or "off" by applying a voltage differential across the junction. Rueckes *et al.* demonstrate switched devices using suspended nanotubes to realize a bistable junction with an energy barrier between the two states [39]. In the "off" state the junction exhibits only small tunneling current (high resistance $\sim$GΩs); when the devices are in contact in the "on" state, there is little resistance ($\sim$100KΩ) between the tubes. Separately, UCLA and HP have demonstrated a number of molecules which exhibits hysteresis [15, 16]. HP has demonstrated an 8×8 crossbar made from [2]rotaxane molecules and observed that they could force an order of magnitude resistance difference between "on" and "off" state junctions [12].

An early crossbar based on molecular switchpoints demonstrated that 85% of the junctions were programmable [12]. Naeimi and DeHon [38] and Snider, Kuekes, and Williams [41] show that this rate of crosspoint defects can easily be tolerated with modest impact on device size. Naeimi shows the population overhead is below 10% for non-programmable crosspoint defect rates up to 20%.

2.3 Sublithographic Fabrication Strategy

The basic fabrication strategy is as follows:

1. Prepare individual nanowires – grow nanowires [19, 37] with axial differentiation [28] and radial differentiation [33]. A common radial differentiation is to place an oxide shell around the (semi-) conducting nanowire core.
2. Prepare a lithographic substrate with a flat surface.
3. Use Langmuir-Blodgett techniques to align nanowires in a single direction, tight pack them, and transfer them to a surface [45]. The oxide shell defines the spacing between nanowire conductors (See Figure 2).
4. Lithographically etch breaks in the nanowires to distinguish conduction regions (See Figure 2).
5. Use directional or timed lithographic etches to remove the oxide coating and expose the (semi-) conducting core of the nanowires where appropriate (*e.g.* contacts, some crosspoints) [44].
6. Lithographically mask and deposit metal coatings and anneal to convert desired portions of nanowires into metal silicide [47].
7. Use Langmuir-Bloldgett techniques to construct and transfer a uniform layer of molecules over the nanowire conductors, if appropriate [8].
8. Repeat the Langmuir-Blodgett transfer of an orthogonal layer of nanowires to provide crossed nanowires (See Figure 2).

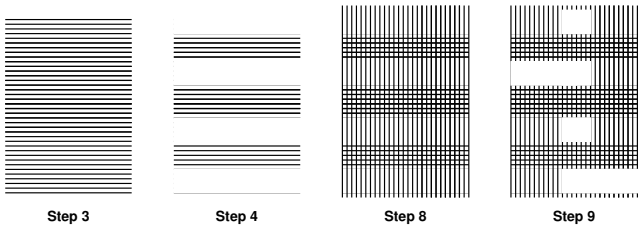


Figure 2: Illustration of Key Steps in Sublithographic Fabrication Strategy

9. Repeat metal silicide conversion.
10. Repeat Lithographically defined etching to segment the orthogonal nanowire layer and expose their ends appropriately (See Figure 2).
11. Add additional lithographic layers for contacts.

As a result, we can have tight pitch nanowires in both directions. We can deterministically define the extents of these regions by lithographic etches. We cannot deterministically cut nanowires, define their lengths, or place contacts on nanowires below the lithographic resolution. We can differentiate the nanowires at nanowire pitch by defining features in the nanowires using timed growth when the nanowires are initially prepared [25, 26, 28]. As a result, we are driven to regular architectures which use a large number of parallel nanowires; the length of the nanowires is of lithographic scale, as is the width of the ensemble of parallel nanowires.

3. INTERCONNECT ARCHITECTURE

3.1 Basic Idea

The key idea for interconnecting nanoPLA blocks is to overlap the restored output nanowires from each such block with the wired-OR input region of adjacent nanoPLA blocks (See Figure 1). In turn, this means each nanoPLA block receives inputs from a number of different nanoPLA blocks. With multiple input sources and outputs routed in multiple directions, this allows the nanoPLA block to also serve as a switching block. By arranging the overlap appropriately, we can configure the array of nanoPLAs to route signals between any of the blocks in the array.

3.2 nanoPLA Block

Input Wired-OR Region One or more regions of programmable crosspoints serves as the input to the nanoPLA block. Figures 1 and 3 show a nanoPLA block design with a single such input region. The inputs to this region are restored output nanowires from a number of different nanoPLA blocks. The programmable crosspoints allow us to select the inputs which participate in each logical product term (PTERM). The output nanowires from this region that implement the PTERMs perform a wired OR on the inputs associated with all crosspoints which are programmed into the low-resistance, "on" state.

Internal Inversion and Restoration Array The nanowire outputs from the input block cross a set of orthogonal nanowires each coded with a single, field-effect controllable region. The field-effect region allows conduction through each crossed nanowire to be gated by a single nanowire input. The output nanowires of this region are oxide coated and only load the inputs, the wired-OR outputs of the input block, capacitively such that the inputs are isolated from the outputs. We arrange the restoration logic at this stage to be inverting so that we provide the logical NOR of the selected input signals into the second plane of the nanoPLA.

"OR" Plane The restored outputs from the internal inversion plane become inputs to a second, programmable crosspoint region. Physically, this region is the same as the input plane. Each nanowire in this plane computes the wired OR of one or more of the restored PTERMs computed by the input plane.

Selective Output Inversion The outputs of this plane are then restored in the same way as the internal restoration plane. On this output, however, we have two restoration arrays. One provides the buffered (non-inverted) sense of the OR output and the second provides the inverted sense. This gives us both polarities of each output so we can provide them to the succeeding input planes (See [26] for circuit and operational details). This selective inversion plays the same role as a local inverter on the inputs of conventional, VLSI PLA; here we place it with the output to avoid introducing an additional logic plane into the design.

Taken together, the two planes provide NOR-NOR logic. This is logically equivalent to an OR-AND arrangement. With the selective inversion on the outputs, we can strategically invert the signals and use the appropriate DeMorgan's equivalents to view this as a conventional AND-OR PLA.

Feedback As shown in Figure 3, one set of outputs from each nanoPLA block feeds back to its own input region. This completes a PLA cycle similar to the designs in [26]. These feedback paths serve the role of intra-cluster routing similar to internal feedback in conventional, Island-style [7] FPGAs. The nanoPLA block implements registers by routing signals around the feedback path [26]; with separate precharge and evaluation of each of the planes, the register design is similar to two-phase clocked register design in conventional VLSI circuits. We can route signals around this feedback path multiple times to form long register delay chains for data retiming.

3.3 Interconnect

Block Outputs In addition to self feedback, output groups are placed on either side of the nanoPLA block and can be arranged so they cross input blocks of nanoPLA blocks above or below the source nanoPLA block (See Figure 1). Like segmented FPGAs [5, 9] output groups can run across multiple nanoPLA block inputs (*cf.* Connection Boxes) in a given direction. The nanoPLA block shown in Figure 3 has a single output group on each side, one routing up and the other routing down. We will see that the design shown is sufficient to construct a minimally complete topology.

Since the output nanowires are directly the outputs of gated fields: (1) an output wire can be driven from only one source, and (2) it can only drive in one direction. Consequently, unlike segmented FPGA wire runs, we must have directional wires which are dedicated to a single producer. If we coded multiple control regions into the nanowire runs, conduction would be the AND of the producers crossing the coded regions. Single direction drive arises from the fact that one side of the gate must be the source logic signal being gated; so the logical output is only available on the opposite side of the controllable region. Interestingly, recent work suggests that conventional, VLSI-based FPGA designs would also benefit from directional wires [34].

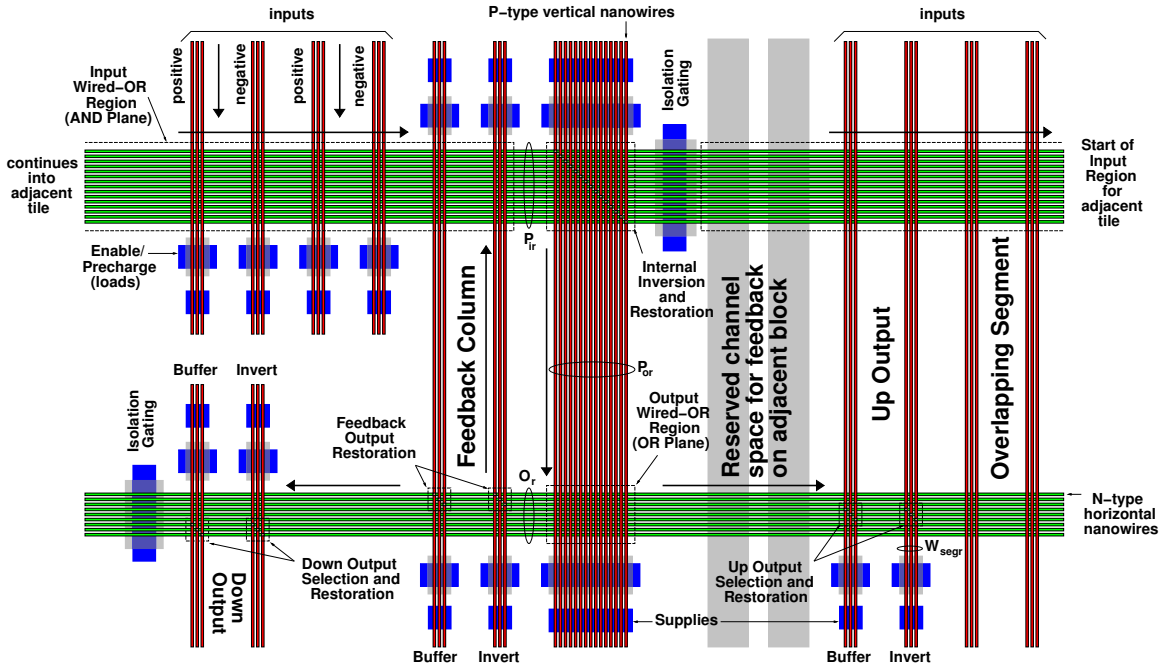


Figure 3: nanoPLA Logic Block Tile

Y Route Channels With each nanoPLA block producing output groups which run one or more nanoPLA block heights above or below the array, we end up with vertical routing channels between the logic cores of the nanoPLA blocks (See Figure 1). The segmented, nanowire output groups allow a signal to pass a number of nanoPLA blocks. For longer routes, the signal may be switched and rebuffered through a nanoPLA block (See Figure 4). Because of the output directionality, we end up with separate sets of wires for routing up and routing down in each channel.

X Routing While Y route channels are immediately obvious in Figure 1, the X route channels are less apparent. All X routing occurs through the nanoPLA block. As shown in Figure 3, we place one output group on the opposite side of the nanoPLA block from the input. In this way, one can route in the X direction by going through a logic block and configuring the signal to drive a nanowire in the output group on the opposite side of the input. If all X routing blocks had their inputs on the left, then we would only be able to route from left to right. To allow both left-to-right and right-to-left routing, we alternate the orientation of the inputs in alternate rows of the nanoPLA array (See Figures 1 and 4). In this manner, even rows provide left-to-right routing while odd rows allow right-to-left routing.

Relation to Island-Style Manhattan Design Logically viewed, the nanoPLA block is very similar to conventional, Island-style FPGA designs, especially when the Island-style designs use directional routing [34]. As shown in Figure 4, we have X and Y routing channels, with switching to provide X-X, Y-Y, and X-Y routing.

3.4 CMOS IO

These nanoPLAs will be built on top of a lithographic substrate. The lithographic circuitry and wiring provides a reliable structure from which to probe the nanowires to map their defects and to configure the logic [24–26, 38].

For input and output to the lithographic scale during operation, we can provide IO blocks to connect the nanoscale logic to lithographic scale wires much as we connect lithographic scale wires to bond pads on FPGAs. As shown in Figure 1, the simplest arrangement resembles the traditional, edge IO form of a symmetric FPGA with inputs and outputs attached to nanowires at the edges of the routing channels.

Nanowire inputs can easily be driven directly by lithographic scale wires. Since the lithographic-scale wires are wider pitch, a single lithographic wire will connect to a number of nanowires. With the lithographic wire connected to the nanowires, the nanowire crosspoints in the nanoPLA block inputs can be programmed in the same way they are for nanowire inputs.

It is possible to connect outputs in a similar manner. Such a direct arrangement could be particularly slow as the small nanowires must drive the capacitance of a large, lithographic-scale wire. Alternately, the nanowires can be used as gates on a lithographic-scale Field-Effect Transistor (FET) (See Figure 5). In this manner, the nanowires are only loaded capacitively by the lithographic-scale output and only for a short distance. We can tune the nanowire thresholds and the lithographic FET thresholds into comparable voltage regions so the nanowires can drive the lithographic FET at adequate voltages for switching. As shown, multiple nanowires will cross the lithographic-scale gate. The OR-terms driving these outputs are all programmed identically, allowing the multiple-gate configuration to provide strong switching for the lithographic-scale FET.

3.5 Parameters

Figure 6 shows the key parameters in the design of the nanoPLA block.

- W_{seg} – number of nanowires in each output group
- L_{seg} – number of nanoPLA block heights up or down which each output crosses; equivalently, the number of

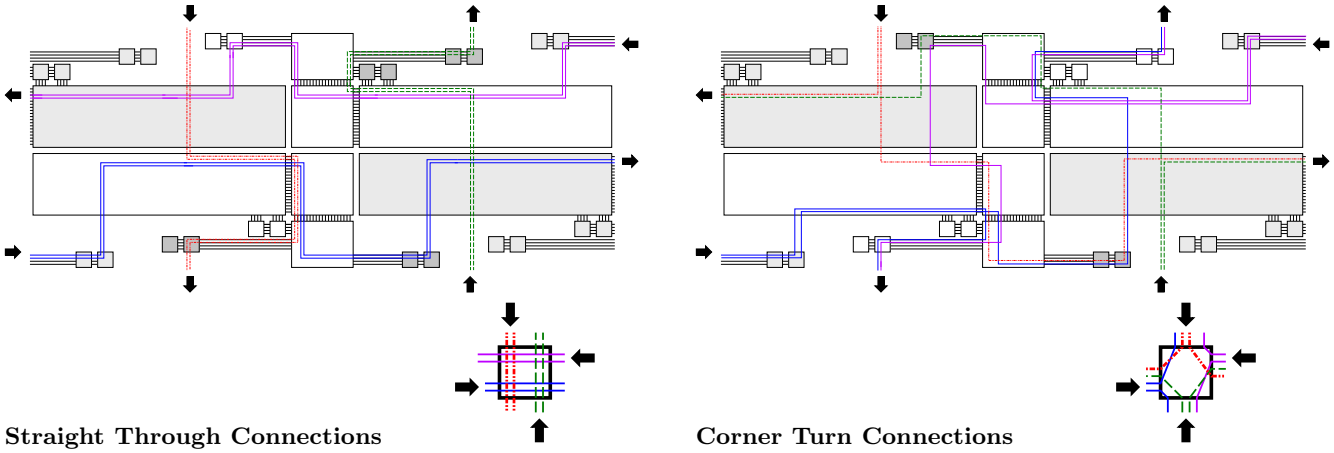


Figure 4: Routing View of nanoPLA Logic Block

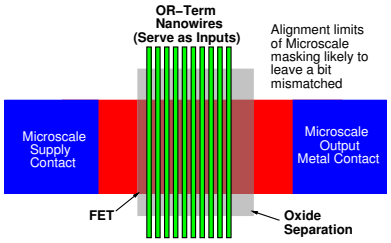


Figure 5: Nanoscale to Lithographic-Scale FET Output Structure

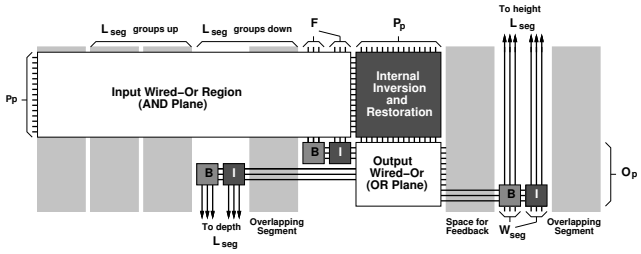


Figure 6: nanoPLA Block Parameters

parallel wire groups across each Y route channel in each direction. We show $L_{seg} = 2$ in Figure 1 and maintain $L_{seg} = 2$ throughout this paper.

- F – number of nanowires in feedback group; for simplicity we take $F = W_{seg}$ throughout this paper.
- P – number of logical PTERMS in the input (AND) plane of the nanoPLA logic block.
- O_p – number of physical outputs in the OR plane. Since each output is driven by a separate wired-OR nanowire, $O_p = 2 \times W_{seg} + F$ for the nanoPLA block we focus on in this paper with two routing output groups and a feedback output group.
- P_p – number of physical PTERMS in the input (AND) plane. Since these are also used for route-through connections, this is larger than the number of logical PTERMS in each logic block.

$$P_p \leq P + 2 \times W_{seg} + F \tag{1}$$

That is, in addition to the P logical PTERMS, we may need one physical wire for each signal that routes through the array for buffering; there will be at most O_p of these.

Additionally, we could parameterize the number and distribution of inputs (*e.g.* one side (as shown), from both sides, subsets of PTERMS from each side), the output topology (*e.g.* route both up and down on each side of the array), and segment length distributions. However, we will focus on this simple topology with $L_{seg} = 2$ for this paper. Consequently, our main physical parameters for this study will be W_{seg} and P_p.

3.6 Area

We use the following technology parameters:

- W_{litho} - lithographic interconnect pitch. *E.g.* for the 45nm node, $W_{litho} = 105$nm [1].
- W_{dnano} - nanowire pitch for nanowires which are inputs to diodes (*i.e.* Y route channel segments and restored PTERM outputs).
- W_{fnano} - nanowire pitch for nanowires which are inputs to field-effect gated nanowires; this may be larger than W_{dnano} in order to prevent inputs from activating adjacent gates and to avoid short-channel FET limitations.

From Figures 3 and 6, we can see the basic area composition of each tile.

$$\begin{aligned} TW &= (3 + 4\,(L_{seg} + 1)) \times W_{litho} \\ &\quad + (P_{or} + 4\,(L_{seg} + 1)\,W_{segr}) \times W_{dnano} \quad (2) \\ TH &= 12 \times W_{litho} + (O_r + P_{ir}) \times W_{fnano} \quad (3) \\ AW &= (N_{address} + 2) \times W_{litho} \quad (4) \\ Area &= \left(\frac{AW}{N_{share}} + TW\right) \times TH \quad (5) \end{aligned}$$

P_{or}, P_{ir}, O_r, and W_{segr} (shown in Figure 3) are the raw number of wires we need to populate in the array in order to yield P_p restored inputs, O_p restored outputs, and W_{seg} routing channels. See [24] and [26] for details of these calculations. The two 4's in TW (Tile Width) arise from the fact that we have $L_{seg} + 1$ wire groups on each side of the array (2×) and each of those is composed of a buffer/inverter selective inversion pair (2×). We charge a lithographic spacing for each of these groups since they must be etched for isolation and controlled independently by lithographic scale wires. The twelve lithographic pitches in TH (Tile Height) account for the 3 lithographic pitches needed on each side of a group of wires for the restoration supply and enable gating. Since we end and begin segmented wire runs between the input and output horizontal wire runs, we pay for these

three lithographic pitches four times in the height of a single nanoPLA block: once at the bottom of the block, twice between the blocks for segments begin/ends, and once at the top of the block (See Figure 3).

$N_{address}$ is the number of microscale address wires needed to address individual, horizontal nanoscale wires [25, 26]; for the nanoPLA blocks in this work, $N_{address}$ is typically 16–20. As introduced in [26], we share an address decoder across a number of horizontal nanoPLA blocks to amortize its cost for smaller array widths; N_{share} is the number of nanoPLA blocks which share an address decoder. Two extra wire pitches in the Address Width (AW) are the two power supply contacts at either end of a shared address run.

3.7 Delay

We keep the same, basic precharge timing model as in [26]. The two major plane evaluations are asymmetric here.

$$T_{plane} = T_{precharge} + T_{no} + T_{eval} + T_{ab} \quad (6)$$

$$T_{cycle} = T_{input\_plane} + T_{output\_plane} \quad (7)$$

The input plane will be slower because the Y route segment is longer than the internal restored PTERM segment.

The key timing is in the evaluation phase:

$$\begin{aligned} T_{in\_eval} &= R_c \times (C_{yroute} + f_{yr} \times C_{pin}) \\ &\quad + 0.5R_{yroute} \times (C_{yroute} + f_{yr} \times C_{pin}) \\ &\quad + R_{don} \times (C_{pin}) \\ &\quad + 0.5R_{pin} \times (C_{pin}) \end{aligned} \quad (8)$$

$$\begin{aligned} T_{out\_eval} &= R_c \times (C_{prestore} + f_p \times C_{out}) \\ &\quad + 0.5R_{prestore} \times (C_{prestore} + f_p \times C_{out}) \\ &\quad + R_{don} \times (C_{out}) \\ &\quad + 0.5R_{out} \times (C_{out}) \end{aligned} \quad (9)$$

f_{yr} is the maximum output fanout in the Y route channel, and f_p is the maximum PTERM fanout following PTERM restoration. Here, we refine our modeling of nanowire resistance and capacitance from [26].

With recent advances [20, 47], it appears reasonable to expect contact resistance (R_c) to come down to or below 10KΩ. If the wires are simply doped silicon, the 10μm nanowires used in these arrays will have R_{wire} (*e.g.* R_{yroute}, R_{pin}, $R_{prestore}$, R_{out}) on the order of megaohms. However, with the ability to convert the wiring portions of nanowires into Nickle-Silicide (NiSi) [47], the long Y route channel and restored PTERM (vertical) nanowires which only need a small active portion for restoration and can have their resistance reduced to around tens of kiloohms. The diode (horizontal) nanowires have long diode regions and will have hundreds of kiloohms of resistance. Total capacitance of a nanowire is around a femtofarad.

4. TECHNOLOGY AND AREA MODELS

The area of these designs will depend on a number of technology and fabrication assumptions. To calibrate ourselves on the impact of these different technology assumptions, we examine variations in three technology features as a function of our physical parameter W_{seg}.

1. lithographic pitch (W_{litho}) – we evaluate [1]:
   - current, 90nm lithography ($W_{litho} = 210$nm)
   - 45nm lithography ($W_{litho} = 105$nm)
   - 22nm lithography ($W_{litho} = 50$nm)

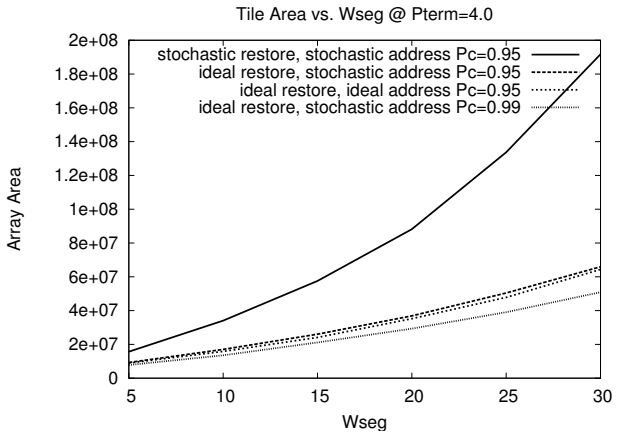


Figure 7: Impact of Stochastic vs. Deterministic Construction at W_{litho} = 105nm, W_{fnano} = W_{dnano} =10nm

2. W_{dnano}, W_{fnano} – we examine the impact of diode nanowire pitch being half the FET nanowire pitch.
3. deterministic vs. stochastic construction – we compare three different scenarios for nanowire feature construction.

Stochastic vs. Deterministic Construction As introduced in [26], we assumed the restoration array and the addressing were both stochastically populated. An ideal restoration array is simply a diagonal of enabled crosspoints which is narrow in feature size, but does not require fine pitch; consequently, it is plausible that we could deterministically manufacture an ideal restoration array by augmenting lithographic-scale masks with timed growth or etches to reduce the feature size (*e.g.* [14]). The address array does require tighter pitches and a less regular pattern, so will be a harder feature to construct deterministically. Nonetheless, to assess the of impact stochastic vs. deterministic construction, we compute the area required under various assumptions:

1. stochastic restore, stochastic address
2. ideal restore, stochastic address
3. ideal restore, ideal address

Figure 7 compares the area of the array as a function of W_{seg} for the technology point W_{litho} = 105nm, W_{fnano} = W_{dnano} =10nm. For larger arrays, we see the stochastic construction of the restoration unit costs us roughly a factor of three in density. This comes from the need to overpopulate both the input (P_{ir}, O_r) and restoration (P_{or},W_{segr}) nanowires to yield the desired P_p and W_{seg} unique wires; the larger number of raw wires also makes all the wires longer resulting in lower yield of the wires.

We also see that the ideal addressing has only a modest impact on area; in fact, as the final line in Figure 7 shows, the impact of ideal addressing is less than the impact of improving the yield in making nanoscale-to-microscale contacts (P_c). For addressing, it only takes a few additional address lines to get mostly unique addressing. The overhead area added by stochastic addressing is only these extra address lines and a small percentage of overpopulation in P_{ir} and O_r in order to accommodate the few redundant addresses which do appear. See DeHon [24] for a more detailed treatment of uniqueness versus address space size.

Feature Sizes Figure 8 shows the impact of lithographic support technology and reduced diode pitch. Overall we see

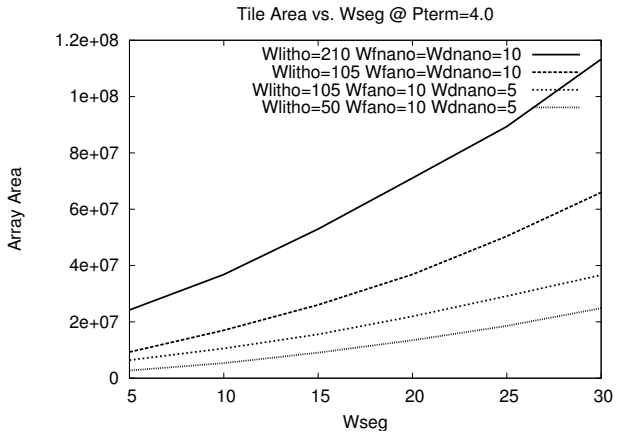
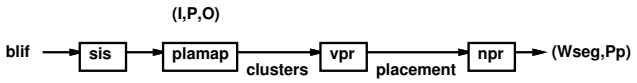


Figure 8: Impact of Technology Features for Ideal Restore, Stochastic Address Construction



[netlist format conversion steps not shown]

Figure 9: Design Automation Flow for nanoPLA Mapping

almost a factor of 9 in area difference between the 90nm lithography with 10nm nanowires and 22nm lithography with 10nm FET nanowires and 5nm diode nanowires for small arrays where the lithographic overhead dominates; for large arrays, this gap narrows just below a factor of 5. For 45nm lithography, we show both 10nm and 5nm diode nanowire pitch and see the smaller diode pitch reduces area just over 30% at the low end and over 45% at the high end.

Nanowire Lengths Figure 10 shows the lengths of the key nanowire features as a function of W_{seg} for the equal FET and diode pitch case and the reduced diode pitch case. For small arrays, nanowire lengths are 2–4μm due to the lithographic spacing required to supply and separate array features. For larger arrays, with $L_{seg} = 2$, the Y Route segment is around 10μm long; we currently expect to reliably yield assembled nanowires around 10–20μm long.

Delay Assuming on-diode resistance (R_{don}) of 100KΩ, delay is a strong function of nanowire resistance and capacitance which, in turn, is a function of nanowire length. Figure 11 shows the total cycle delay under various fanout ($f_{yr} = f_p = f$) and technology assumptions. Both graphs show that the NiSi conversion reduces delay by an order of magnitude. While address sharing improves the density of small arrays, it leaves us with long wires that are slow to precharge. Figure 11 shows sharing case on the left; since smaller arrays share addresses across long nanowires, they do not fully benefit from the reduced array width. On the right, Figure 11 show the non-shared case. With NiSi, low fanout, and no sharing, cycle delays in the hundreds of picoseconds may be feasible.

5. DESIGN AUTOMATION

To map from standard logic netlists (*e.g.* BLIF [40]) to the nanoPLA arrays, we used a combination of conventional and custom tools as shown in Figure 9. SIS [40] performs standard, technology independent optimizations and decomposes the logic into small fanin nodes for covering. PLAMAP [11] covers the logic into (I,P,O) PLA clusters, where:

- I - number of inputs to PLA block
- P - number of PTERMS in PLA block
- O - number of outputs to PLA block

These clusters can then be placed with VPR [3, 4]. While VPR can also route designs, the routing architecture for the nanoPLA array is sufficiently different to merit separate treatment. Consequently, we developed our own nanoPLA router (npr) for routing. Along with a route, npr returns the key physical design parameters W_{seg} and P_p.

The cluster mapping variables to PLAMAP (I,P,O) only account for the logical mapping. I and O will impact W_{seg}; routing along with P will impact P_p.

npr The nanoPLA router is a global, directional wire router using Pathfinder-like history [36]. Since the nanoPLA inputs are effectively a fully populated crossbar, there are no detail routing limitation; inputs can be switched in from just about any channel upon which they arrive. Similarly, outputs can be placed on any wire channel by programming the output channel's wired OR appropriately in the OR plane of the PLA block. The route search proceeds through each nanoPLA logic block it encounters, accounting for the extra PTERMS required for such route-through logic so that P_p is measured and minimized.

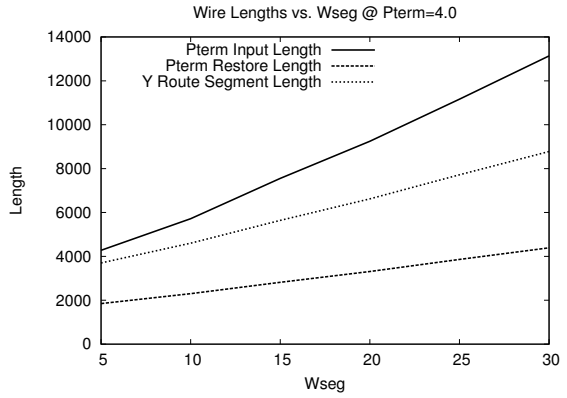
6. DESIGN SPACE EXPLORATION

To assess the density benefits of these sublithographic PLAs, we mapped 19 designs from the Toronto 20 benchmark suite [6] to various PLAs using the flow described in the previous section.

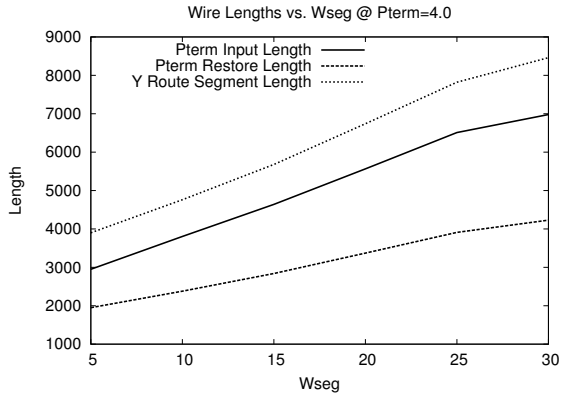
- Original source was the 4-LUT covered BLIFs for the Toronto 20 benchmark set.
- These were re-optimized using `script.algebraic` in `sis`.
- They were then decomposed with `tech_decomp` (available in the RASP Suite version of `sis` [17]).
- PLAMAP [11] mappings were performed without depth reduction. The parameter set for our exploration was I={12,14,16,18,20}, P={24,28,32,36,40,44,48}, and O={2,4,6,8}.
- We set the VPR architecture IO Ratio to 16.
- After routing with `npr`, we extracted P_p and W_{seg} and used these in the area models shown in Section 3.6.
- Areas shown in Table 1 for nanoPLA designs are for the entire rectangle in which the design placed and routed.

Table 1 rounds up the minimum area mappings and compares them to lithographic 4-LUT FPGAs at the 22nm node [1]. For the 4-LUT areas in 22nm, we took the known LUT counts and simply multiplied these by 10^8nm^2. Here we make use of the fact that 4-LUT blocks run about 1Mλ^2 [21], with $\lambda \approx$ 11nm for the 22nm roadmap node. We round 121×10^6nm^2 to 10^8nm^2 for the estimation used here. Electrical length in Table 1 is the sum of the Y route channel length ($2 \cdot TH$) and the PTERM input length (TW).

Table 1 shows that routed nanoPLA designs are one to two orders of magnitude smaller than 22nm lithographic FPGAs. The fact that many designs achieve their minimum area point at the extremes of the explored parameter suggests the need to expand the parameter search further to see the full potential density benefits.

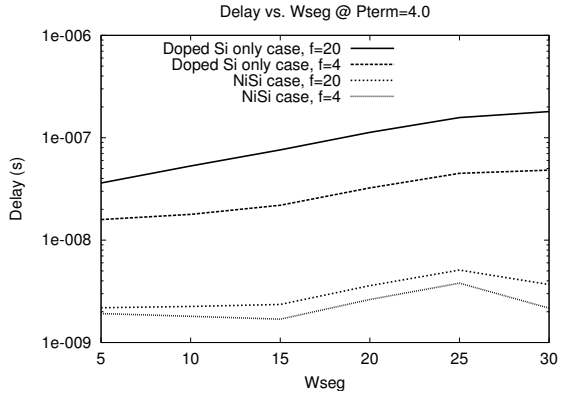


$W_{dnano} = 10$nm

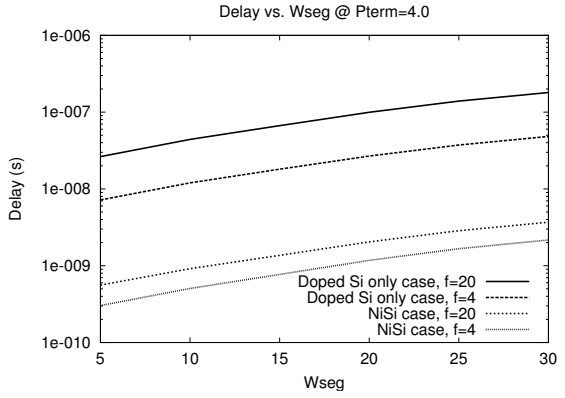


$W_{dnano} = 5$nm

Figure 10: Nanowire Lengths as a Function of W_{seg} for Ideal Restore, Stochastic Address Case with $W_{litho} = 105$nm, $W_{fnano} = 10$nm



Shared Addresses



No Address Sharing

Figure 11: Delay as a Function of W_{seg} for Ideal Restore, Stochastic Address Case with $W_{litho} = 105$nm, $W_{fnano} = 10$nm, $W_{dnano} = 5$nm

	Parameters						22nm	nanoPLA			Electrical
	Map			Physical			CMOS	array	Area	Area	Length
Design	**I**	**P**	**O**	P_p	W_{seg}	4-LUTs	($\times 10^{11}$nm^2)	org.	($\times 10^8$nm^2)	**Ratio**	(μm)
alu4	18	44	2	60	8	1522	1.52	5×5	2.8	547	8.8
apex2	20	24	8	54	15	1878	1.88	14×14	28.3	66	9.8
apex4	12	48	2	62	7	1262	1.26	6×6	3.9	325	8.8
bigkey	16	24	8	44	13	1707	1.71	11×11	15.1	112	9.1
clma	20	48	8	104	28	8382	8.38	23×23	161.6	51	13.7
des	18	28	8	78	25	1591	1.59	12×12	35.2	45	13.0
diffeq	16	44	8	86	21	1497	1.50	11×11	27.1	55	12.3
dsip	20	24	6	58	18	1370	1.37	9×9	13.5	101	10.6
elliptic	18	24	8	78	27	3604	3.60	17×17	74.8	48	13.3
ex1010	20	48	4	66	9	4598	4.60	9×9	9.8	468	9.0
ex5p	12	32	8	67	18	1064	1.06	3×3	1.6	668	11.0
frisc	18	24	8	92	34	3556	3.56	18×18	110.8	32	14.7
misex3	18	48	4	64	8	1397	1.40	7×7	5.6	249	9.0
pdc	16	48	8	74	13	4575	4.58	7×7	7.5	610	10.1
s298	18	48	8	79	15	1931	1.93	8×8	11.0	176	10.8
s38417	14	32	8	76	22	6406	6.41	23×23	115.5	55	12.2
seq	20	36	8	72	18	1750	1.75	9×9	14.9	117	11.1
spla	20	44	8	68	12	3690	3.69	5×5	3.6	1025	9.8
tseng	16	28	8	78	25	1047	1.05	11×11	29.6	35	13.0

Table 1: Area Minimizing Design Points ($W_{litho} = 105$nm, $W_{fnano} = 10$nm, $W_{dnano} = 5$nm)

7. DISCUSSION AND FUTURE WORK

The results in Section 6 demonstrate promising net density for these interconnected nanoPLA architectures. The routed designs demonstrate that the simple, directional interconnect topology is adequate to provide programmable wiring for the nanoPLA logic blocks. Combining the results in Table 1 with the technology variants explored in Section 4, we can begin to see the density landscape for large-scale logic implementations beyond the limits of lithographic fabrication. The focus microarchitecture here is one of the simplest designs which is sufficient to support complete, interconnected logic; it immediately suggests many variations, some of which have already been noted in Section 3.5, which merit exploration as we refine our understanding of these architectures.

The areas in Table 1 cannot be achieved simultaneously with the best delays in Figure 11. Understanding how to optimize the nanoPLA architecture for delay and how to map to achieve the least delays remains an important area for further research. This will allow us to characterize the area-delay tradeoffs available for mapped nanoPLA designs.

A physical array will have a fixed P_p and W_{seg}. Consequently, just as we need to perform fixed wire-schedule or channel-width mapping for real FPGAs (*e.g.* [22, 43]), we will need to perform P_p and W_{seg} mapping to target a particular, interconnected nanoPLA device. Efficient algorithms and detailed analysis under these constraints are also important avenues for future work.

8. SUMMARY

Nanowire synthesis and alignment allows us to build dense nanowire arrays with tight, controlled pitches whose dimensions do not depend on our ability to lithographically image small features. These tight nanowire arrays can be carved up and controlled from the lithographic scale to realize nanoscale PLAs. By arranging the restored, OR-term outputs of nanoPLA blocks so they extend over the inputs of adjacent blocks, each nanoPLA block can provide both Manhattan routing and computation. The resulting architecture can be viewed as a mesh of PLA building blocks. We can adapt conventional PLA covering and FPGA routing tools to map arbitrary logic onto these devices. Preliminary experience mapping the Toronto 20 benchmark set suggests this allows us to reach one to two orders of magnitude in density beyond the 22nm roadmap node.

9. ACKNOWLEDGMENTS

This research was funded in part by the DARPA Moletronics program under grant ONR N00014-01-0651 and N00014-04-1-0591. Sun Microsystems donated one of the computers used to support this work. This architectural work would not have been possible or meaningful without close cooperation and support from Charles Lieber. The broad mapping study was facilitated by the active support of Deming Chen for PLAMAP. Raphael Rubin, Betta Dawson, and David LeBlanc provided rapid installation of additional computing resources which made it possible to explore the design space as broadly as reported in this work. Michael Hutton provided the BLIFs necessary to remap the Toronto 20 benchmark set for various PLA organizations. The author benefited from valuable discussions with Paul Solomon and Gary Dittlov during the development of this work.

This material is based upon work supported by the Department of the Navy, Office of Naval Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Office of Naval Research.

10. REFERENCES

[1] International Technology Roadmap for Semiconductors. <http://public.itrs.net/>, 2003.

[2] Altera Corporation, 2610 Orchard Parkway, San Jose, CA 95134-2020. *MAX 7000 Programmable Logic Device Family Data Sheet*, v6.6 edition, June 2003. <http://www.altera.com/literature/ds/m7000.pdf>.

[3] V. Betz. VPR and T-VPack: Versatile Packing, Placement and Routing for FPGAs. <http://www.eecg.toronto.edu/~vaughn/vpr/vpr.html>, March 27 1999. Version 4.30.

[4] V. Betz and J. Rose. VPR: A New Packing, Placement, and Routing Tool for FPGA Research. In W. Luk, P. Y. K. Cheung, and M. Glesner, editors, *Proceedings of the International Conference on Field-Programmable Logic and Applications*, number 1304 in LNCS, pages 213–222. Springer, August 1997.

[5] V. Betz and J. Rose. FPGA Routing Architecture: Segmentation and Buffering to Optimize Speed and Density. In *Proceedings of the International Symposium on Field-Programmable Gate Arrays*, pages 59–68, February 1999.

[6] V. Betz and J. Rose. FPGA Place-and-Route Challenge. <http://www.eecg.toronto.edu/~vaughn/challenge/challenge.html>, 1999.

[7] V. Betz, J. Rose, and A. Marquardt. *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic Publishers, 101 Philip Drive, Assinippi Park, Norwell, Massachusetts, 02061 USA, 1999.

[8] C. L. Brown, U. Jonas, J. A. Preece, H. Ringsdorf, M. Seitz, and J. F. Stoddart. Introduction of [2]Catenanes into Langmuir Films and Langmuir-Blodgett Multilayers. A Possible Strategy for Molecular Information Storage Materials. *Langmuir*, 16(4):1924–1930, 2000.

[9] S. Brown, M. Khellah, and Z. Vranesic. Minimizing FPGA Interconnect Delays. *IEEE Design and Test of Computers*, 13(4):16–23, 1996.

[10] M. Butts, A. DeHon, and S. Goldstein. Molecular Electronics: Devices, Systems and Tools for Gigagate, Gigabit Chips. In *Proceedings of the International Conference on Computer Aided Design*, pages 433–440, November 2002.

[11] D. Chen, J. Cong, M. Ercegovac, and Z. Huang. Performance-Driven Mapping for CPLD Architectures. *IEEE Transactions on Computed-Aided Design for Integrated Circuits and Systems*, 22(10):1424–1431, October 2003.

[12] Y. Chen, G.-Y. Jung, D. A. A. Ohlberg, X. Li, D. R. Stewart, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, and R. S. Williams. Nanoscale Molecular-Switch Crossbar Circuits. *Nanotechnology*, 14:462–468, 2003.

[13] Y. Chen, D. A. A. Ohlberg, X. Li, D. R. Stewart, R. S. Williams, J. O. Jeppesen, K. A. Nielsen, J. F. Stoddart, D. L. Olynick, and E. Anderson. Nanoscale

Molecular-Switch Devices Fabricated by Imprint Lithography. *Applied Physics Letters*, 82(10):1610–1612, 2003.

[14] S. Y. Chou, P. R. Krauss, W. Zhang, L. Guo, and L. Zhuang. Sub-10 nm Imprint Lithography and Applications. *Journal of Vacuum Science and Technology B*, 15(6):2897–2904, November-December 1997.

[15] C. Collier, G. Mattersteig, E. Wong, Y. Luo, K. Beverly, J. Sampaio, F. Raymo, J. Stoddart, and J. Heath. A [2]Catenane-Based Solid State Reconfigurable Switch. *Science*, 289:1172–1175, 2000.

[16] C. P. Collier, E. W. Wong, M. Belohradsky, F. M. Raymo, J. F. Stoddard, P. J. Kuekes, R. S. Williams, and J. R. Heath. Electronically Configurable Molecular-Based Logic Gates. *Science*, 285:391–394, 1999.

[17] J. Cong, D. Chen, E. Ding, Z. Huang, Y.-Y. Hwang, J. Peck, C. Wu, and S. Xu. RASP_SYN release B 2.1: FPGA/CPLD Technology Mapping and Synthesis Package. <http://ballade.cs.ucla.edu/software_release/rasp/htdocs/>, 2004.

[18] Y. Cui, X. Duan, J. Hu, and C. M. Lieber. Doping and Electrical Transport in Silicon Nanowires. *Journal of Physical Chemistry B*, 104(22):5213–5216, June 8 2000.

[19] Y. Cui, L. J. Lauhon, M. S. Gudiksen, J. Wang, and C. M. Lieber. Diameter-Controlled Synthesis of Single Crystal Silicon Nanowires. *Applied Physics Letters*, 78(15):2214–2216, 2001.

[20] Y. Cui, Z. Zhong, D. Wang, W. U. Wang, and C. M. Lieber. High Performance Silicon Nanowire Field Effect Transistors. *Nanoletters*, 3(2):149–152, 2003.

[21] A. DeHon. Reconfigurable Architectures for General-Purpose Computing. AI Technical Report 1586, MIT Artificial Intelligence Laboratory, 545 Technology Sq., Cambridge, MA 02139, October 1996.

[22] A. DeHon. Balancing Interconnect and Computation in a Reconfigurable Computing Array (or, why you don't really want 100% LUT utilization). In *Proceedings of the International Symposium on Field-Programmable Gate Arrays*, pages 69–78, February 1999.

[23] A. DeHon. Array-Based Architecture for FET-based, Nanoscale Electronics. *IEEE Transactions on Nanotechnology*, 2(1):23–32, March 2003.

[24] A. DeHon. Law of Large Numbers System Design. In S. K. Shukla and R. I. Bahar, editors, *Nano, Quantum and Molecular Computing: Implications to High Level Design and Validation*, chapter 7, pages 213–241. Kluwer, 2004.

[25] A. DeHon, P. Lincoln, and J. Savage. Stochastic Assembly of Sublithographic Nanoscale Interfaces. *IEEE Transactions on Nanotechnology*, 2(3):165–174, 2003.

[26] A. DeHon and M. J. Wilson. Nanowire-Based Sublithographic Programmable Logic Arrays. In *Proceedings of the International Symposium on Field-Programmable Gate Arrays*, pages 123–132, February 2004. Extended Version: <http://www.cs.caltech.edu/research/ic/abstracts/nanopla_fpga2004.html>.

[27] S. C. Goldstein and M. Budiu. NanoFabrics: Spatial Computing Using Molecular Electronics. In *Proceedings of the International Symposium on Computer Architecture*, pages 178–189, June 2001.

[28] M. S. Gudiksen, L. J. Lauhon, J. Wang, D. C. Smith, and C. M. Lieber. Growth of Nanowire Superlattice Structures for Nanoscale Photonics and Electronics. *Nature*, 415:617–620, February 7 2002.

[29] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams. A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology. *Science*, 280:1716–1721, June 12 1998.

[30] Y. Huang, X. Duan, Y. Cui, L. Lauhon, K. Kim, and C. M. Lieber. Logic Gates and Computation from Assembled Nanowire Building Blocks. *Science*, 294:1313–1317, 2001.

[31] Y. Huang, X. Duan, Q. Wei, and C. M. Lieber. Directed Assembly of One-Dimensional Nanostructures into Functional Networks. *Science*, 291:630–633, January 26 2001.

[32] J. Kouloheris and A. E. Gamal. PLA-based FPGA Area versus Cell Granularity. In *Proceedings of the Custom Integrated Circuits Conference*, pages 4.3.1–4. IEEE, May 1992.

[33] L. J. Lauhon, M. S. Gudiksen, D. Wang, and C. M. Lieber. Epitaxial Core-Shell and Core-Multi-Shell Nanowire Heterostructures. *Nature*, 420:57–61, 2002.

[34] G. Lemieux, E. Lee, M. Tom, and A. Yu. Directional and Single-Driver Wires in FPGA Interconnect. In *Proceedings of the International Conference on Field-Programmable Technology*, pages 41–48, December 2004.

[35] Y. Luo, P. Collier, J. O. Jeppesen, K. A. Nielsen, E. Delonno, G. Ho, J. Perkins, H.-R. Tseng, T. Yamamoto, J. F. Stoddart, and J. R. Heath. Two-Dimensional Molecular Electronics Circuits. *ChemPhysChem*, 3(6):519–525, 2002.

[36] L. McMurchie and C. Ebling. PathFinder: A Negotiation-Based Performance-Driven Router for FPGAs. In *Proceedings of the International Symposium on Field-Programmable Gate Arrays*, pages 111–117. ACM, February 1995.

[37] A. M. Morales and C. M. Lieber. A Laser Ablation Method for Synthesis of Crystalline Semiconductor Nanowires. *Science*, 279:208–211, 1998.

[38] H. Naeimi and A. DeHon. A Greedy Algorithm for Tolerating Defective Crosspoints in NanoPLA Design. In *Proceedings of the International Conference on Field-Programmable Technology*, pages 49–56. IEEE, December 2004.

[39] T. Rueckes, K. Kim, E. Joselevich, G. Y. Tseng, C.-L. Cheung, and C. M. Lieber. Carbon Nanotube Based Nonvolatile Random Access Memory for Molecular Computing. *Science*, 289:94–97, 2000.

[40] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. Sangiovanni-Vincentelli. SIS: A System for Sequential Circuit Synthesis. UCB/ERL M92/41, University of California, Berkeley, May 1992.

[41] G. Snider, P. Kuekes, and R. S. Williams. CMOS-like Logic in Defective, Nanoscale Crossbars. *Nanotechnology*, 15:881–891, June 2004.

[42] D. R. Stewart, D. A. A. Ohlberg, P. A. Beck, Y. Chen, R. S. Williams, J. O. Jeppesen, K. A. Nielsen, and J. F. Stoddart. Molecule-Independent Electrical Switching in Pt/Organic Monolayer/Ti Devices. *Nanoletters*, 4(1):133–136, 2004.

[43] R. Tessier and H. Giza. Balancing Logic Utilization and Area Efficiency in FPGAs. In R. W. Hartenstein and H. Grübacher, editors, *Proceedings of the International Conference on Field-Programmable Logic and Applications*, number 1896 in LNCS, pages 535–544. Springer, August 2000.

[44] D. Whang, S. Jin, and C. M. Lieber. Nanolithography Using Hierarchically Assembled Nanowire Masks. *Nanoletters*, 3(7):951–954, July 9 2003.

[45] D. Whang, S. Jin, Y. Wu, and C. M. Lieber. Large-Scale Hierarchical Organization of Nanowire Arrays for Integrated Nanosystems. *Nanoletters*, 3(9):1255–1259, September 2003.

[46] Y. Wu, Y. Cui, L. Huynh, C. J. Barrelet, D. C. Bell, and C. M. Lieber. Controlled Growth and Structures of Molecular-Scale Silicon Nanowires. *Nanoletters*, 4(3):433–436, 2004.

[47] Y. Wu, J. Xiang, C. Yang, W. Lu, and C. M. Lieber. Single-Crystal Metallic Nanowires and Metal/Semiconductor Nanowire Heterostructures. *Nature*, 430:61–64, July 1 2004.

[48] Xilinx, Inc., 2100 Logic Drive, San Jose, CA 95124. *XC9500XV Family High-Performance CPLD Data Sheet*, v2.1 edition, June 2002. DS049 <http://www.xilinx.com/bvdocs/publications/ds049.pdf>.