Nanoscale Application Specific Integrated Circuits

Csaba Andras Moritz, Pritish Narayanan, and Chi On Chui[1]

1. Fabric Introduction

This chapter provides an overview of the Nanoscale Application Specific IC (NASIC)[2] fabric. The NASIC fabric has spun several research directions by multiple groups. This overview is a snapshot of the thinking, techniques and some of the results as of to date. NASICs is targeted as a CMOS-replacement technology. The project encompasses aspects from the physical layer and manufacturing techniques, to devices, circuits and architectures, and is funded by NSF, and FENA/FCRP and CHM/NSEC nanotechnology centers[3].

NASICs rely on 2-D grids of semiconductor nanowires with computational streaming supported from CMOS. A fabric-centric mindset or integrated approach across devices (i.e., state variables), circuit style, manufacturing techniques and architectures is followed. This mindset is anchored in a belief that at nanoscale the underlying fabric, broadly defined as the state variable in conjunction with the circuit style scalable into large-scale computational systems and its associated manufacturing approach, rather than the device alone, is how significant progress could be made in system-level capabilities.

This work is in contrast with efforts that focus on CMOS-replacement devices essentially preserving both CMOS circuit and manufacturing paradigms intact, or directions focusing on the state variables first and then attempting to connect them into a functional structure or circuit. This latter approach might not be feasible: nanoscale devices typically consist of a few atoms and interconnecting such structures post-device formation is a real challenge and might not even be possible. Assembling a fabric with self-assembly would require a mindset where the devices are formed at the same time as the main interconnect. While other more conventional approaches might be possible, they would come with a considerable impact on density and system level performance, potentially losing the benefit of otherwise well-performing devices.

Some of the key design objectives of NASICs include:

- 2-D nanowire grid layout to allow partial self-assembly. This is motivated by the fact that self-assembly techniques typically form regular structures;
- Integrated design of device, interconnect and circuit style to accommodate and minimize manufacturing requirements while providing an ultra-dense, high-performance, and low-power solution competitive to or surpassing projected state-of-the-art scaled CMOS at a much lower manufacturing cost;

[1] Csaba Andras Moritz and Pritish Narayanan are with the University of Massachusetts at Amherst. Chi On Chui is with the University of California, Los Angeles.

[2] A new derivative of NASICs is MagNASICs that follows magnetic physical phenomena instead of being based on semiconductors and charge-based electronics.

[3] The authors acknowledge support from the Focus Center Research Program (FCRP) - Center on Functional Engineered Nano Architectonics (FENA). This work was also supported by the Center for Hierarchical Manufacturing (CHM/NSEC 0531171) University of Massachusetts at Amherst and NSF awards 0508382, 0541066 and 0915612.

- Short interconnects that are inherently part of the fabric or means of inter-device interaction based on new physical phenomena, to bring significantly better power efficiency than in CMOS designs.
- Streaming control mechanism with peripheral CMOS support but all logic implemented at nanoscale. The goal is to minimize device types, eliminate need for sizing and arbitrary routing, and reduce manufacturing requirements at the nanoscale while providing flexibility in circuit style and achieving better noise margins.
- Reliance on built-in fault masking techniques to mask defects, parameter variation, and noise related faults. Support in all circuits for 10-15% defect rates (roughly 10 orders of magnitude higher rate than in CMOS, e.g., at 90 nm CMOS has 0.4 defects per cm^2 vs. 100s of millions of defects at the rates assumed in NASICs per cm^2). Support for 3σ = 30% parameter variation: this is for compensating for imperfections and device parameter variations in the fabric. This assumption further reduces requirements for precision in manufacturing. The built-in fault masking direction does not exclude using reconfigurable devices in the future; the belief of the group was that the challenges with reconfiguration are still at the material science level. Furthermore, certain types of faults – e.g., due to device parameter variation and noise - will be difficult to overcome even if reconfiguration is made possible with materials such as rotaxane [1], amorphous silicon and others [2].
- The simpler manufacturing and fault tolerance could not only reduce costs but also allow more aggressive scaling that can mean higher performance, density, and power efficiency.
- New types of applications such as neuromorphic architectures resembling brain-like information processing and new types of nanoprocessors focusing on streaming could directly exploit capabilities in these more regular nanoscale fabrics. Regular architectures and in general architectures with minimal feedback paths are better suited to fabrics where arbitrary routing is not supported.

This chapter is organized as follows. First we describe the core components of the NASIC fabric including the underlying materials, semiconductor nanowires, and cross-nanowire-devices (Section 2). This is followed by a discussion on circuit style and methodology for streaming from CMOS, and ways to preserve noise margins during cascading (Section 3). We also discuss what this means to device level work and describe a methodology for integrated device-circuit exploration at nanoscale. Additional sections discuss NASIC logic styles (Section 4), applications designed for NASICs such as a nanoprocessor and an image processing architecture (Section 5), built-in defect tolerance and parameter variation mitigation techniques (Section 6), power and performance implications (Section 7), and ongoing manufacturing efforts (Section 8).

2. NASIC Building Blocks: Nanowires and xnwFETs

The building blocks of a computational fabric are its materials and active devices. For example, advanced CMOS technologies are built on Silicon or Silicon-on-insulator (SOI) substrates; with MOSFET devices and copper interconnect. To help readers develop an understanding of the NASIC fabric, this section provides an overview of the fundamental NASIC building blocks: Semiconductor Nanowires and Crossed Nanowire Field Effect Transistors (xnwFETs).

2.1 Semiconductor Nanowires

Semiconductor nanowires are nanostructures made of semiconductor material with diameters typically between 2nm–100nm. Nanowires can be grown to up to a few microns in length and have been shown with a variety of materials including Silicon, Germanium, Zinc Oxide, Indium Phosphide and Indium Antimonide [3][4][5][6][7][8][9][10].

Semiconductor nanowires are very promising candidates for nanoscale electronics for the following reasons:

- Early studies of the electrical characteristics of nanowires have shown diode-like and field effect transistor (FET) behavior [10].
- Techniques for growth and assembly of nanowires on to a substrate have been shown in laboratory settings. Scalable assembly is an important criterion for realizing nano-systems. (A discussion of nanowire assembly techniques is presented in Section 8 as part of the NASIC manufacturing pathway).
- It is possible to control the electrical properties, size, composition and doping of nanowires during the growth process [3].
- Techniques such as ion implantation and silicidation [11] can be used to create high conductivity interconnect regions on nanowires.
- High density, high performance integrated systems may be possible since nanowires can be synthesized to ultra-small dimensions.

More information about the synthesis, assembly and properties of nanowires can be found in various review papers and textbooks including [3][12]. In this chapter, we explore how integrated systems can be built given certain electrical properties and techniques for assembly of nanowires and the benefits of such systems over end-of-the-roadmap CMOS.

2.2 xnwFET Devices

A Crossed Nanowire Field Effect Transistor (xnwFET) is a transistor consisting of two perpendicular nanowires separated by a dielectric; one nanowire acts as gate and the other as channel. Figure 1 shows a schematic of an xnwFET. As with any field effect transistor, the electric field/voltage applied at the gate attracts or repels charge carriers in the channel, modulating its conductivity. Prototyping xnwFET behavior has been experimentally demonstrated for Silicon and Gallium Nitride nanowires in [10][13].

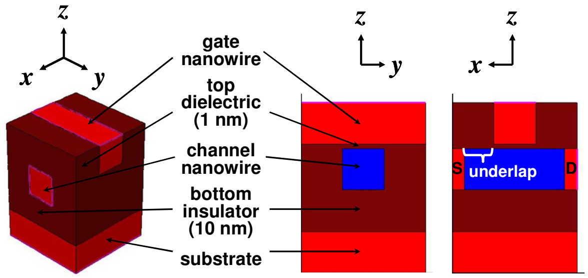


Figure 1. Crossed Nanowire Field Effect Transistor (xnwFET).

Figure 1 shows a baseline *n*-type xnwFET whose gate and channel nanowires have 10 nm × 10 nm cross-section. While a rectangular cross-section is shown in this figure, it must be noted that both rectangular [14] and cylindrical nanowire [3] structures are possible depending on the performance target as well as the manufacturing/ synthesis process used. In this device, the gate, source, drain, and substrate regions are all doped n^+ (doping concentration is typically 10^{20} dopants/cm^3) to reduce series and contact resistances. The channel is doped *p*-type and doping concentration choices are discussed below. This xnwFET is an inversion mode device similar to conventional MOSFETs: applying a positive voltage at the gate terminal attracts electrons into the *p*-doped channel leading to *n*-type FET behavior.

In the NASIC design methodology, the electrical characteristics of silicon xnwFET devices are extensively simulated using accurate physics based 3D simulations of the electrostatics and operations using Synopsys Sentaurus Device^{TM} [15]. Since any quantum confinement induced threshold voltage (V_{TH}) shift [16] is absent in the nanowire diameter of interest, the device electrostatic behavior can be accurately accounted for by the simulator. The drift-diffusion transport models employed need to be calibrated against experimental data to include effects such as carrier scattering due to surface roughness and dielectric/channel interface trapped charges. In the absence of relevant xnwFET characterization data, experimental data from well characterized nanowire channel FET with similar dimension could instead be chosen [17]. To ensure the representativeness of the calibrated models, the simulated device characteristics are further scaled to confirm similarity with other experimental nanowire FETs. Finally, current-voltage characteristics of the xnwFET device as well as capacitances can be extracted from these simulations. This data can then be incorporated into circuit models and abstracted to higher design levels for validating circuit behavior and evaluating key system level metrics.

2.2.1 Design Motivation and Nanowire-Level Approaches

Generally speaking, the electric field coupling from the gate nanowire to the channel nanowire in xnwFETs is inherently weak for the following reasons:

- The contact area through the top dielectric at the nanowire cross-point is ultra-small. Especially for the cylindrical structure, the closest contact is made only at the very center of the cross-point owing to the surface curvature in both gate and channel nanowires.
- The device channel length, i.e. the gate nanowire diameter, is roughly the same as the device channel thickness, i.e. the channel nanowire diameter.

Since poor electrostatic gate-to-channel coupling often compromises the xnwFET integrity as a switching device for circuit implementation (see Section 8 for more details), its improvement is mandated through technology-aware explorations of device design and optimization. The individual and combined effects of changing nanowire geometry, sizing, and channel doping concentration are initially analyzed.

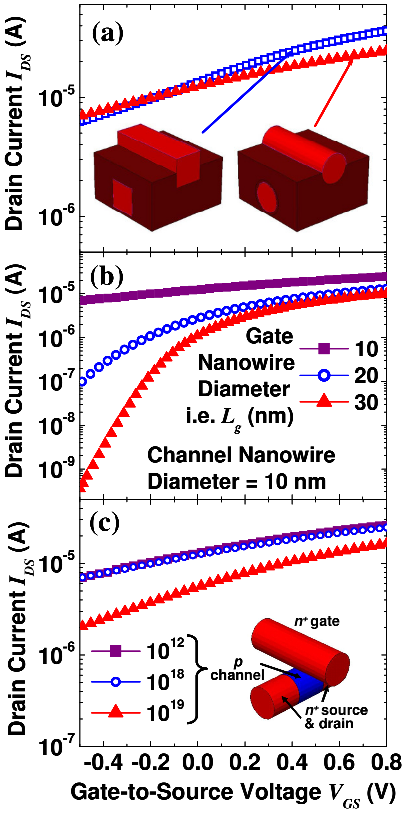


Figure 2. Simulated xnwFET transfer characteristics with different (a) geometry, (b) gate nanowire diameter, and (c) channel nanowire doping.

The simulated transfer characteristics of both rectangular and cylindrical xnwFET structures are compared in

Figure 2(a). A slightly better on-to-off state current ratio (I_{ON}/I_{OFF}) is obtained with the rectangular geometry due to a better gate coupling as anticipated. This advantage is, however, offset by the technological limitations inherent to the formation and transfer of rectangular nanowires discussed in Section 8.

The better gate control can alternatively be achieved with an increased gate-to-channel nanowire diameter ratio as illustrated in Figure 2(b). The use of nanowires with different diameter for NASIC circuits is nonetheless expected to be impractical as xnwFETs in various cascading circuit stages would need to be separately optimized.

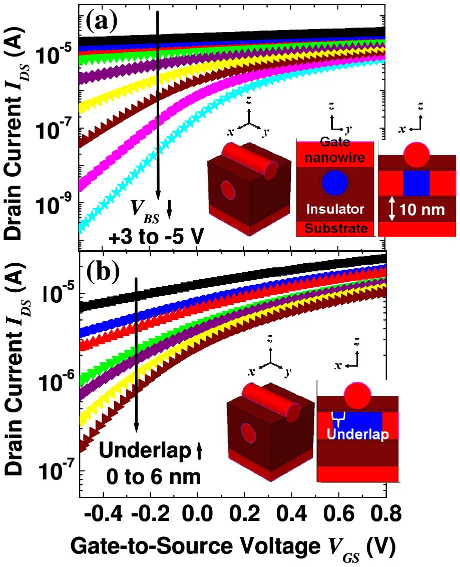


Figure 3. Simulated xnwFET transfer characteristics with different (a) substrate bias, and (b) source-drain junction underlap distance.

A marginal improvement in xnwFET I_{ON}/I_{OFF} can be obtained using a very high channel doping concentration of 10^{19} dopants/cm^3 as shown in Figure 2(c). This heavy doping unfortunately leads to a more pronounced random dopant fluctuation issue [18] since only 8-9 dopant atoms would be present in the channel region. Since there resides no dopant atom inside the channel at concentration less than 10^{18} dopants/cm^3, the theoretical concentration is thus "quantized" to be either 0 or above 10^{18} dopants/cm^3 which in turn constraints the device V_{TH} tunability.

In summary, neither the individual nor combined engineering of the above nanowire properties can secure enough gate-to-channel coupling for a sufficient xnwFET I_{ON}/I_{OFF} and V_{TH}. Device-level measures should therefore be considered to meet the NASIC circuit requirements.

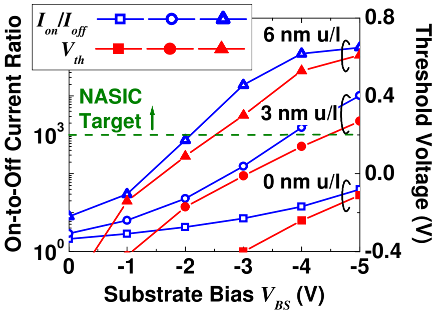


Figure 4. I_{ON}/I_{OFF} and V_{TH} optimization with substrate bias and junction underlap.

2.2.2 xnwFET Device-Level Design Approaches

To be employed in cascading multiple logic circuits, the xnwFETs are required to possess enhancement mode behavior with a positive V_{TH}. The key device design parameters are I_{ON}/I_{OFF} ratio and V_{TH}. Since nanowire-level measures (doping, shape, and geometry) do not satisfy the requirements, device-level approaches should be considered. The two approaches chosen are substrate biasing and source-drain junction underlapping.

A substrate potential bias is applied through the bottom insulator from the wafer backside. As shown in Figure 3(a), a more negative substrate bias (V_{BS}) increases both I_{ON}/I_{OFF} and V_{TH} of the xnwFET. This can be attributed to an effective channel doping concentration increase electrostatically induced by the substrate bias. The resultant diminishing of short-channel effects simultaneously improves both key design parameters and the overall device integrity.

As depicted in the inset of Figure 3(b), an underlap region between the gate nanowire edge and the source-drain junction can also be introduced by applying standard spacer technology. In doing so, the effective channel length can be larger than the gate nanowire diameter (or gate length) such that an enhanced channel length-to-thickness ratio becomes apparent. Consequently, diminished short-channel effects are similarly obtained to improve both the I_{ON}/I_{OFF} and V_{TH} metrics as confirmed in Figure 3(b).

This work is currently ongoing yet the combination of both approaches has revealed great promises (Figure 4). It must be noted that these techniques do not introduce any new manufacturing constraints/challenges. A more detailed discussion on integrated device-circuit explorations validating NASIC devices and circuit styles will be presented in the next section.

To summarize, in this section we provided an overview of semiconductor nanowires and xnwFETs. The next section addresses how these basic building blocks are organized into functional networks and circuits for electronics applications.

3. NASIC Circuit Styles

An important challenge for a circuit designer at the nanoscale is to identify/invent suitable circuit styles that are feasible with nanoscale devices and the capabilities of non-conventional manufacturing. This is a specific instance of the integrated design approach for nanofabrics where design choices at individual levels are compatible with the fabric as a whole. For example, non-conventional and self-assembly based approaches favor the formation of regular structures, such as parallel arrays and grids with limited ability to customize and interface with these structures. This implies that sizing, arbitrary placement of devices or multilayer complex routing of signals may not be possible given manufacturing restrictions. Consequently a conventional circuit style such as CMOS, which has many desirable circuit characteristics but places stringent requirements on manufacturing, may not be suitable for nanofabrics.

In general, any circuit style for digital nanofabrics must incorporate all of the following:

- **Combinational circuits** for implementing arbitrary logic functions
- **Sequential circuits/Latching** for temporary storage of signals
- A **clocking mechanism** for coordinating the flow of data and synchronization with external interfaces
- **Cascading** of circuits and **signal restoration** for propagation of data through a large scale design
- Sufficient **noise margins** to ensure correct functionality.

Furthermore, some circuit characteristics that are desirable for robust system design include:

- **Rail-to-rail voltage swing:** Logic ‘1’s should be at a voltage level of VDD and logic ‘0’s at VSS. This corresponds to the maximum noise margin available.
- No direct paths between VDD and VSS implying **no static power dissipation**.

- **High input impedance:** There is very little leakage from input to output nodes.
- **Low output impedance:** Output nodes are tied to power supplies through networks of devices and are not floating.

In this section we discuss circuit styles that have been developed for the NASIC fabric including static and dynamic styles as well as the pros and cons of each. These circuits are realized on regular arrays and grids of parallel and perpendicular nanowires, in keeping with the aforementioned restrictions of non-conventional manufacturing. We also present an integrated device circuit exploration that uses detailed 3D physics based models of xnwFET devices to validate NASIC circuit styles. It must be noted that circuit style is perhaps the most important factor in determining some key system level metrics such as area, operating frequencies, power and energy of the design.

3.1 NASICs with Static Ratioed Logic

Initial NASIC designs employed static ratioed logic and assumed the availability of reconfigurable devices. A typical example of a preliminary NASIC combinational circuit is shown in Figure 5. It consists of a regular 2-D nanowire grid with Field-Effect Transistors at certain crosspoints. Thick lines along the periphery are microwires carrying VDD (yellow), VSS (green) and signals for programming crosspoints (red and blue). Thin red lines are *p*-type nanowires and thin blue lines are *n*-type. PFET channels are aligned along the horizontal direction and NFET channels are along the vertical. Pull-up and Pull-down arrays imprinted on the grid serve as resistive loads to achieve ratioed logic. This functional network of nanowires, and associated peripheral microwire circuitry is called a **tile**.

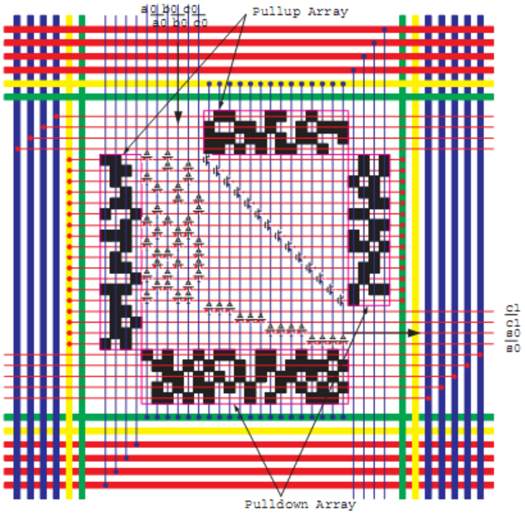


Figure 5. Initial NASIC Design of a 1-bit full adder with static ratioed logic and reconfigurable devices.

In Figure 5, inputs to the tile are received from the top from a previous tile or an external interface. These gate *p*-type FETs. If all inputs are '0', the output goes to '1', accomplishing NOR functionality. The outputs of this stage gate a diagonal arrangement of *n*-type FETs; this is a routing stage that 'turns the corner' and propagates the signals to the next set of *p*-type FETs. Outputs from the tile (in this case carry and sum

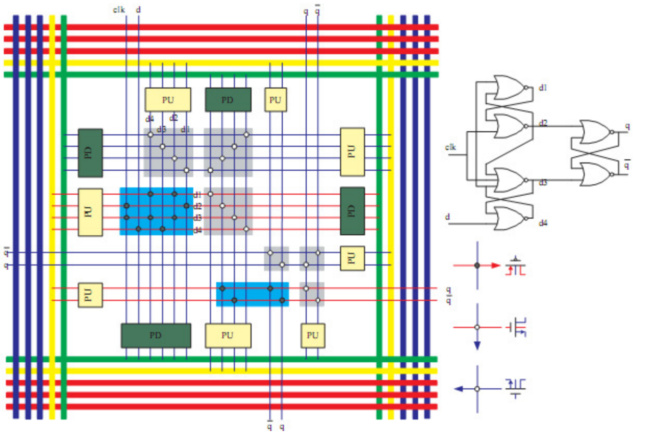


Figure 6. 1-bit flip-flop using NASIC Static Ratioed Logic.

signals) may then be propagated to other tiles or to external interfaces.

Figure 6 shows a sequential circuit (1-bit flip-flop) implemented on a NASIC tile using static ratioed logic with NOR gates and feedback paths. One key observation from Figure 6 is that implementing feedback paths and extensive routing on the grid can lead to very poor fabric utilization; incorporating sequential circuits on the grid can thus cause severe area penalties. Therefore, other mechanisms for latching signals may need to be explored.

Static ratioed logic suffers from key design issues that may make it unsuitable for large scale system design. Firstly, careful sizing of resistive loads is required for correct operation. This may be difficult given constraints of nano-manufacturing. Large static currents can occur due to direct paths between VDD and VSS. The voltage swing is not rail-to-rail, leading to smaller noise margins. Furthermore, implementing sequential circuits on NASIC tiles was shown to be very area inefficient. In the next section we explore dynamic circuit styles for the NASIC fabric that overcome all of the above issues.

3.2 NASIC Dynamic Circuits and Timing Schemes

To overcome significant challenges associated with static ratioed logic, dynamic circuit styles were developed for the NASIC fabric. Figure 7 shows a simple NASIC dynamic circuit implemented on a single semiconductor nanowire with xnwFETs. xnwFET source-channel-drain regions are along the length of the nanowire and gates are on perpendicular nanowires. Precharge (*pre*) and evaluate (*eva*) transistors at the top and bottom of the nanowires are used for implementing dynamic logic. The gates for these control transistors are typically microscale wires driven from external reliable CMOS circuitry. The material used for microwires can be tailored based on design requirement: e.g. precharge transistors could be depletion mode to achieve rail-to-rail voltage swing at the output node.

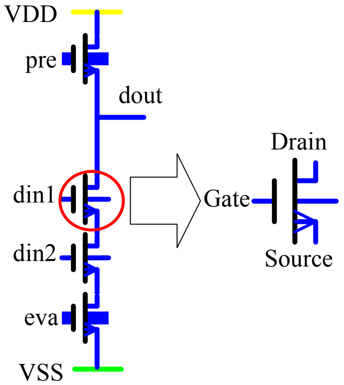


Figure 7. NASIC Dynamic Circuit on a single semiconductor nanowire.

The output node is precharged to logic ‘1’ by asserting the *pre* signal. Subsequently, the *eva* signal is asserted (and *pre* de-asserted) to evaluate the outputs. In Figure 7, if all inputs are at logic ‘1’, the output node will be discharged to logic ‘0’ implementing NAND functionality. Other logic functions such as AND, OR and NOR are also possible by suitably altering CMOS control and xnwFET doping type. However, typically NASICs use only NAND and AND functions in the design to implement arbitrary logic, since these require only *n*-type xnwFETs. Elimination of *p*-type FETs is advantageous because it considerably eases manufacturing requirements and leads to significant performance improvements, since *n*-type FETs are typically faster [19].

The dynamic circuit style has many advantages over static ratioed logic. Dynamic circuits are *ratioless*; therefore sizing of resistive loads and Pull-up/Pull-down arrays are not required. Furthermore, since *pre* and *eva* signals are never asserted simultaneously, static power dissipation does not occur (Leakage power may be consumed due to device leakage mechanisms). Rail-to-rail voltage swing may also be achieved by carefully engineering the microwire materials and suitably altering the work function of *pre* and *eva* FETs.

NASIC dynamic style circuits are different from static/dynamic circuits implemented with CMOS – control schemes will be discussed further; the NASIC fabric also incorporates fault tolerance at circuit level and the control schemes can be engineered to allow more flexibility in the design.

3.2.1 Sequential Circuits Using Dynamic Circuit Styles

Implementing explicit sequential components such as latches and flip-flops on regular 2-D nanowire grids is highly area inefficient given issues with routing signals and implementing feedback mechanisms. NASIC dynamic circuit styles therefore use *implicit* latching of signals on the nanowires themselves without the need for sequential components. This is enabled by using unique control schemes generated from external CMOS.

Figure 8 highlights this concept. It shows a two stage circuit, with outputs of dynamic circuits in Stage 1 acting as inputs for the Stage 2 dynamic circuit. The associated timing diagram shows data and control signals. First, Stage 1 is precharged and evaluated using *pre1* and *eva1* control signals, generating outputs *do1a* and *do1b*. Separate control signals *pre2* and *eva2* are then asserted to evaluate Stage 2, with *do1a* and *do1b* as inputs. During this time, Stage 1 is in a *hold* phase (with *pre1* and *eva1* de-asserted), and *do1a* and *do1b* are *implicitly latched* to their evaluated values on the nanowires. The *hold* phase thus enables correct cascading of data across dynamic stages. Multiple NASIC dynamic stages can be cascaded together to achieve large scale designs such as the NASIC WISP-0 nanoprocessor, with external control schemes to achieve sequential flow of data and pipelining[4].

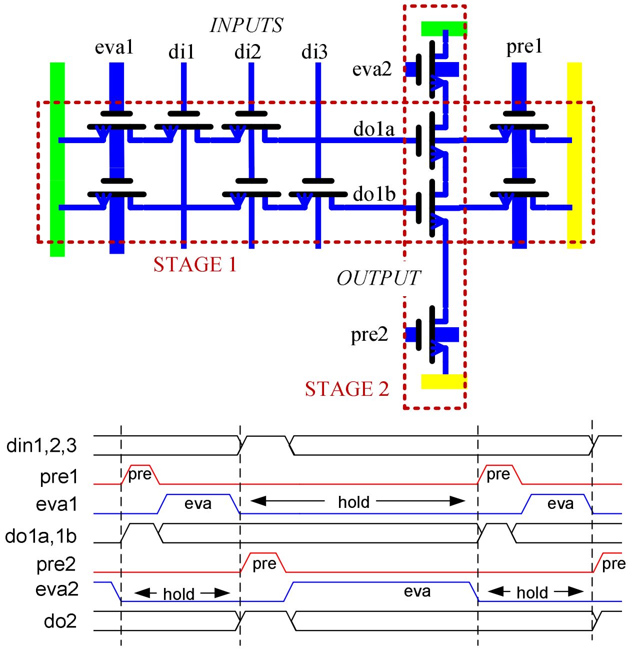


Figure 8. ***Top:*** **NASIC circuit with 2 dynamic stages – Output of one stage is cascaded to the next.** ***Bottom:*** **Control scheme for implicit latching – Successive stages are clocked using different precharge and evaluate signals.**

[4] The use of separate precharge and evaluate signals for successive stages also implies that signal monotonicity issues prevalent in conventional dynamic circuits (that typically require either domino logic i.e. insertion of a static inverter between dynamic stages or np-CMOS type circuit styles) do not affect NASIC dynamic circuits.

Figure 9 shows an example a 1-bit full adder implemented in two stages using NASIC dynamic circuit styles and control. Inputs are received on vertical nanowires, which are then NAND-ed together to generate midterms using *pre1* and *eva1* signals. The xnwFET channels in this stage are along horizontal nanowires, with vertical input gates. The midterms act as inputs for the second NAND stage (whose channels are on the next set of vertical nanowires), and carry and sum outputs are generated along with their complements on vertical nanowires. Subsequent NASIC tiles could then receive these signals as their inputs, achieving large scale cascaded designs.

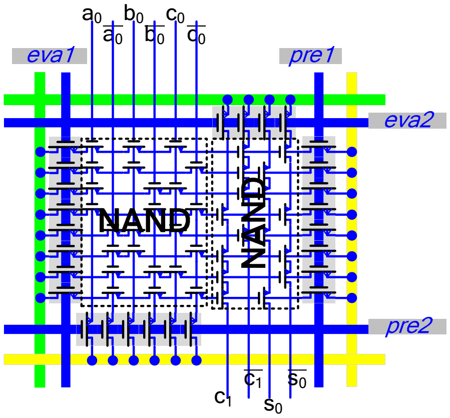


Figure 9. NASIC 1-bit full adder using two cascaded dynamic NAND stages.

3.3 Integrated Device-Circuit Exploration

Nanoscale computing fabrics face challenges at all design levels including manufacturing, devices, circuits, architecture and fault-tolerance. Therefore, design choices at individual levels must be compatible with the fabric as a whole. For example, in addition to having the requisite electrical characteristics, nanodevices should i) meet circuit requirements and function as expected and ii) not require extensive customization that poses insurmountable challenges to non-conventional manufacturing processes.

In this section we explore devices and circuits for NASICs in a tightly integrated fashion, with simulations at the circuit level built on accurate 3-D device simulations of the electrostatics and operations. Using an integrated approach, co-design of devices and circuits can be accomplished with accurate physics based device models. Accurate modeling is especially important for NASIC dynamic circuits with single-type FETs [19], where using only *n*-type devices simplifies manufacturing requirements and improves performance, but may lead to reduced noise margins.

We present the methodology for integrated device-circuit exploration [39], investigate different devices and show how the optimal choice of device parameters such as V_{TH}, I_{ON}/I_{OFF} ratios etc. enables correctly functioning cascaded dynamic circuits. Importantly, we also discuss how these device level characteristics can be achieved without unrealistic customization requirements on the manufacturing process.

3.3.1 Methodology

Figure 10 summarizes the methodology used for integrated device-circuit explorations. Device level characterization of xnwFET structures is done using 3D simulations on the Synopsys Sentaurus Device^{TM} simulator [15]. Drain current vs. gate voltage (transfer) characteristics are obtained for drain voltages varying between 0.01 V to 1.0 V, which covers the operating range of the devices in the NASIC dynamic circuits. Gate-source and gate-drain capacitances are also extracted as a function of the gate voltage. Regression analysis is carried out on the drain current data, and multivariate polynomial fits (for FET on-region behavior) and exponential fits (for off-region behavior) are extracted using DataFit software[5]. These relationships express the drain current as a function of two independent variables, gate-source (V_{GS}) and drain-source (V_{DS}) voltages. These fits are then incorporated into subcircuit definitions for voltage-controlled resistors in HSPICE [20]. Capacitance data from Sentaurus is directly integrated into HSPICE using voltage-controlled capacitance (VCCAP) elements and a piece-wise linear approximation since only small variations in capacitance were observed with V_{GS} increments. The regression fits for currents together with the piece-wise linear model for capacitances and subcircuit interconnections define the behavioral model for xnwFET devices used in circuit simulation.

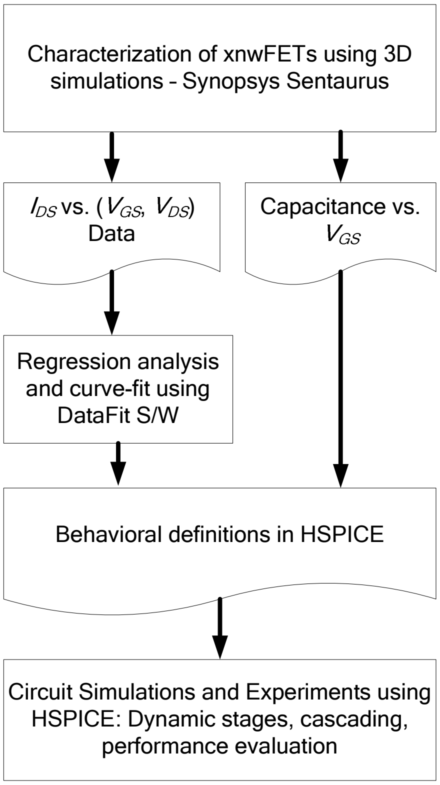

Characterization of xnwFETs using 3D simulations - Synopsys Sentaurus
I_{DS} vs. (V_{GS}, V_{DS}) Data
Capacitance vs. V_{GS}
Regression analysis and curve-fit using DataFit S/W
Behavioral definitions in HSPICE
Circuit Simulations and Experiments using HSPICE: Dynamic stages, cascading, performance evaluation

Figure 10. Methodology for integrated device-circuit explorations.

Once behavioral data has been incorporated into HSPICE, multiple experiments are conducted including: DC sweeps of individual devices to verify behavioral models, simulation of a single dynamic stage to verify functionality, and simulation of multiple stages to evaluate the effects of nanowire cascading, charge sharing and diminished noise margins. Devices can also be optimized for circuit-level behavior targeting key metrics such as performance and power consumption.

Table I. xnwFET Devices Explored

Device	Gate and channel NWs	N_{ch}	$N_{G/S/D/Sub}$	Ulap	V_{sub}
#1	10×10 nm^2	10^{19} cm^{-3}	10^{20} cm^{-3}	0 nm	0V
#2				7 nm	0V
#3				7 nm	-1V

[5] A Note on Regression-based vs. Analytical Modeling: A regression based approach is very generic and can be used to fit arbitrary device characteristics. Coefficients extracted from regression data fits are representative of the device behavior over sweeps of drain-source and gate-source voltages. This is in contrast to conventional in-built models in SPICE for MOSFETs and other devices, which use analytical equations derived from theory and physical parameters such as channel length and width. The regression coefficients in our approach may not directly correspond to conventional physical parameters. Therefore different regression fits will need to be extracted for devices with varying geometries, doping etc.

As an example we show 3 devices that were investigated using this approach. The device parameters are shown in Table I. By modifying the substrate bias voltage and the underlap, key device parameters such as the V_{TH} and I_{ON}/I_{OFF} ratios were modulated. Current-voltage (I-V) characteristics for the devices are shown in Figure 11. Device #1, without any underlap or substrate bias has poor electrostatic control. Device #2, incorporating underlap has $I_{ON}/I_{OFF} > 10^3$, but the V_{TH} is small. Finally, Device #3, with both underlap and substrate bias meets the requirements for correct NASIC functionality.

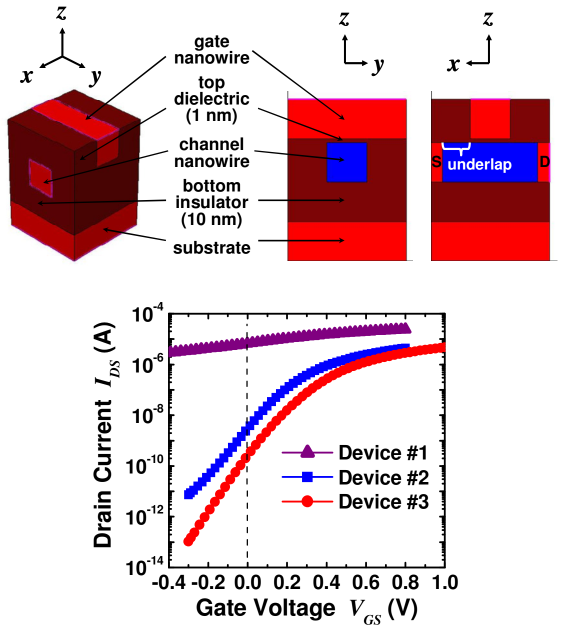


Figure 11. Simulated transfer characteristics of xnwFETs with parameters listed in Table I.

Using the curve-fit models of the device characteristics, multiple test simulations may be done in HSPICE. DC sweeps of the models verifies correct abstraction of device data. Simulations of single NASIC NAND stages using Devices #2 and #3 were shown to function correctly.

For evaluation of dynamic circuit cascading and noise margin issues, a simple 3-stage test circuit was used (Figure 12). In this circuit, Stage 1 generates imperfect outputs and signal integrity was checked at the outputs of Stages 2 and 3. Stage 3 has a single input device and represents a worst-case noise scenario; this stage has the least possible effective resistance and capacitance between the output and VSS nodes. Therefore, it is susceptible to noise on its input node. Separate precharge and evaluate signals were used for the three stages in accordance with NASIC dynamic control concepts.

Figure 13.A and B show the results of circuit simulation using Device #2 and Device #3 respectively. From Figure 13.A, we see that a glitch (due to charge sharing effects) at the output of Stage 2 causes loss of signal integrity at the output of Stage 3. This is because the magnitude of the glitch is larger than the threshold voltage for Device #2, causing the input transistor to operate in the linear region. However, in Figure 13.B, the glitch does not affect the Stage 3 output, since its magnitude is smaller than Device #3 V_{TH}, which ensures that the input transistor stays switched off. Thus, in this example, the integrated device-circuit exploration helps identify and evaluate device characteristics for correct cascading and circuit functionality. Additional experiments such as detailed evaluation and optimization of key system level metrics such as performance and power may be carried out using the behavioral models and HSPICE simulations.

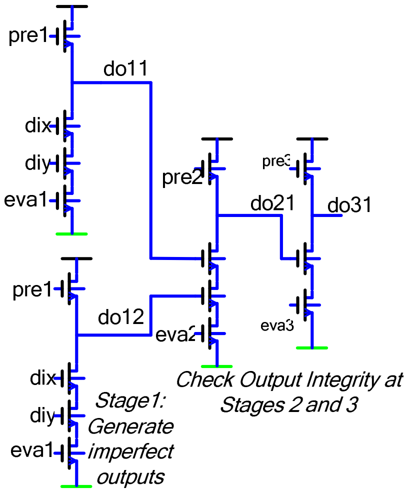

pre1
do11
dix
diy
eva1
pre2
pre3
do21
do31
pre1
do12
eva2
eva3
dix
diy
eva1
Stage1: Generate imperfect outputs
Check Output Integrity at Stages 2 and 3

Figure 12. Circuit for validation of cascading in dynamic NASIC design.

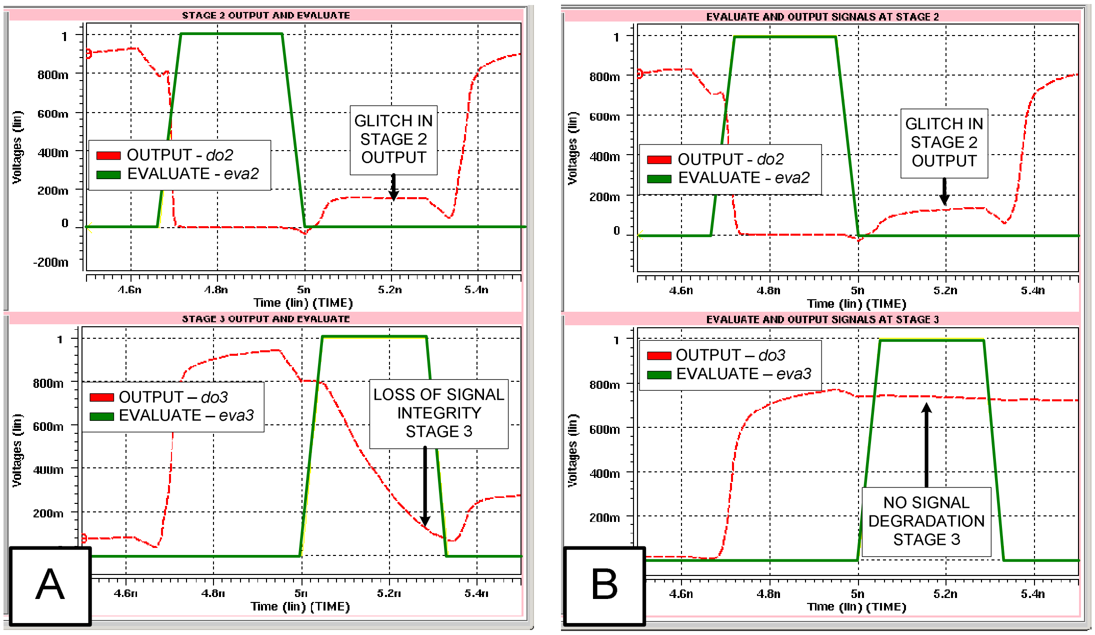


Figure 13. Results of Circuit Simulations with xnwFET devices. A) Device #2 with low VTH is susceptible to noise, B) Device #3 does is not affected by glitch

A Note on Manufacturing Implications: Manufacturing of nanodevice based computational systems continues to be very challenging. Therefore, while devices should possess the requisite characteristics to meet circuit requirements and expected functionality, it is equally important that they can be integrated in a manufacturing process without introducing new challenges. For example, while large gate to channel ratios (e.g. 20×20 nm^2 gate, 10×10 nm^2 channel) can achieve the required electrostatic control and device characteristics including V_{TH} and current ratio, the inherent problem with these is the dissimilar gate and channel dimensions. Since the output node of one nanowire acts as gate for the next stage, using 20/10 devices would need asymmetry along the length of the nanowire. This would require varying the radius during growth of nanowires themselves or contacting nanowires of different diameters together after transferring to a substrate. Nanowires with identical gate and channel dimensions do not have these issues, but may suffer from poor electrostatics as shown in Section 2. We therefore tune electrical characteristics using techniques such as underlap and substrate biasing. V_{TH} tuning using these techniques does not impose any new manufacturing challenges. Biasing in NASIC designs is done for the entire circuit, which is much simpler than biasing individual devices. Spacer requirements for underlap are expected to be comparable to end-of-the-line CMOS technologies.

3.3.2 Control Schemes for the NASIC Fabric

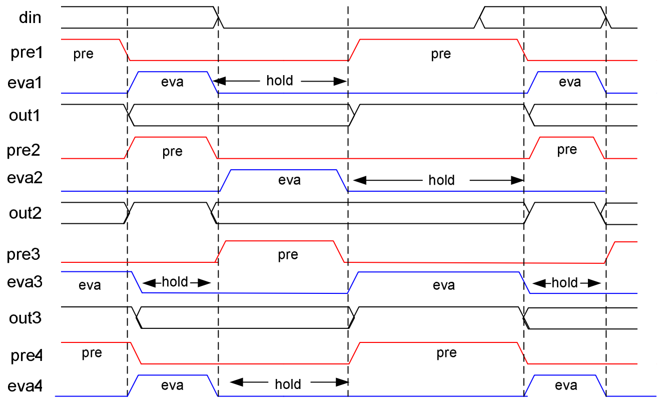


Figure 14. High performance 3-phase control scheme for NASIC Designs

One possible control scheme for the NASIC fabric was shown in Figure 8. Control schemes may be optimized for higher performance and noise resilience. This does not have any associated cost at the nanoscale in terms of manufacturability since the control schemes are generated by reliable external CMOS circuitry. Figure 14 and Figure 15 show two possible control schemes for the NASIC fabric.

The control scheme in Figure 14 improves the performance of NASIC designs. In this scheme, *pre2* is overlapped with *eva1*, and *pre3* is overlapped with *eva2* leading to a 3-phase clock (there are 3 separate precharge and evaluate signals that repeat every 3 stages). By overlapping the phases, performance of the design can be significantly improved. For example, assuming that the phases in Figure 8 are of approximately the same duration, up to 33% improvement in performance can be achieved. However there are two key issues with this scheme:

i) Feedback circuits may not be possible: Consider a 4-stage feedback path; the output of Stage 4 will be the input for Stage 1. This requires *eva4* to complete before *eva1*, which is not the case with the 3 phase control scheme. One possible way to circumvent this issue is to use static portions of the circuits for feedback, however issues previously discussed for static circuits will be applicable.

ii) Noise issues: Consider a 3 Stage circuit similar to Figure 12. While *pre1* is asserted, inputs to Stage 2 go to logic ‘1’, causing charge sharing between Stage 2 output and intermediate nodes. This can cause glitching of the Stage 2 output. Since Stage 3 is evaluating at this time (*eva3* is asserted), the glitch at its input can lead to loss of signal integrity at its output node (this issue is also applicable to the ‘basic’ control scheme shown in Figure 8).

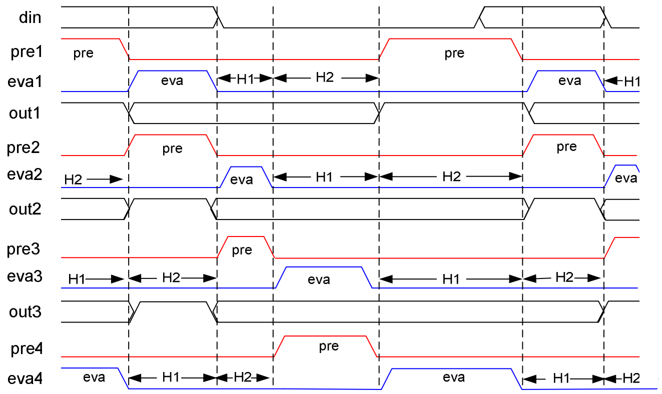


Figure 15. 4-phase noise resilient control scheme for NASIC

To overcome issues with a 3-phase clocking scheme, a 4-phase scheme may be constructed (Figure 15). In this scheme, in addition to *pre* and *eva*, each dynamic stage goes through two hold phases (labeled *H1* and *H2*). During *H1*, the output is held constant while the next stage is evaluated, similar to the *hold* phase in previous schemes. However, before *pre* is asserted again, an additional *H2* phase is inserted. This ensures that glitches due to precharging do not propagate to subsequent stages. For example, we see that when *eva3* is asserted, Stage 1 is in *H2* and there is no switching activity in this stage. This implies that there is no glitching at the output of Stage 2, whose outputs will be correctly evaluated. When *pre1* is asserted, glitching can occur at Stage 2 output. However by this time Stage 3 has completed evaluation (*eva3* is de-asserted) and is not affected by the glitch.

The 4-phase clock also implies that feedback paths can be correctly implemented. Since there are 4 phases and 4 unique *pre* and *eva* signals, *eva4* completes just ahead of *eva1* meeting the requirement for feedback.

To summarize, this section dealt extensively with possible NASIC circuit styles and associated control schemes. Dynamic circuits with single-type FETs in conjunction with unique control schemes generated from CMOS are suitable for a regular 2-D grid based computational fabric with limited customization requirements. These circuit styles achieve:

- **Combinational Circuits:** NAND gate implementations, which are universal, were shown.
- **Sequential Elements:** Implicit latching on nanowires is achieved by using unique control schemes where successive stages use different *eva* signals.
- **Clocking Mechanism:** The flow of data along multiple NASIC stages is achieved by the control scheme, enabling pipelined designs.
- **Sufficient Noise Margins:** An integrated device-circuit co-design approach was shown, and techniques to tune devices to meet circuit requirements and noise margins were discussed.

Additionally, these dynamic circuits were shown to have **rail-to-rail voltage swings, no static power dissipation** and **high input impedance**. At the time of writing, new control schemes are under investigation to achieve low output impedance that could lead to designs with even better noise resilience.

4. NASIC Logic Styles

NASIC designs typically follow 2-level logic styles such as NAND–NAND. 2-level logic lends itself to relatively straightforward implementation on regular 2-D nanowire grids. Figure 16 shows different implementations of a 1-bit full adder on the NASIC fabric. Figure 16.A uses an AND–OR logic implementation with *n*-type xnwFETs (white boxes) implementing AND logic on the horizontal plane and *p*-type FETs (black boxes) implementing OR logic on the vertical plane. An improvement over the AND-OR based logic implementation is the NAND-NAND implementation (Figure 16.B), using *n*-type FETs only. The migration is achieved by suitably altering the external dynamic control scheme. The single-type FET based NAND–NAND scheme eases requirements on manufacturing and significantly improves the performance of NASIC designs by eliminating the slower *p*-FET devices without any deterioration in density [19].

Simple 2-level logic schemes have some inefficiency in terms of generating midterms for complementary signals separately from true values, as well as propagating all complements. Figure 16.C shows a Heterogeneous 2-level (H2L) logic scheme that provides a more efficient logic implementation than simple 2-level logic. H2L combines different logic functions into a single stage, e.g. the vertical logic gates in Figure 16.C implement both NAND and AND logic functions. This is notated as NAND–NAND/AND H2L logic. The availability of multiple logic functions in a single NASIC stage implies efficient logic implementation and higher density compared to simple NAND–NAND schemes. H2L based NASIC processors have been shown to be up to 30% denser than their NAND–NAND based counterparts [21]. H2L logic also significantly improves the yield of NASIC designs. This is due to two factors:

- Simplifying the design – fewer devices are used to implement the same logic function
- Ability to do nanoscale voting – H2L can be used to design high-yielding NASIC based majority voting blocks.

H2L is enabled by modifying control circuitry and nanowire connections to VDD and VSS. Consequently, H2L schemes for the NASIC fabric do not impose any new challenges from a manufacturability perspective. It must be noted that H2L is a generic logic style that can be applicable to other nanoscale fabrics employing 2-level logic schemes. Ongoing work on enhancements to H2L includes 3 stage, 2-level NASIC design with tiles consisting of one input and 2 separate output stages generating outputs in a time multiplexed fashion improving area

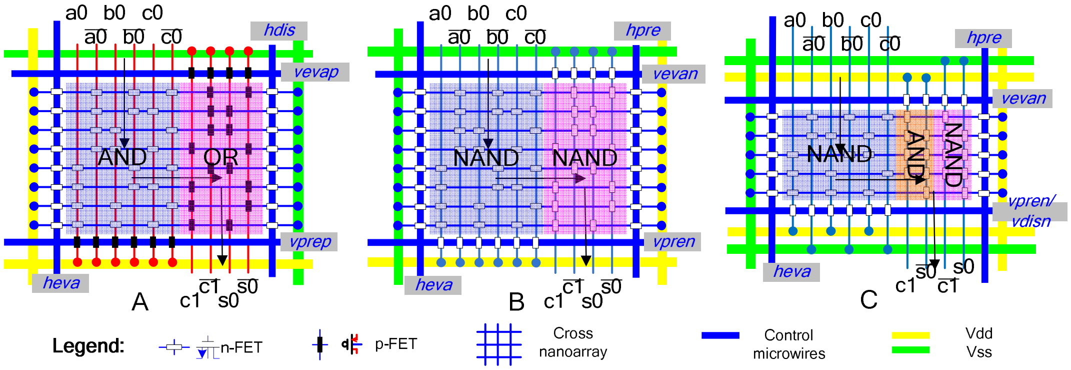

a0 b0 c0
$\overline{a0}$ $\overline{b0}$ $\overline{c0}$
hdis
vevap
AND
OR
vprep
heva
A
c1 $\overline{c1}$ s0 $\overline{s0}$
hpre
vevan
NAND
NAND
vpren
heva
B
c1 $\overline{c1}$ s0 $\overline{s0}$
hpre
vevan
NAND
AND
NAND
vpren/
vdisn
heva
C
c1 $\overline{s0}$ s0 $\overline{c1}$
Legend:
n-FET
p-FET
Cross nanoarray
Control microwires
Vdd
Vss

Figure 16. NASIC Logic styles A) AND-OR with 2 types of FETs B) NAND-NAND with *n*-FETs only xnwFETs and C) Heterogeneous 2-level (H2L) logic

utilization and performance.

5. NASIC Architectures

Large scale computing systems may be designed using the NASIC fabric and the associated framework of building blocks, xnwFET based circuits, and logic styles discussed in the preceding sections. This section discusses two key architectures for the NASIC fabric: the WISP-0 general purpose processor and a massively parallel architectural framework with programmable templates for image processing.

5.1 WIre Streaming Processor (WISP-0)

WISP-0 is a stream processor that implements a 5-stage microprocessor pipeline architecture including *fetch, decode, register file, execute* and *write back* stages. WISP-0 consists of five nanotiles: Program Counter (PC), ROM, Decoder (DEC), Register File (RF) and Arithmetic Logic Unit (ALU). Figure 17 shows its layout. A nanotile is shown as a box surrounded by dashed lines in the figure. In WISP designs, in order to preserve the density advantages of nanodevices, data is streamed through the fabric with minimal control/feedback paths. All hazards are exposed to the compiler. It uses dynamic circuits and pipelining on the wires to eliminate the need for explicit flip-flops and therefore improve the density considerably. WISP-0 supports a simple instruction set including *nop, mov, movi, add* and *multiply* functions. It uses a 7-bit instruction format with 3-bit instruction and 2-bit source and destination addresses. The WISP-0 is used as a design prototype for evaluating key metrics such as area and performance as well as the impact of various fault-tolerance techniques on chip yield and process variation mitigation. Additional enhancements to this design are ongoing in the NASIC group.

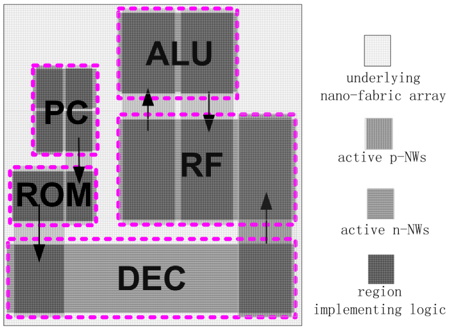


Figure 17. WISP-0 Processor Floorplan

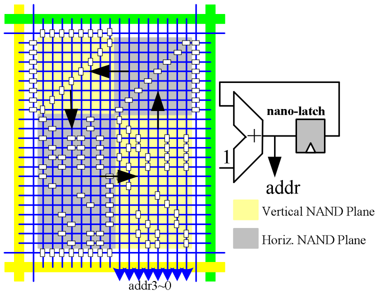


Figure 18. WISP-0 Program Counter

5.1.1 WISP-0 Program Counter

The WISP-0 program counter is implemented as a four bit accumulator. Its output is a four bit address that acts as an input to the ROM. The address is incremented each cycle and fed back using a nano-latch. Figure 18 shows the implementation of the Program Counter with a NAND-NAND scheme. Diagonal transistors on the upper two NAND planes implement the nano-latch to delay the output by one cycle and allow the signals to ‘turn the corner’.

5.1.2 WISP-0 ALU

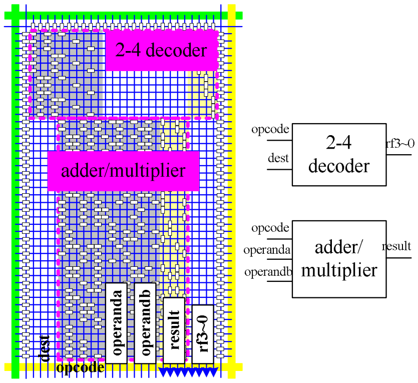


Figure 19. WISP-0 ALU

Figure 19 shows the layout and schematic of the WISP-0 ALU that implements both addition and multiplication functions in a single stage. The arithmetic unit integrates an adder and multiplier together to save area and ease routing constraints. It takes the inputs (at the bottom) from the register file and produces the write-back result. At the same time, the write-back address is decoded by the 2-4 decoder on the top and transmitted to the register file along with the result. The result will be written to the corresponding register in the next cycle.

More information about WISP-0 can be found in [22][23].

5.2 NAnodevice-based Programmable Architectures (NAPA)

The processor architecture reviewed in the previous section is composed of dissimilar logic blocks performing a variety of functions interconnected in a prescribed fashion to build nanoscale systems. An alternative architectural style is to use a massively parallel array of identical logic blocks communicating with their neighbors to achieve data processing. Nanodevice-based Programmable Architectures (NAPA) is an architectural framework for the NASIC fabric using these sorts of collective computation models.

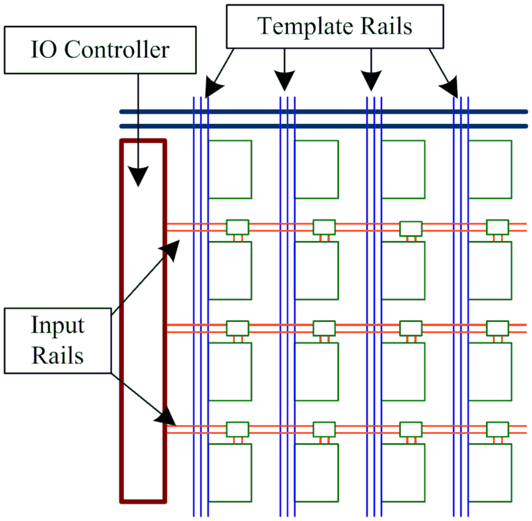


Figure 20. The NAPA Architectural Framework

The NAPA design (Figure 20) is regular and highly parallelized, with a large number of identical functional units or cells performing a given task. Instructions to each cell are transmitted on a limited number of global signals (template rails) controlled from reliable peripheral CMOS circuitry. There is a CMOS Input-Output Controller that serially loads the inputs and reads out outputs. The cells themselves are composed of a small number of nanodevices that perform simple computations and are locally interconnected. The collective behavior of a large number of interconnected cells achieves information processing.

These sorts of collective computation systems may be a 'natural fit' for nanoscale implementations because i) high densities are available for high-level of parallelism, ii) local interconnections with minimal global routing remove the need for complex interconnect structures and iii) 100% fault free components are not a requirement since some defective cells in the design may not adversely affect the overall output integrity and allow more aggressive manufacturing processes. However, given the high rates of defects in nanomanufacturing, built-in defect/fault tolerance techniques will need to be incorporated at various levels of NAPA to ensure that a sufficiently large number of cells are functioning correctly for output integrity.

Different applications that use cellular architectures such as Single Instruction Multiple Data (SIMD), Digital Signal Processing (DSP), and Cellular Nonlinear Networks (CNN) may be mapped to a NAPA design. The architecture and capabilities provided in a single cell determine the application, while the framework for programmability and input-output operations is unchanged. We show one possible implementation based on a computational model used in discrete-time digital Cellular Neural Network (CNN) [24][25] for image processing tasks. Multiple image processing tasks are possible by controlling the global instruction signals, which act as the templates for the CNN cells. Hence, the design is fully programmable and can run a variety of processing tasks in a sequence.

In a digital CNN design, each cell processes one pixel of an image (the input) and has an associated multi-bit state. A multi-bit template determines the nature of the image processing task. The state evolves as a function of the inputs, the template and signals exchanged with neighbor cells. The output is simply the MSB of the state variable. The state evolution is given by the relation:

$$x(n+1) = \sum_{i \in N} a_i\, y_i(n) + b_i\, u_i + C$$

where each a_i and b_i are instantaneous template values, $y_i(n)$ and u_i are the outputs and inputs to a particular cell, $x(n+1)$ is the state and C is a constant term. The term $a_i\, y_i(n) + b_i\, u_i$ represents the i^{th} partial sum received from a neighbor. The templates are space invariant, i.e. the same template values are transmitted to all cells in the design.

Figure 21 shows the layout and circuit level implementation of a single cell. There is a NASIC tile at the center for generating partial sums and state accumulator tiles at the corners of each cell. The execution inside a cell progresses through the following stages:

1. **Generation of Partial Sums:** Initially, the control circuitry for the state accumulators is off and the partial sum generator is on. Each cell then generates 5 partial sums in sequence, one per nearest neighbor, and one for itself. These are transmitted on the broadcast network (blue clockwise loop).
2. **State Accumulation:** Once all partial sums are available, state accumulation datapaths are activated (anti-clockwise loop). This is enabled by switching off the control circuitry for the partial sum and switching on the control for the state accumulator tiles. During this stage, the broadcast networks may be envisioned to be in an extended hold phase, with no precharge-evaluate operations. Thus, by suitably triggering the dynamic control circuitry, selective activation of datapaths and propagation of signals is achieved on a NASIC fabric without the need for arbitrary routing or additional multiplexers.
3. **Output Calculation:** After all partial sums from neighbors have been accumulated, the new value of the output is calculated and the next partial sum generation cycle will begin. After a specified number of iterations, the final output may be readout using the IO controller.

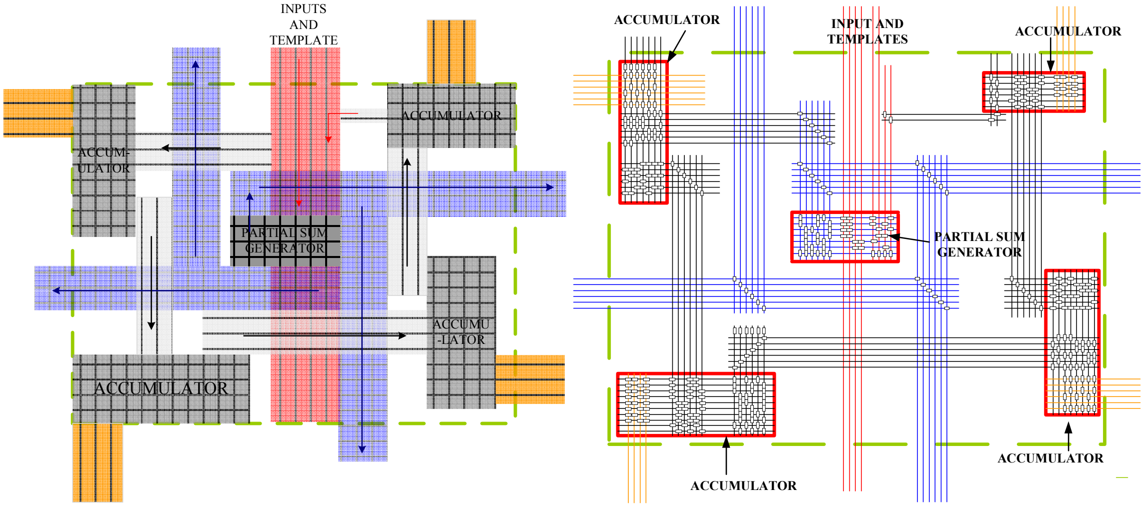


Figure 21. Floorplan (left) and Circuit Diagram (right) of NAPA Cell Implementing Digital CNN.

More information on cellular architectures for NASIC can be found in [26][27].

6. Built-in Fault Tolerance

Nanoscale manufacturing is expected to have manufacturing defects at many orders of magnitude higher than conventional CMOS manufacturing processes. Furthermore, the effects of transient faults due to noise, soft errors and parameter variations are expected to be significant. Consequently, nanoscale systems need to incorporate some form of fault tolerance to mitigate these effects and obtain functional chips. In this section we investigate built-in fault-tolerance techniques for the NASIC fabric and evaluate their impact on WISP-0's yield, area and timing. Additionally, WISP-0's density is compared with an equivalent CMOS version developed with state-of-the-art conventional CAD tools and scaled to projected technologies at the end of the ITRS-defined semiconductor roadmap [28].

Two main directions have been proposed to handle defects/faults at nanoscale: reconfiguration and built-in fault tolerance. Most conventional built-in defect/fault-tolerance techniques [29][30][31][32][33][34][35][36][37][38], however, are not suitable in nanoscale designs because they were designed for very small defect rates and assume arbitrary layouts for required circuits. Moreover, the circuits used for fault correction are often assumed to be defect free, which cannot be guaranteed in nanoscale fabrics.

Secondly, if reconfigurable devices are available, defective devices might be replaceable after manufacturing. Reconfiguration based approaches [40][41][42][43], however, include significant technical challenges: (i) highly complex interfaces are required between micro and nano circuits for extracting defect maps and reprogramming around defects - this is considered by many researchers a serious manufacturing challenge due to the alignment requirement of a large number of nanowires with programming microwires (MWs), (ii) special reconfigurable nanodevices are needed requiring unique materials with programmable, reversible and repeatable characteristics, (iii) an accurate defect map has to be extracted through a limited number of pins from a fabric with perhaps orders of magnitude more devices than in conventional designs, and

(iv) reconfiguration based approaches do not protect systems from noise and transient faults due to process variations.

Alternatively, as shown in this section, we can introduce fault tolerance at various granularities, such as fabric, circuit, and architecture levels, to make nanoscale designs functional even in the presence of errors, while carefully managing area tradeoffs. Such built-in fault tolerance could possibly address more than just permanent defects. Faults caused by speed irregularities due to device parameter variations, noise, and other transient errors could be potentially masked. Compared with reconfiguration based approaches, this strategy also simplifies the micro-nano interfacing: no access to every crosspoint in the nanoarray is necessary. Furthermore, a defect map is not needed and the devices used do not have to be reconfigurable. Introducing built-in fault-tolerance does not impose any new constraints on manufacturability.

6.1 Techniques for Masking Manufacturing Defects

Permanent defects are mainly caused by the manufacturing process. The small nanowire dimensions combined with the self-assembly process, driven by the promise of cheaper manufacturing, is expected to contribute to high defect rates in nanoscale designs. Examples of permanent defects in NASIC fabrics would include malfunctioning FET devices, broken nanowires, bridging faults between nanowires, and contact problems between controlling MWs and nanowires. For example, in a process that requires ion implantation of segments connecting NASIC FETs for high conductivity, the channels of transistors could be unintentionally implanted and therefore stuck-on.

In NASICs we consider a fairly generic model with both uniform and clustered defects and three main types of permanent defects: nanowires may be broken, the xnwFETs at the crosspoints may be stuck-on (no active transistor at crosspoint) or stuck-off (channel is switched off). A stuck-off transistor can also be treated as a broken nanowire. The initial thinking is that the more common defect type is due to stuck-on FETs as a consequence of the ion implantation process used. NASIC fabrics require lithographic masking for implantation steps (to avoid channels at crosspoints where no FETs are placed). Thinking from [44] suggests that we will be able to control the reliability of nanowires fairly well so broken nanowires will be likely less frequent than stuck-on FETs.

6.1.1 Circuit-Level and Structural Redundancy

Figure 22 shows a simple example of a NASIC circuit implementing a NAND-NAND logic function with built-in redundancy: redundant copies of nanowires are added and redundant signals are created and logically merged in the logic planes with the regular signals. As shown in the figure, horizontal nanowires are precharged to ‘1’ and then evaluated. Vertical nanowires are subsequently precharged to ‘1’ and then evaluated. The circuit implements the logic function $o_1 = ab+c$; *a’* is the redundant copy of *a* and so on. Signal *a* and *a’* are called a nanowire pair.

A NASIC design is effectively a connected chain of NAND-NAND (or equivalent) logic planes. Our objective is to mask defects/faults either in the logic stage where they occur or following ones. For example, a break on a horizontal nanowire in the NAND plane (see, for example, position "A" in the figure) causes the signal on the nanowire to be '1'. This is because the nanowire is disconnected from VDD. The faulty '1' signal can, however, be masked by the following logic NAND plane if the corresponding duplicated/redundant nanowire is not defective.

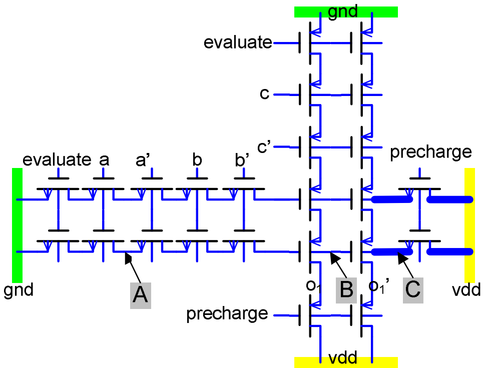


Figure 22. Simple 2-way redundancy

A nanowire break at position "B" can be masked by the NAND plane in the next stage. Similar masking can be achieved for breaks on vertical nanowires. For stuck-on FETs, simple redundancy schemes work well: if one of the two transistors is stuck-on, the redundant copy ensures correct functionality.

6.1.2 Interleaving Nanowires

While the previous technique can mask many types of defects, faults at certain positions are difficult to mask. For example, if there is a break at position "C" in Figure 22, the bottom horizontal nanowire is disconnected from VDD preventing precharge. The signal on this nanowire may potentially retain logic '0' from a previous evaluation. Because of NAND logic on the vertical nanowires, the two vertical nanowires would then be set to logic '1'. Since both outputs on the vertical nanowires are faulty, the error cannot be masked in the next stage. In Figure 22, the thicker segments along the horizontal nanowires show the locations at which faults are difficult to mask. We call these segments *hard-to-mask segments*.

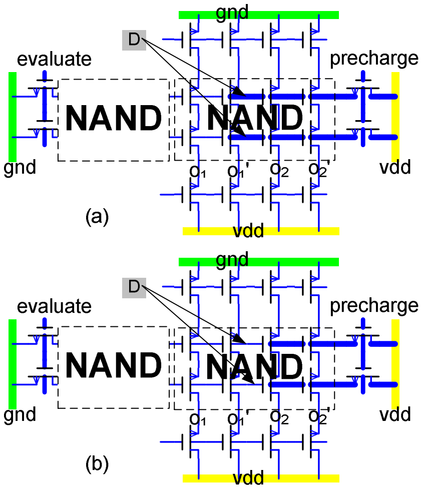


Figure 23. Interleaving to reduce hard-to-mask regions

For nanotiles with multiple outputs, a particular arrangement of output nanowires and their redundant copies could significantly reduce the size of hard-to-mask segments. This is shown in Figure 23: Figure 23(a) presents a design in which each output nanowire and its redundant copy are adjacent to each other. In this arrangement, all segments to the right of the leftmost output nanowire pair (o_1 and o_1*'* in Figure 23(a)) are hard-to-mask.

Alternatively, the interleaved version in Figure 23(b), shows an arrangement in which the output nanowires and their redundant copies are separated into two groups (o_1 and o_2 form one group; o_1*'* and o_2*'* form another group). In this case, the size of the hard-to-mask segments is reduced. In general, the size of hard-to-mask segments can be reduced in larger scale designs to half, i.e., to half of the region covered by the vertical nanowires plus the segment related to the control FET. This latter region is fixed and for most designs adds a negligible area. Interleaving is also helpful in masking clustered defects because duplicated nanowires are set apart from one another.

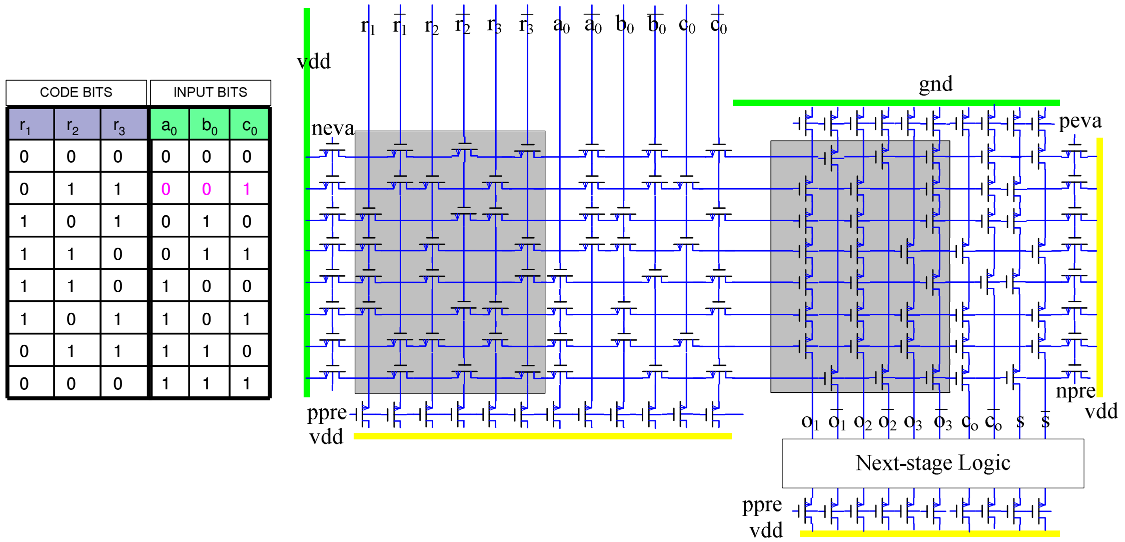
6.1.3 Integrated Code-based Error Masking

A variety of code based techniques exist for correcting errors in conventional systems such as Hamming codes [29] or Residue codes [30]. However, most of these require dedicated fault-free encoding and decoding circuitry and may not be directly applicable to nanoscale fabrics with much higher defect rates since correcting circuitry may also have faults. We explore a new integrated code-based error- masking scheme designed for nanoscale logic systems, where codes are directly merged into the fabric at circuit and logic levels. This scheme is built on the principles of Hamming distance and codes, but is achieved without the need for explicit encoding and decoding components.

The Hamming distance between two input codes of the same length is defined as the number of disparate bits. For example, if we consider two input codes “001” and “101”, the Hamming distance is 1. Another interpretation of the Hamming distance is the minimum number of bits that need to be flipped for one input code to be indistinguishable from another. In communication channels and memory systems, increasing the distance between input codes (using additional bits) implies that faults (flipped bits) may be detected or corrected.

We apply these basic principles to logic circuits in a novel way. The key innovation is that the positions of transistors in an input logic plane of a NASIC tile can be envisioned to be input codes. This concept is highlighted in Figure 24. The unshaded region in the input plane of the NASIC tile shows the original input plane of the NASIC 1-bit full adder. We use the following rule to determine the codes based on FET positions: if there is a FET at the true value of an input signal we designate the input bit to be ‘1’, if there is a FET at the complementary position, we designate as ‘0’. For example, the first horizontal row has FETs at *a’, b’* and *c’*. The corresponding codeword is “000”. Similarly, input codes for all the other horizontal nanowires are assigned. These are summarized in the table on the left.

Once the input table is constructed, the Hamming distance may be increased by adding code bits.



CODE BITS			INPUT BITS		
r_1	r_2	r_3	a_0	b_0	c_0
0	0	0	0	0	0
0	1	1	0	0	1
1	0	1	0	1	0
1	1	0	0	1	1
1	1	0	1	0	0
1	0	1	1	0	1
0	1	1	1	1	0
0	0	0	1	1	1

r_1 $\bar{r}_1$ r_2 $\bar{r}_2$ r_3 $\bar{r}_3$ a_0 $\bar{a}_0$ b_0 $\bar{b}_0$ c_0 $\bar{c}_0$
vdd
neva
gnd
peva
npre
vdd
ppre
vdd
o_1 $\bar{o}_1$ o_2 $\bar{o}_2$ o_3 $\bar{o}_3$ c_o $\bar{c}_o$ s $\bar{s}$
Next-stage Logic
ppre
vdd

Figure 24. Integrated Error Correction in NASIC Designs

For example, a distance of 2 may be achieved with a single parity bit. This example shows one possible code achieving a distance of 3 with 3 additional code bits.

Once the code table has been constructed it may be directly integrated into the logic plane with the positions of xnwFETs on the coding nanowires (left shaded plane) determined by the aforementioned code conversion rule. The coding xnwFETs provide resilience to faults on the input xnwFETs, with at least 3 errors required to erroneously evaluate a horizontal nanowire to ‘0’ (corresponding to the Hamming distance of 3). Any combination of 2 or fewer stuck-on defects or broken nanowires (in either the input or code nanowires) will not be propagated to the next stage.

The Hamming distance of 3 was found to be optimal for WISP-0 designs, as will be shown in the results section. The manufacturing process, defect models and yield-area tradeoffs may determine the choice of distance for a particular design.

6.1.4 Voting at the Nanoscale

Majority voting is a well understood technique for conventional fault tolerance. N independent identical modules generate copies of an output that go to a voting block that determines the correct value of the signal to be a simple majority of the copies. The reliability of the voting scheme (i.e. the probability that a correct output is generated) is determined by the reliability of the individual modules (R_M), the reliability of the voter (R_V), as well as the number of identical copies. This is illustrated in Figure 25, which plots the reliability of the system against the reliability of the individual module. Different curves correspond to different voter reliabilities and schemes including Triple Modulo Redundancy (TMR – voting on 3 identical copies) and 6MR (i.e. voting on 6 copies of the signal). The dotted black line is the break-even point (system reliability = module reliability). Points above the line show regions where voting helps improve the reliability of the system.

Three thin solid lines represent the reliabilities of TMR outputs and three thick lines for the reliabilities of 6MR outputs. From the figure, we can see that if the voting is perfect (R_V=1), it always improves the reliability for R_M > 0.5. 6MR is more efficient than TMR but typically with the cost of more components. If the voting circuits could be faulty, voting helps only in a certain range of R_M.

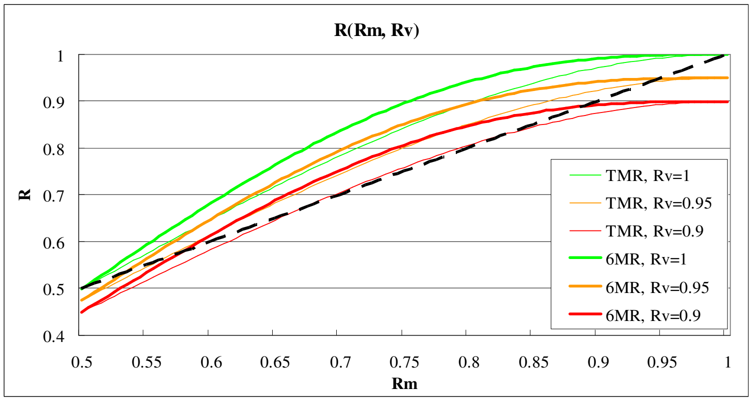


Figure 25. Reliabilities of voting system vs. module reliability with unreliable voters

From the figure, there are 3 possible ways to improve the overall reliability: a) improve the reliability of each module (R_M), b) improve the reliability of voting circuits (R_V), and c) improve the voting logic itself (e.g., 6MR vs. TMR). Based on this discussion, we will show the implementations of nanoscale voting in simple 2-level logic (i.e. NAND-NAND) and H2L

NAND-AND/NAND logic styles and discuss the benefit of the H2L based implementation.

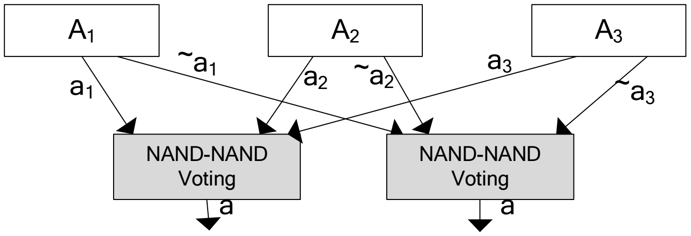


Figure 26. Nanoscale TMR design in pure NAND-NAND fabric

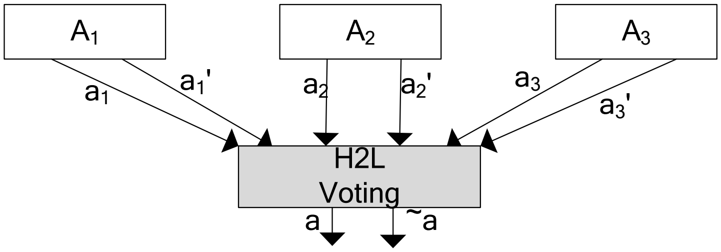


Figure 27. 6MR voting enabled by H2L

As mentioned before, complementary signals are necessary in a fabric based on 2-level logic. An original signal and its complementary version can provide "dual-rail" redundancy. However, voting on "dual-rail" signals (e.g., a_1 and ~a_1 in Figure 26) requires signal inversion, which is difficult to achieve with 2-level NAND-NAND logic on the layout constrained 2-D grid. Therefore, as shown in Figure 26, the voting on original signals and complementary signals are separated from each other in a pure NAND-NAND fabric and the "dual-rail" redundancy is effectively unutilized.

With H2L logic it is possible to produce complementary signals. Given this capability, we can vote only on original outputs and generate the complementary signals in voting circuits using NAND-AND logic only when necessary. As shown in Figure 27, there is no need to generate complementary output in each module (A_1, A_2, and A_3) for voting. Instead, we generate 3 more original copies (a_1', a_2', and a_3') for more redundancy and vote on all 6 signals (6MR). In the voting circuit, the original and complementary outputs are generated for the next stage using H2L NAND-AND/NAND logic.

Note that the area of A_1, A_2 and A_3 in Figure 27 is actually smaller than in Figure 26 (since only one set of horizontal midterms is used for output generation) and they provide more redundancy (6 copies compared with 3 copies in Figure 26). By using H2L, we improve the voting scheme (use 6MR instead of TMR) as well as the yield of each computing module (R_M). Simulation results in subsequent sections will show that the overall reliability is significantly improved.

6.2 Density Evaluation

One of the key drivers for nanoscale systems design is the perceived improvements in density over conventional CMOS technology due to smaller features of nanomaterials as well as the smaller 'footprint' of nanodevices and interconnect. We compare the density of NASIC designs with various fault tolerance schemes incorporated against an equivalent fault-free CMOS based implementation. We use WISP-0 as a design prototype for this study.

To get a more accurate evaluation on density, we need to take the area overhead of MWs into account. Note that the pitch between MWs in nanoscale WISP-0 also scales down with CMOS technology nodes: a reason why NASIC WISP-0 density changes somewhat with assumptions on MWs. Technology parameters from ITRS 2007 [28] used in the calculations are listed in Table II.

Table II. Parameters for Density Calculations

Parameter	**Values**
NW-pitch	10nm
NW-width	4nm
Technology Node (ITRS 2007)	**MW pitch**
45nm	90nm
32nm	64nm
22nm	44nm
16nm	32nm

To get a better sense of what the densities actually mean we normalize the density of nanoscale designs to an equivalent WISP-0 processor synthesized in CMOS. We designed this processor in Verilog, synthesized it to 180nm CMOS. We derived the area with the help of the Synopsys Design Compiler. Next, we scaled it to various projected technology nodes based on the predicted parameters by ITRS 2007 [28] assuming area scales down quadratically. For the purpose of this paper, we assume that the CMOS version of WISP-0 is defect-free with no area overhead for fault tolerance. It is nevertheless expected that even CMOS designs would need redundancy and other techniques to deal with defects, mask irregularities as well as significant parameter variations.

Density comparisons with CMOS are shown in Figure 28. The notation used in the graphs is: *w/o Red* stands for WISP-0 without fault-tolerance techniques (or baseline); *2-way Red* stands for WISP-0 with 2-way redundancy; *2-way Red+TMR/6MR* stands for 2-way redundancy plus nanoscale TMR/6MR. While other combinations are possible, we found these to be insightful and representative.

We observe that NASIC designs without redundancy have orders of magnitude density advantages over conventional CMOS design. For example, NASIC H2L WISP-0 without redundancy is 33X denser than an end-of-the-roadmap 16nm equivalent CMOS implementation.

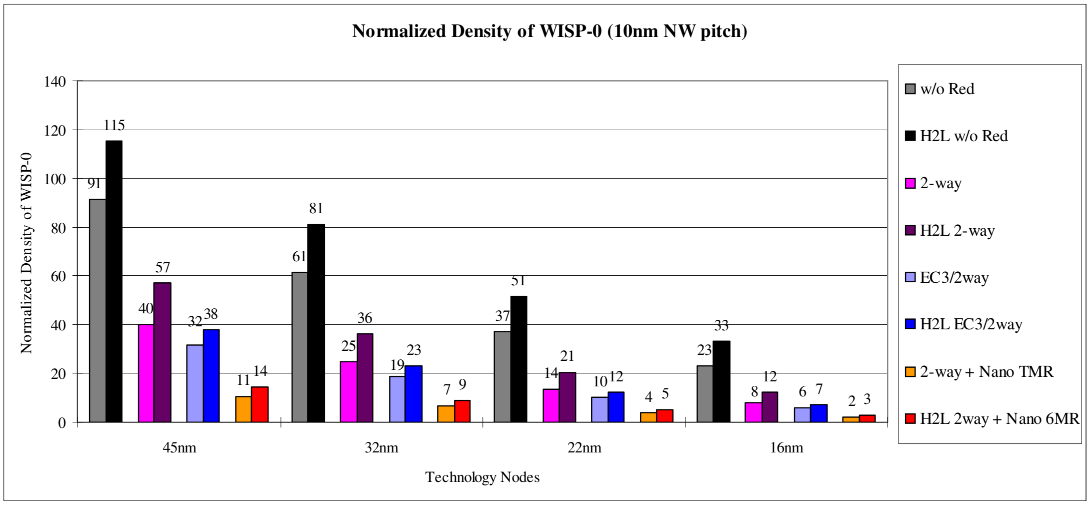


Figure 28. Density Comparison of NASIC WISP-0 with Equivalent CMOS Implementation

However, with various fault-tolerance techniques, some density advantages are lost. Using 6MR voting schemes with H2L still has 3X density improvement over a fault-free CMOS implementation at 16nm. This is a somewhat pessimistic number for NASICs because i) CMOS is assumed to scale perfectly quadratically, ii) no fault-tolerance is assumed for CMOS designs and iii) more aggressive reliable manufacturing processes may be leveraged to achieve NASICs at denser pitches and requiring fewer components with very high levels of fault-tolerance.

6.3 Yield Evaluation for WISP-0

The yield impacts of various built-in fault tolerance techniques described in this section were evaluated for NASIC WISP-0 using a logic and circuit level simulator we developed. Results are summarized in this subsection.

We assumed two types of defects - stuck-on transistors and broken nanowires - in our simulations. Stuck-off transistors are treated as broken nanowires as they have the same effects. From the results (Figure 29 and Figure 30), we can see that our fault tolerance improves the yield of WISP-0 significantly. The yield of WISP-0 without redundancy is 0 when defect rate is only 2%. With 2-way redundancy, however, the yield of WISP-0 is still more than 20% when 5% transistors are defective. As expected, error correction technique achieves better yield than 2-way redundancy. Compared with a 2-way redundancy approach, the improvement of the hybrid approach (*EC3/2-way*) on the yield of WISP-0 is 14% when the defect rate of transistors is at 2%, 129% at 5% defect rate, and 28.6X at 10%. Note that the improvement is greater for higher defect rates. Similar improvements are achieved for broken nanowires.

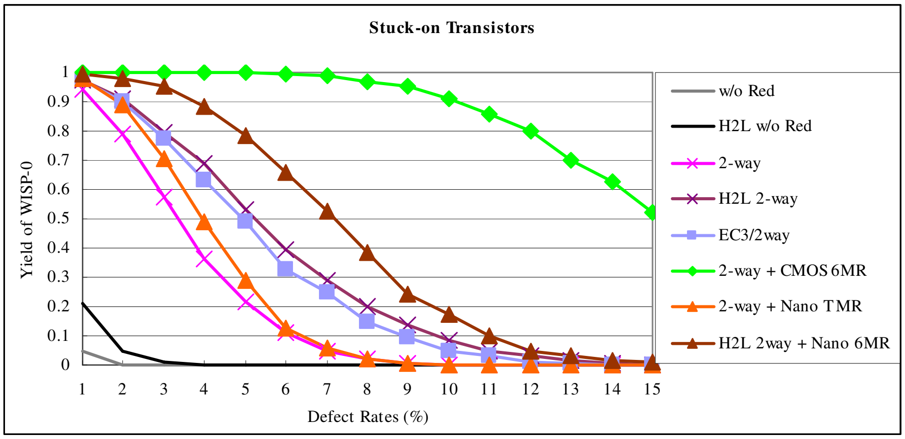


Figure 29. Yield for WISP-0 with different techniques when only considering defective FETs

The H2L logic technique improves the yield considerably. Compared with the AND-OR approach in *2-wuay Red+TMR* scenario, the improvement of H2L logic on the yield of WISP-0 with *2-way Red+6MR* is 15% when the defect rate of transistors is 2% and 147% at 5% defect rate. CMOS 6MR refers to using reliable CMOS voters utilizing the dual-rail redundancy from NASIC tiles. As shown in the figures, CMOS 6MR technique improves the yield of WISP-0 dramatically. This, however poses some serious constraints: i) interfacing between nanodevices and CMOS circuitry will be required between NASIC tiles complicating manufacturability considerably and ii) nanowires will need to drive large capacitive loads associated with

MOSFET gates and parasitics leading to very poor performance. In these figures, yield values of CMOS 6MR are provided as the upper bound for other voting schemes.

Nanoscale voting eliminates the aforementioned issues imposed by CMOS voting. From the figures, we can see that nanoscale TMR technique in NAND-NAND NASIC fabric improves the yield when the defect rate is low. However, the improvement vanishes for higher defect rates. For example, if the defect rate for transistors is higher than 7% or the defect rate for broken nanowires is higher than 3%, the nanoscale TMR is actually deteriorating the overall yield. This is because with high defect rate, voting circuits themselves become so unreliable that they negatively impact yield.

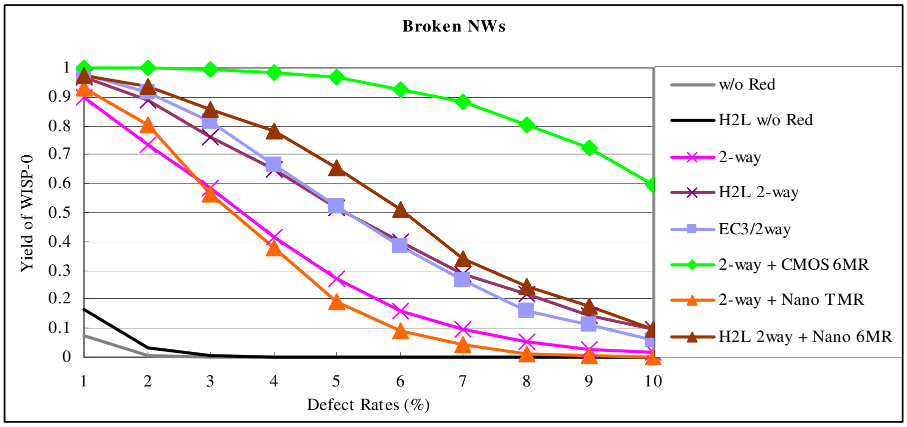


Figure 30. Yield for WISP-0 with different techniques when only considering broken nanowires.

With H2L logic, the effectiveness of nanoscale voting is significantly better. In addition to the yield improvement for the logic itself, the nanoscale 6MR technique in new NASIC fabric consistently improves the yield of WISP-0. Compared with WISP-0 with H2L logic and 2-way redundancy, the improvement of nanoscale 6MR technique is 7% and 47% respectively when 2% and 5% transistors are defective respectively. For broken nanowires, similar results are achieved.

It is important to understand the area overhead (or impact on density) of the different fault-

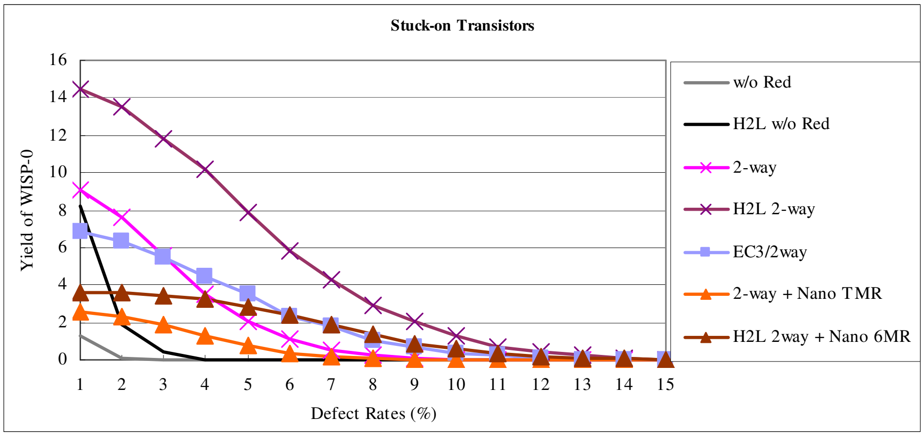


Figure 31. Yield-density products achieved for WISP-0 when only considering defective FETs

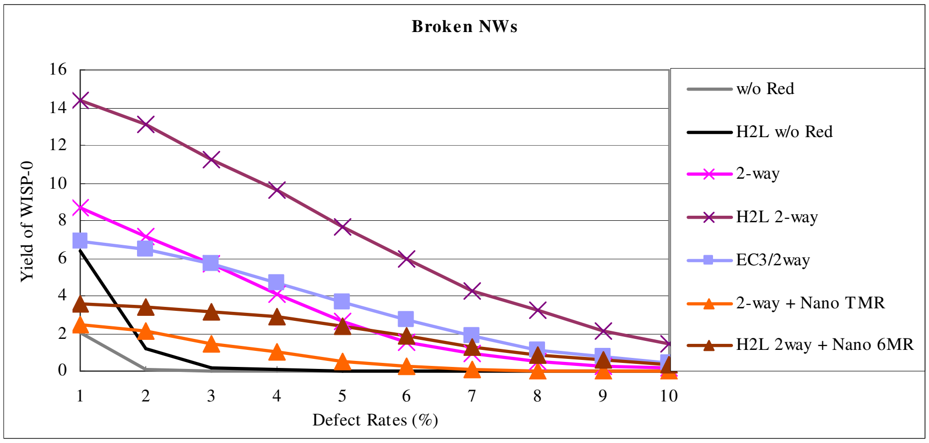


Figure 32. Yield-density products achieved for WISP-0 when only considering broken nanowires.

tolerance techniques in conjunction with their fault masking ability. We define the yield-density product (YDP) metric to capture these effects. The YDP may be interpreted as an 'effective yield' or the number of functional chips obtained per unit wafer size in a parallel manufacturing process.

The yield-density product results for various defect rates are presented in Figure 31 and Figure 32 respectively. We can see that the proposed built-in fault tolerance techniques significantly improve the yield-density products of WISP-0. H2L WISP-0 with 2-way redundancy achieves the best YDP as H2L logic technique improves both yield and density. Nanoscale voting techniques (i.e. *Nano TMR* and *Nano 6MR*) improves the yield of WISP-0 with more than 3X area overhead, deteriorating their YDPs. Future NASIC CAD tools can take yield and yield-density product as different design constraints: if designers care more about yield, nanoscale voting may be necessary. However, if both yield and density are critical, some less area expensive fault tolerant schemes such as 2-way redundancy or EC technique are desirable.

6.4 Process Variation Mitigation

While it is widely accepted that state-of-the-art CMOS designs need to deal with high levels of parameter variation, the defect rates expected are still fairly low. For example, at 90 nm, the expected defect rate is only 0.4 defects/cm^2. Process variation is often dealt with independently from defect tolerance and can be even handled at the architectural level [45][46][47][48]. In contrast, nanoscale fabrics based on self-assembly manufacturing processes are expected to have much higher defect rates (in NASICs we assume 10 orders of magnitude higher or 100s of millions to billions of defective devices per cm^2) in conjunction with high levels of parameter variation. These high defect rates require a layered approach for fault tolerance and typically involve incorporating carefully targeted redundancy at multiple system levels, e.g., structural, circuit, and architectural levels, as has been shown in the previous section. Given that this redundancy is already built in for yield purposes, it is imperative to understand whether it could be exploited to mask delay faults caused by process variation. We can then develop redundancy-based techniques that are more tailored towards process variation than defect tolerance, e.g., by trading off yield for higher performance, or the opposite, depending on design requirements. We can also apply redundancy non-uniformly, taking into consideration defect and process variation masking needs in specific circuits. For example, it might be more critical to have higher resilience against variations in circuit blocks that are part of the critical path, than in blocks that have plenty of timing slack. In non-critical regions, we might be able to apply techniques focusing on improving yield, without sacrificing performance.

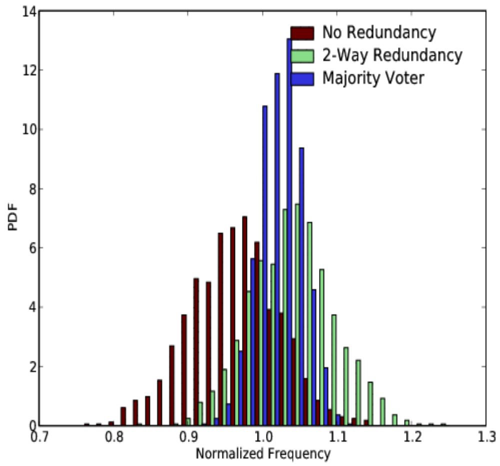


Figure 33. Frequency distribution of simulated NASIC WISP-0 processor

6.4.1 Initial Study on Impact of Redundancy for Masking Delay Faults

To see whether redundancy can mitigate the impact of process variation, we run Monte Carlo simulations of a WISP-0 NASIC streaming nanoprocessor design based on the NASIC fabric with built-in defect tolerance. In our simulations, we consider five parameters that may vary: wire width; metalized wire resistivity; contact resistance; pitch; and gate doping. The wire width modifies both the gate width and length, as NASIC xnwFETs are at the cross-points between nanowires. Gate doping varies the FET equivalent ON resistance [49]. We assume variations of up to 3σ = 30% for all of these parameters. We have explored the frequency distribution of a design without redundancy and after applying two techniques that were developed for defect tolerance: 2-way redundancy and majority voting. 2-way redundancy is based on duplicating every nanowire and device. For majority voting, multiple copies of a logic block are created and a voter determines the block output based on the majority of the individual outputs.

Representative simulation results are shown in Figure 33. All speeds are normalized to the speed at which that type of circuit would run with zero process variation to allow comparing the distributions despite differences in their mean frequency. The figure shows that the normalized frequency for circuits using redundancy tends to be significantly higher than without redundancy. In other words, introducing redundancy greatly increases the percentage of circuits that will operate at or above the nominal frequency in the presence of process variations. Furthermore, when majority voting is used, the variation in frequency is barely half that of the non-redundant and 2-way redundant circuits. These results attest to the potential for redundancy to counter the effects of process variation in addition to introducing defect tolerance and motivated us to consider these techniques in more detail and investigate specific approaches to trade off yield and performance. A number of techniques we are currently actively pursuing are presented next.

6.4.2 Design of FastTrack Techniques

The key idea we are actively exploring is based on inserting nanoscale voters at key architectural points in a design but also unbalancing the redundancy in each of the voter input blocks. The intuition behind this technique is to have some inputs (in some of the blocks) arrive faster than others to address the increased delay due to parameter variation and the delay due to the added redundancy itself. In addition, we leverage the property of NASIC circuits that logic '0' faults are much less likely than logic '1' related faults: therefore, we bias the voters towards logic '0'. Other types of nanoscale fabrics might require a different biasing logic level. In NASICs faulty

‘0’s are much less likely, as their occurrence require a higher degree of faults on nanowires or devices in NASICs. The combination of unbalanced input blocks and biasing define a variety of new techniques depending on the input configuration, redundancy levels for each input, voting type, and biasing applied. The biasing towards ‘0’ also helps mask the impact of the lesser redundancy inputs on overall yield and helps preserve yield. If a lesser redundancy block outputs a ‘0’, there might be no need to wait for slower outputs from higher redundancy blocks. By biasing towards ‘0’, the faulty logic ‘1’s are more easily overridden by the correctly operating circuits, allowing the system as a whole to be run at a higher frequency. We have explored many combinations of redundancy configurations for both defects and process variations and fully implemented in the NASIC processor simulator.

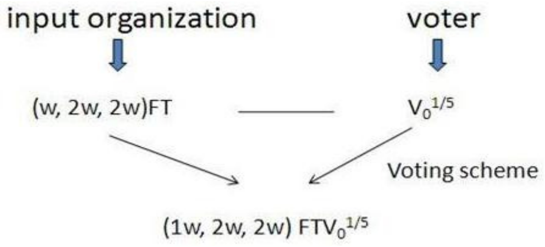


Figure 34. FastTrack redundancy schemes and associated notation.

For example, along these ideas, a so-called FastTrack voting scheme example and notation is shown in Figure 34. Note that the redundancy applied to input blocks to the voter is unbalanced. The figure implies a voter with three blocks, two having 2-way redundancy, and one with no redundancy. Furthermore, the voter is biased towards ‘0’ (see subscript on FTV): if 1 out of 5 inputs coming from the three blocks are ‘0’ the output is voted as logic ‘0’. This is in contrast to a regular majority voter wherein 3 out of 5 inputs would need to be ‘0’ to vote logic ‘0’. Similarly, a plurality voter would require a plurality of ‘0’s to output ‘0’. Plurality and majority voters are in essence instances of biased voters. The combination of input organization and voter type results in a voting scheme denoted, e.g., as (1w, 2w, 2w)$FTV_0^{1/5}$. This notation can be generalized to voting schemes with unbalanced redundancy in their input blocks and biasing in general. If there is no fast tracking the FTV notation is replaced with V. If there is no biasing of output the subscript and superscript after the voter type can be omitted. In the next section the experimental setup and method of evaluation are outlined. This is followed by initial results for a variety of voting schemes based on the above concepts.

6.4.3 Method of Evaluation

Delay calculations are done initially using RC equivalent circuits. A more accurate approach would be to incorporate the device physical model and include changes in actual characteristics due to manufacturing imperfections: This is a direction that is currently pursued but more challenging since for every variation assumed a new 3D model for the devices is extracted from the device simulation.

To this end, we model nanowires with each transistor being changed into ON (low resistance) and OFF (high resistance) modes by its gate input. There is also the resistance of the nanowire interconnect and contact resistance with the CMOS wires that drive the gates. Capacitance sources include inter-wire and junctions between the wires where transistors are formed. The values of resistive and capacitive elements are calculated using their geometry together with several parameters which vary between devices. For our initial experiments, the nominal value of the transistor length evaluated is 4 nm and the width is 4 nm, the square aspect ratio being due to the crossed-nanowire devices. The width of the nanowire depends on e.g., the size of the catalyst nanoparticles used as seeds for Vapor-Liquid-Solid nanowire growth. Variations in nanoparticle sizes therefore directly correlate with variations in nanowire width. The standard deviation of wire widths has been shown to be around 10% of the mean [50]. In addition, the width of the nanowire is assumed to be uniform along its length.

The transistor ON resistance is also varied by the gate geometry – i.e., length and width – that are determined by the width of the nanowires. We do not use such traditional CMOS parameters as V_{TH} and L_{eff}. Instead, we abstract variations in physical parameters into a variation in transistor resistance R_{ON}. Similarly, we calculate the capacitances based on geometries of the nanowires. This allows us to make predictions of delay such that they are anchored in experimental work and geometric variation.

All device parameters are taken from the literature on manufacturing of nanoscale devices and our own devices. *n*-type xnwFETs are used in these calculations. The *n*-type devices are assumed to be silicon nanowires lightly doped with phosphorus. The nominal ON resistance for the specified geometries for *n*-type devices (R_{ON}) has been calculated to be 3.75 Ω based on experimental work [3]. The overall standard deviation of transistor ON resistance has been found to be ~20% [51], including variation of gate geometry. After removing the variation in gate geometry, this reflects a variation of 10% in transistor ON resistance for a square transistor. The nominal contact resistance with CMOS wires delivering VDD and GND has been found to be ~10 kΩ [52].

Interconnect variation is modeled geometrically. The diameter, resistivity, and contact resistance of each nanowire varies independently. Interconnect is assumed to be made by transforming silicon nanowires into nickel silicide (NiSi) via silicidation. The resistivity of the resulting nanowires after thermal annealing averages 9.5 μΩ-cm [11]. Non-uniform metallization can lead to variations in the resistivity of interconnect.

The capacitance is calculated geometrically using the standard expressions for cylindrical wires. These are calculated using the wire diameter and pitch together with circuit layout. The dielectric is SiO_2 with a dielectric constant of 3.9 The nominal value for parallel nanowire capacitance is calculated to be 53.49 pF/m and for junction overlap capacitance is 0.602 aF. Variations in the spacing of nanowires, due to self-assembly based techniques for creating parallel arrays, will lead to variations in parallel nanowire capacitance. The thickness of the dielectric layer between wires (nominal = 5 nm) is determined by the inter-wire spacing, as the space between them is filled with SiO_2 during manufacturing. These parameters are summarized in Table III.

Table III. Parameters used for timing simulation

Parameter	Nominal Value	Standard Deviation
Wire resistivity of NiSi (ρ_{NiSi})	9.5 μΩ-cm	10%
Wire diameter (d)	4nm	10%
Wire pitch	10nm	10%
Contact Resistance	10Ω	10%
Transistor ON Resistance (R_{ON})	3.75kΩ	10%
Oxide dielectric constant (ε_r)	3.9	None
Oxide dielectric thickness (t_{ox})	5nm	10%

In order to evaluate the effects of parameter variation on the WISP-0 processor, we built a simulator that incorporates physical design parameters and is capable of handling both parameter variation and device faults. The simulator uses Monte Carlo techniques to capture statistical distributions. It takes as input a WISP-0 circuit design and a set of manufacturing parameters as described above, including the amount of variation to apply. The simulator based on the variation model given as input varies the characteristics of each nanowire, xnwFETs, and so forth, independently.

Additionally, it takes settings for defects. The defect model consists of a set of defect types and probabilities for each, plus a description of any clustering behavior [22]. For instance, the defect model for a given test might be "5% probability of each transistor being stuck on, with uniform distribution of defects".

The simulator then generates a series of designs based on these parameters, with the values of each individual circuit element changing based on the parameters defined. The speed is then determined by simulating the WISP-0 design and measuring whether it generates correct output, searching to find the fastest speed at which correct output is generated by that test WISP-0 nanoprocessor. If correct output is not generated at any speed due to defects, then the nanoprocessor is noted as faulty and no speed is recorded for that sample. This is repeated many times and each maximum speed is recorded, giving an overall statistical distribution of operating speeds.

It may seem that this could be more quickly done by simply measuring the speed of each gate and taking the speed of the slowest gate as the speed of the nanoprocessor *in toto*, but this would be inaccurate. This is because in many cases, the fault caused by a gate working incorrectly will be masked by the built-in fault tolerance. Therefore, the only accurate way to determine the speed is through simulation of the complete nanoprocessor including all its circuits. Additionally, the nominal frequency of the design, equal to the maximum frequency of the design with all parameters set to their nominal (i.e., zero variation) values, is measured.

6.4.4 FastTrack Results

This section shows initial results with the FastTrack techniques for WISP-0. Three sets of graphs capture the effectiveness of the techniques by plotting effective yield, normalized performance, and effective-yield normalized-performance products. The effective yield is defined as the yield per unit area. The performance is normalized to the slowest result. The third metric is introduced to provide an initial idea for how the techniques work for both performance and yield given the same area overhead.

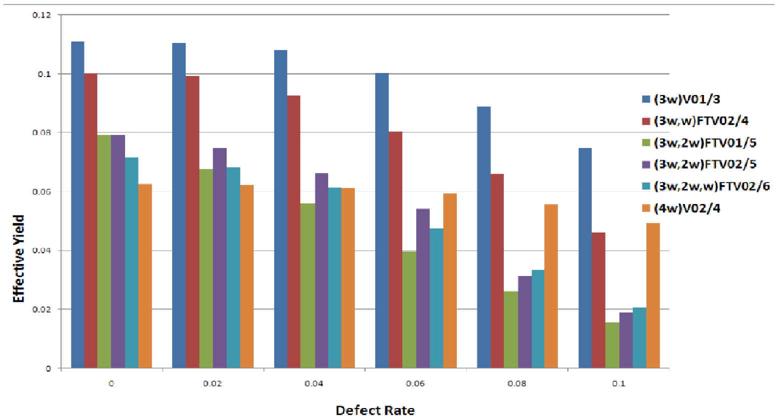


Figure 35. Effective yield comparison of techniques used for parameter variation applied on WISP-0

The results (Figure 35, Figure 36, Figure 37) show that the new FastTrack techniques do best in improving performance but impact yield somewhat especially at higher defect rates. There is no one single technique that performs best overall, considering various defect rates, when both yield and performance are important. The scheme (3w,2w,w)FTV$_0^{2/6}$ performs best in terms of performance and does well up to 6% defect rates even in terms of yield. As expected, the schemes having highest redundancy levels such as (4w)V$_0^{2/4}$ do best at very high defect rates. In these experiments the fault-tolerance techniques have been applied uniformly across all circuits but voting has been added only after the ROM, DECODER, and ALU modules in WISP-0. However, a different balance between fast tracking and redundancy can be used depending on criticality of path in practice. For more details please refer to [53][54].

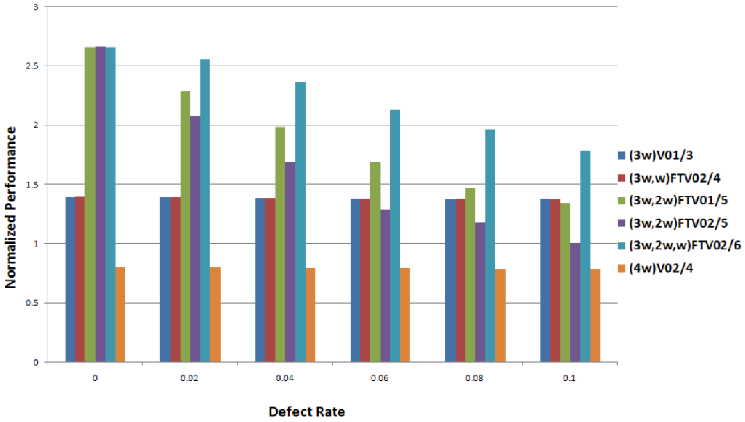


Figure 36. Normalized performance of techniques used for parameter variation applied WISP-0 processor.

Another important result is that the FastTrack schemes can be used also for a performance play. If the parameter variation is less of an issue but a high level of redundancy is needed for yield purposes, FastTrack can mitigate the performance impact of the high redundancy blocks. While FastTrack voters add to the overall latency of the design, they do not affect significantly the throughput. For example, in a processor design they would increase the overall latency of the pipeline somewhat. H2L allows voting to be implemented with very few NASIC gate delays.

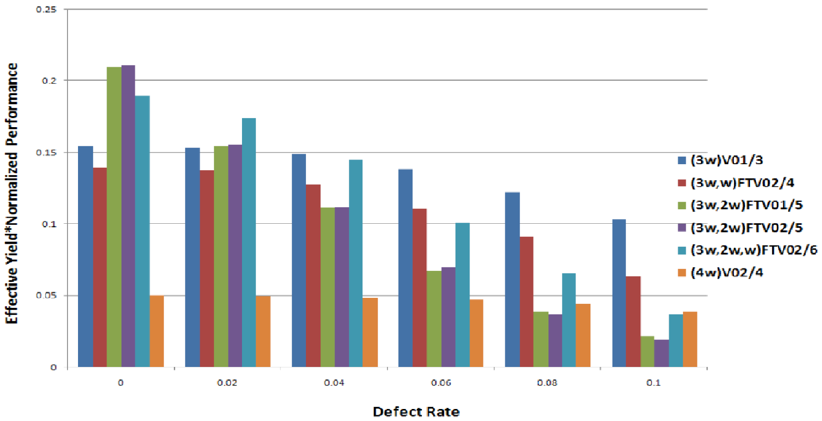


Figure 37. Effective yield * Normalized performance of parameter variation mitigation techniques for WISP-0 processor.

7. Discussion on Performance and Power

Performance and power consumption in NASICs designs depend on a number of factors such as the underlying semiconductor materials, device structures, manufacturing modules, amount of redundancy, logic and circuit schemes, as well as optimizations. Ongoing effort is to derive projections similar to the ITRS roadmap depending on these factors. A brief review of the potential benefits and tradeoffs compared to end-of-the-line projected CMOS is discussed below.

Fundamentally, xnwFET devices should be able to attain a per-device performance comparable to or better than in CMOS devices of similar dimension. The better performance is projected to be a result of more aggressive scaling. As mentioned, the devices for NASICs with various materials and structures are currently in the process of being optimized: a key challenge has been to increase the I_{ON} current. NASICs are based on dynamic style of evaluations that have been shown to be faster than static CMOS MOSFETs at the same feature sizes. Without redundancy, the NASICs circuits have fewer devices per gate although ultimately at the system level the number of devices also depends on the effectiveness of the logic style in the fabric. In this direction, the NASIC H2L and other emerging NASIC logic styles based on three-stage tiles overcome some of the limitations of traditional 2-level logic (e.g., there is no need to carry complementary signals across all tiles anymore) commonly adopted on 2-D layouts. NASIC redundancy techniques have been simulated and the results show a close to linear degradation of performance per tile for higher degrees of redundancy rather than a perhaps expected quadratic degradation, i.e., 2-way redundancy would impact performance roughly 2X. The performance impacts could also be masked somewhat by the Fast Track techniques as described in previous sections.

NASICs have no static power: no direct current path exists between VDD and VSS. Conventional wisdom would dictate that a high I_{ON}/I_{OFF} ratio is necessary to avoid leakage from becoming an issue. However, in NASICs, the control FETs are microwire-based and their work function can be tailored. This allows these devices to act like conventional ground and VDD-gating FETs strongly limiting leakage during precharge and hold phases. The leakage in between the intermediate xnwFET nodes primarily contributes to charge redistribution and has been found to be no problem. In our simulation we found that an I_{ON}/I_{OFF} ratio of 1,000 might be sufficient to control leakage if the control FETs are optimized. Active power is related to precharging and discharging of the nanowires. There are many optimizations possible to reduce it. This includes reducing unnecessary precharging events, controlling charge sharing between intermediate nodes on nanowires, optimizing toward logic '1' outputs vs. logic '0', etc. Note that when a nanowire evaluates to logic '1' there is no discharge and the power consumed is primarily related to leakage/charge redistribution to intermediate nodes and possibly leakage to VSS (that as has been discussed above is mitigated with the microwire-based FETs).

As an initial point of comparison between NASICs and CMOS for the WISP-0 processor, a preliminary evaluation has been completed. Assuming a 0.7V VDD, an I_{ON}/I_{OFF} ratio of 1000, and a contact resistance of 10KΩ, the NASIC WISP-0 was found to achieve ~7X better power-per-performance than in a 16nm CMOS WISP-0. The CMOS version assumes a fully synthesized chip from Verilog using commercial CAD tools for synthesis and place-and-route. It assumes a somewhat optimistic performance scaling between technology nodes and does not consider the impact of interconnect delays. These results are thus preliminary and mainly indicative of the NASIC fabric's potential, since they use simple RC models that are not based on detailed device level simulations of xnwFET devices. Our current work with optimizing and using xnwFETs based on accurate physics models and manufacturing imperfections as well as experimental validation, and efforts to build tools allowing a plug-and-play evaluation for various device structures we develop, will provide a more realistic comparison. These efforts are ongoing and hopefully will be reported in the next few years to the research community.

8. Manufacturing

Reliable manufacturing of large-scale nanodevice-based systems continues to be very challenging. Self-assembly based approaches, while essential for the synthesis and scalable assembly of nano-materials and structures at very small dimensions, lack the specificity and long-range control shown by conventional photolithography. Other non-conventional approaches such as electron-beam lithography provide the necessary precision and control and are pivotal in characterization studies; but these are not scalable to large scale systems. Examples of small nanoscale prototypes include a carbon nanotube FET based ring oscillator [55] and an XOR gate using SNAP assembled semiconductor nanowires and electron-beam lithography [14].

A manufacturing pathway for a nanodevice based computing system needs to achieve three important criteria:

- ***Scalability:*** Large scale simultaneous assembly of nanostructures/devices on a substrate must be possible.
- ***Interconnect:*** Nanodevices, once assembled, must be interconnected in a prescribed fashion for signal propagation and achieving requisite circuit functionality.

- ***Interfacing:*** The nanosystem must effectively communicate with the external world.

In this chapter we explore a manufacturing pathway for NASICs [56] that realizes the fabric as a whole including devices, interconnect and interfacing. This pathway employs self-assembly based approaches for scalable assembly of semiconductor nanowires, and conventional lithography based techniques for parallel and specific functionalization of nanodevices and interconnects. While individual steps have been demonstrated in laboratory settings, challenges exist in terms of meeting specific fabric requirements and integration of disparate process steps.

8.1 Fabric Choices Targeting Manufacturability

Before delving into the details of the manufacturing pathway, it is instructive to look at certain NASIC fabric decisions that significantly mitigate requirements on manufacturing. Choices have been made at the device, circuit, and architectural levels targeting feasible manufacturability while carefully managing design constraints. This is in direct contrast to other technologies such as CMOS which optimize designs for performance and area, but place stringent requirements on the manufacturing process.

- NASIC designs use regular semiconductor nanowire crossbars without any requirement for arbitrary sizing, placement or doping. Regular nanostructures with limited customization are more easily realizable with unconventional nanofabrication approaches.
- NASIC circuits require only one type of xnwFET in logic portions of the design.
- Local interconnection between individual devices as well as between adjacent crossbars is achieved entirely on nanowires; interconnection of devices does not introduce new manufacturing requirements.
- NASICs use dynamic circuit styles with implicit latching on nanowires. Implicit latching reduces the need for complex latch/flip-flop components that require local feedback.
- Tuning xnwFET devices to meet circuit requirements is done in a ‘fabric-friendly’ fashion; techniques such as gate underlap and substrate biasing do not impose new manufacturing constraints.
- NASICs use built-in fault tolerance techniques to protect against manufacturing defects and timing faults caused by process variation. Built-in fault tolerance techniques do not need reconfigurable devices, extraction of defect maps, or complex micro-nano interfacing as required by reconfiguration based fabrics. All fault tolerance is added at nanoscale and made part of the design.

These fabric choices reduce manufacturing requirements down to two key issues: assembling nanowire grids on to a substrate and defining the positions of xnwFET transistors and interconnect. The latter step, also called *functionalization*, is a price paid for a manufacturing-time customization. The manufacturing pathway and associated challenges are discussed in the next sub-section. Note that by adjusting the nanowire pitch any manufacturing issue can managed but the goal is to achieve the smallest possible pitch.

8.2 Manufacturing Pathway

Key steps in the NASIC manufacturing pathway are shown in Figure 38. Figure 38(A) shows a NASIC 1-bit full adder circuit. Horizontal nanowires are grown and aligned on a substrate (B). In general, nanowire alignment can be *in-situ* or *ex-situ*. *In-situ* refers to techniques where nanowires are aligned in parallel arrays during the synthesis phase itself. On the other hand, *ex-situ* refers to techniques where nanowire synthesis and alignment are carried out separately. Currently there are groups pursuing both *in-situ* and *ex-situ* assembly. Lithographic contacts for VDD and GND as well as some control signals are created (B). A photolithography step is used to protect regions where transistors will be formed while creating high conductivity regions - using ion implantation- elsewhere (C, D). Ion implantation creates $n^+/p/n^+$ regions along the nanowires which under suitable electrical fields act as inversion mode source/channel/drain regions.

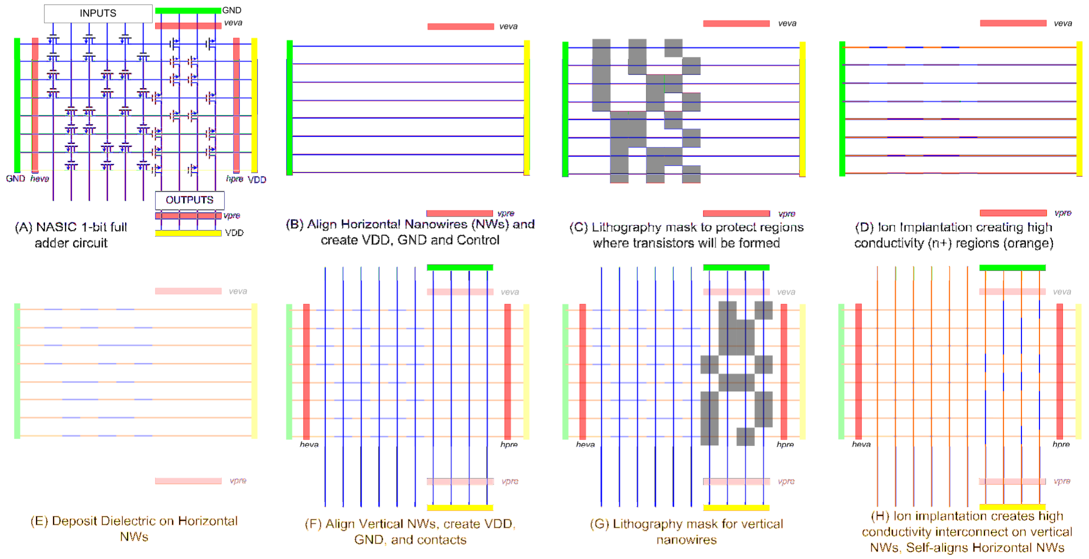
Gate dielectric layer is then deposited (or oxide is grown) (E) followed by alignment of vertical nanowires. The above steps are now repeated for the vertical nanowire layer (F-H). During ion implantation on vertical nanowires (H), channels along horizontal nanowires are self-aligned against the vertical gates.

Key individual steps and challenges are discussed in detail in the following subsections.

8.2.1 Nanowire growth and alignment

The ideal technique to form aligned nanowire arrays should guarantee an intrinsic and concurrent control over three key parameters: (i) the number of nanowires, (ii) the inter-nanowire pitch, and (iii) the nanowire diameter within the array. The state-of-the-art semiconductor nanowire array formation with alignment techniques can be broadly classified into the following three categories:

***In-situ* Aligned Growth**: *In-situ* techniques refer to techniques where cylindrical nanowires are synthesized directly on a substrate in an aligned fashion and do not require a separate transfer step. Examples of these techniques include gas flow guiding, electric field guiding, substrate or template guiding [57][58][59][60][61]. While arrays containing more of less parallel nanowires have been demonstrated using these techniques, the control of pitch and diameter depends on the catalyst engineering, i.e., the ability to form a nanoscale periodic array of catalyst nanoparticles as a dotted straight line, as well as the catalyst material compatibility with the substrate.



(A) NASIC 1-bit full adder circuit

(B) Align Horizontal Nanowires (NWs) and create VDD, GND and Control

(C) Lithography mask to protect regions where transistors will be formed

(D) Ion Implantation creating high conductivity (n+) regions (orange)

(E) Deposit Dielectric on Horizontal NWs

(F) Align Vertical NWs, create VDD, GND, and contacts

(G) Lithography mask for vertical nanowires

(H) Ion implantation creates high conductivity interconnect on vertical NWs, Self-aligns Horizontal NWs

Figure 38. NASIC Manufacturing Pathway

***Ex-situ* Growth and Alignment:** In *ex-situ* growth and alignment, cylindrical nanowires are grown using techniques such as VLS growth [3] and then aligned onto a substrate in a subsequent process step. Examples include Langmuir-Blodgett technique [62][63], fluidic alignment [64], organic self-assembly [65][66] and contact printing [67]. A wide choice of material and synthesis techniques is available with *ex-situ.* Narrow distribution of nanowire diameters is possible since nanowires can be purified post synthesis. However, the key issue is substrate patterning to align nanowires during the transfer step.

Nanolithography Based Pattern and Etch: In these approaches, a semiconductor material layer pre-formed on the target substrate surface is first patterned by nanolithography and then anisotropically etched to create a periodic rectangular nanowire array. Examples include SNAP [14] and Nanoimprint Lithography [68]. These techniques demonstrate excellent pitch and diameter control, but face some key issues: choice of material is limited to silicon; also only one layer of nanowires can be formed using this technique. The surface of those nanowires is usually somewhat damaged too.

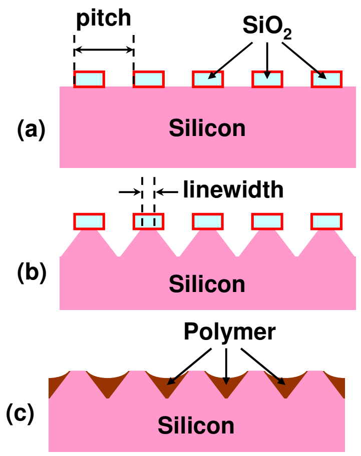


Figure 39. Formation of a nanoscale line array with periodically dissimilar surface properties.

Although many pioneering contributions have been made by the techniques discussed above, there remain several areas for improvement. For instance, some of them rely on the advanced and expensive fabrication processes such as electron-beam lithography and superlattice epitaxy, which have not been used in any mainstream production facilities. In addition, many chemical self-assembly methods could only cover a localized wafer footprint. Even if these nanotechnologies might possibly be scaled up to cover the entire wafer, a wafer-scale precision control as demanded by the low-cost and high-yield manufacturing would deem very difficult if not impossible. The most promising pathway, among others, is the hybrid top-down/bottom-up directed self-assembly (DSA) approach that is however limited by its slow throughput, inaccurate registration, and high defect density.

To address the challenges outlined above in controlling the three key parameters, a novel fabrication process combining top-down lithography and crystallographic etching is being considered that comprises a three-step formation strategy:

- The creation of an intrinsically controlled nanoscale periodic line array on the substrate surface
- The selective attachment of the 1-D nanostructures only onto the lines within the array
- The transfer of the aligned nanostructure array to different substrates

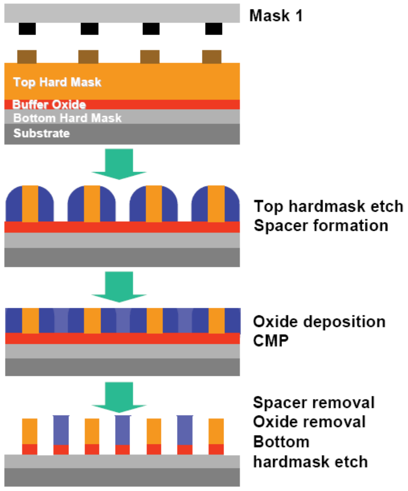


Figure 40. Schematic process flow for spacer double patterning

In the first step, a nanoscale line array on the substrate surface is created with a controlled number of lines, pitch, and linewidth as illustrated in Figure 39. The number of lines and pitch can be defined by conventional lithography (Figure 39(a)) that arguably offers the ultimate intrinsic control and requires a minimal alteration of the existing layout design infrastructure. Even though the state-of-the-art lithography [28] such as the extreme ultraviolet (EUV) lithography might be capable of meeting the pitch requirement, a sub-lithographic pitch could indeed be achieved using the spacer double patterning technique as shown in Figure 40. Besides being a cost-effective way to manufacture the necessary nanoscale pitch from a coarse pitch line array, this approach also permits an alignment to the pre-existing features on the substrate as a side benefit if necessary.

The nanoscale and usually sub-lithographic linewidth can be controlled by using a crystallographic etching (Figure 39(b)) to be larger than, similar to, or smaller than the target nanowire diameter. Contrary to a generic isotropic wet etching, this novel use of the crystallographic etching bears several advantages. First, it offers a very good control over the resultant silicon linewidth as determined by the etching time of the slowest-etching crystal plane. The second one is the creation of precise and repeatable sidewall profiles (Figure 39(b)) for subsequent filling of the inter-space. For example, a material with property different from silicon such as polymer can be used to fill the inter-space to form a resultant surface with periodically dissimilar properties (Figure 39(c)).

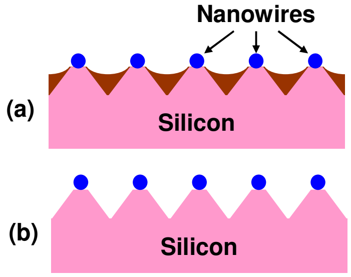


Figure 41. Formation of the aligned nanowire array at the ridge of the triangular features

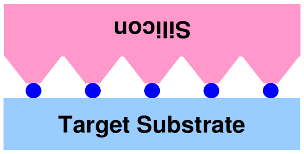


Figure 42. Transfer of the aligned nanowire array onto the target substrate surface

The second step of the formation strategy involves the selective attachment of nanowires only onto the exposed Si line array on the substrate surface, i.e. onto the ridge of the Si triangles (Figure 39(c)), but not onto the inter-space polymer (Figure 41(a)). This can be achieved by manipulating the surface chemistry of the nanoscale substrate lines as well as that of the to-be-assembled nanowires. In other words, the nanowire surface should be attractive to the substrate lines but repulsive to the surface of the inter-space polymer or another nanowire. This desired selectivity could come from either a hydrophobic-hydrophilic interaction or a coulombic interaction. After which, the inter-space polymer would be removed without meddling with the assembled nanowire array (Figure 41(b)). This crucial step could in fact serve as a lift-off process to remove any nanowire that might have randomly adsorbed on the polymer surface, a process that can minimize the resultant array defectivity and thus greatly enhance the manufacturing yield.

The last step is the transfer of the aligned nanowire array onto the target substrate surface (Figure 42), which is readily accomplished with either the wafer manufacturing bonding process or the established stamping transfer process [68][69][70] after slight modifications. The key requirement in this step is an accurate alignment between the Si substrate and target substrate, which could fortunately be fulfilled using a crystallographic alignment mark etched into the Si substrate. This work is currently ongoing yet the preliminary results (Figure 43 and Figure 44) have revealed great promise of this novel technology.

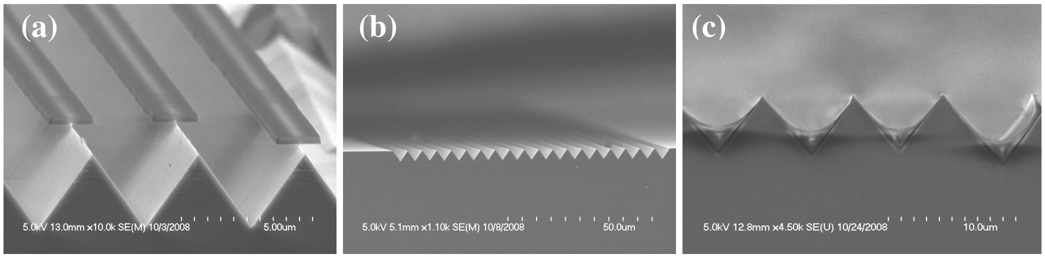

(a)
5.0kV 13.0mm x10.0k SE(M) 10/3/2008 5.00um
(b)
5.0kV 5.1mm x1.10k SE(M) 10/8/2008 50.0um
(c)
5.0kV 12.8mm x4.50k SE(U) 10/24/2008 10.0um

Figure 43. SEM images captured after (a) crystallographic etching (Figure 36(b)), (b) SiO_2 strips removal, and (c) polymer coating (Figure 36(c)), during the formation a surface with periodic nanoscale silicon ridges in between polymer.

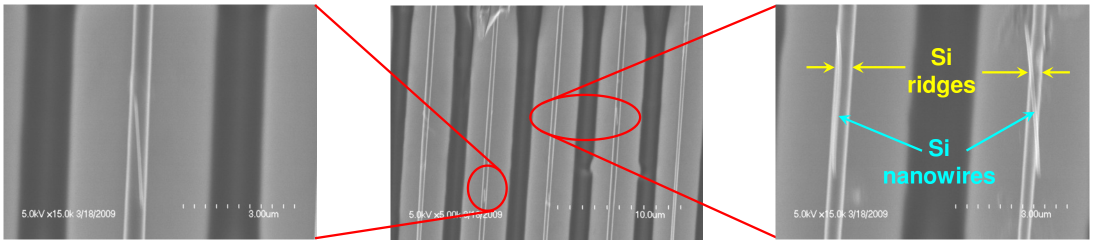

5.0kV x15.0k 3/18/2009 3.00um
5.0kV x5.00k 3/18/2009 10.0um
Si ridges
Si nanowires
5.0kV x15.0k 3/18/2009 3.00um

Figure 44. Preliminary results demonstrating the silicon nanowire selective attachment onto the silicon ridge array (Figure 38(b)). The center SEM image is a low magnification top view above the ridge array. The left and right images are the zoomed-in views unveiling the assembled silicon nanowires.

8.2.2 Nanowire Grid Functionalization

Nanowire grid functionalization refers to defining the positions of xnwFET devices and interconnects on nanowire arrays assembled on a substrate. In the NASIC manufacturing pathway this is achieved by using ion implantation in conjunction with photolithography masks (Figure 38.C-D and G-H). The masks protect the native semiconductor materials on the nanowires to form the xnwFET channels in certain regions, and ion implantation creates high conductivity interconnect as well as the gate material for the nanowires. Thus, devices and interconnect for the nanowires are formed in a parallel process step.

The key consideration for nanowire grid functionalization is the minimum feature size required for photolithography masks. This value is shown to be (2×pitch – width) squares (Figure 45). For example, a nanowire grid with 20nm pitch and 10nm width would require 30nm × 30nm squares.

It is important to note that the lithographic requirements for NASICs are much simpler than a CMOS process of comparable minimum feature size. Masks are used only to protect semiconducting channel regions and not for creating complex patterns on a substrate. Consequently precise shaping and sharp edges are not necessary. Furthermore, built-in fault tolerance techniques can tolerate faults due to improperly formed devices and interconnect. Also the fewer number of

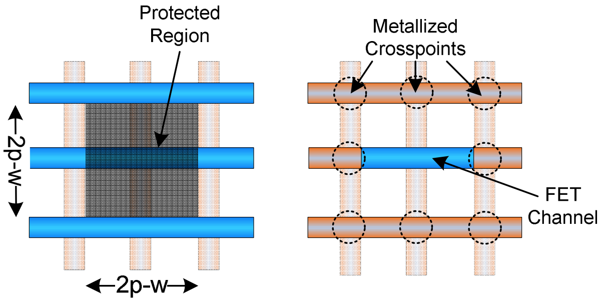


Figure 45 Lithography requirements for NASIC grid functionalization

masks implies much smaller manufacturing costs for NASIC designs.

9. Summary and Future Work

NASIC is a new technology targeting CMOS replacement. All fabric aspects are actively researched including devices, manufacturing, circuit and logic styles, and architectures. A fabric-centric mindset or integrated approach across devices, circuit style, manufacturing techniques and architectures is followed. This mindset is anchored in a belief that at nanoscale the underlying fabric, broadly defined as the state variable in conjunction with the circuit style scalable into large-scale computational systems and its associated manufacturing approach, rather than the device alone, is how significant progress could be made in system-level capabilities. The work to date validated correct fabric operation across both xnwFET devices and circuits, developed circuit structures and fault tolerance techniques, as well as, architectures for processors and image processing. Ongoing efforts are in optimizing devices and developing key manufacturing modules based on both *ex-situ* and *in-situ* approaches. At the circuit level, the current focus is on integrating fault tolerance with support for defects and other fault types caused by process variation and noise, and performance optimizations. Furthermore, there is a concentrated effort to optimize the fabric across devices and circuits for minimizing power consumption. Larger processor designs based on NASICs are in process.

There are several researchers who are actively contributing to various aspects of NASICs, in addition to the authors. Prof Anderson at UMass is developing information theoretical models for projecting the ultimate capabilities in NASICs. In addition to the efforts based on the promising *ex-situ* techniques presented, experimental techniques are investigated for *in-situ* self-assembly based NASIC fabric formation with Profs Mihri and Cengiz Ozkan at UCR, as well as investigators in the UMass CHM nanotechnology center. Profs Pottier, Dezan, and Lagadec and their groups at Universite Occidentale in Bretagne, France, collaborate in developing CAD tools. Profs Koren and Krishna at UMass are collaborating in devising techniques that allow efficient fault masking in NASICs. The authors appreciate the valuable feedback and support at various phases of the project from Dr Avouris, IBM; Dr Kos Galatsis, UCLA; Profs Kostya Likharev, Stony Brook University; Mark Tuominen, UMass; James Watkins, UMass; Richard Kiehl, UCDavis; Kang Wang, UCLA; and many others. This project would have not been possible without the effort of current and former graduate students. Key contributors include Dr Teng Wang, currently at Qualcomm, Dr Yao Guo, currently at Peking University, Dr. Mahmoud Bennaser, currently at Kuwait University, Michael Leuchtenburg, Prachi Joshi, Lin Zhang, Jorge Kina, Kyongwon Park, and Trong Tong Van.

10. Exercises

11. References

[1] C.P. Collier, E.W. Wong, M. Belohradský, F.M. Raymo, J.F. Stoddart, P.J. Kuekes, R.S. Williams, and J.R. Heath, "Electronically Configurable Molecular-Based Logic Gates," *Science*, vol. 285, pp. 391-394, Jul. 1999.

[2] K. Galatsis, Kang Wang, Y. Botros, Yang Yang, Ya-Hong Xie, J. Stoddart, R. Kaner, Cengiz Ozhan, Jianlin Liu, Mihri Ozkan, Chongwu Zhou, and Ki Wook Kim, "Emerging memory devices," *Circuits and Devices Magazine, IEEE*, vol. 22, pp. 12-21, 2006.

[3] W. Lu and C. M. Lieber, "Semiconductor Nanowires," *J. Phys. D: Appl. Physics,* vol. 39, pp. R387-R406, October 2006.

[4] Y. Cui, X. Duan, J. Hu1, and C. M. Lieber, "Doping and Electrical Transport in Silicon Nanowires", *Journal of Physical Chemistry B,* vol. 104, pp. 5213-5216, May 2000.

[5] A. B. Greytak, L. J. Lauhon, M. S. Gudiksen, and C. M. Lieber, "Growth and transport properties of complementary germanium nanowire field-effect transistors", *Appl. Phys. Lett.,* vol. 84, pp. 4176-4178, May 2004.

[6] J. Xiang, W. Lu, Y. Hu, Y. Wu, H. Yan, and C. M. Lieber, "Ge/Si nanowire heterostructures as high-performance field-effect transistor", *Nature*, vol. 441, pp 489-493, May 2006.

[7] M.I. Khan, X. Wang, K.N. Bozhilov, and C.S. Ozkan, "Templated fabrication of InSb nanowires for nanoelectronics," *J. Nanomaterials,* vol. 2008, pp. 1-5, 2008.

[8] Y. Li, G.W. Meng, L.D. Zhang, and F. Phillipp, "Ordered semiconductor ZnO nanowire arrays and their photoluminescence properties," Applied Physics Letters, vol. 76, pp. 2011-2013, Apr. 2000.

[9] X. Duan, Y. Huang, Y. Cui, J. Wang, and C.M. Lieber, "Indium phosphide nanowires as building blocks for nanoscale electronic and optoelectronic devices," Nature, vol. 409, pp. 66-69, Jan. 2001.

[10] Y. Huang, X. Duan, Y. Cui, L.J. Lauhon, K-Y. Kim and C. M. Lieber, "Logic Gates and Computation from Assembled Nanowire Building Blocks", *Science*, vol. 294, no. 5545, pp. 1313-1317, November 2001.

[11] Y. Wu, J. Xiang, C. Yang, W. Lu and C. M. Lieber, "Single-crystal metallic nanowires and metal/semiconductor nanowire heterostructures", *Nature*, vol. 430, pp. 61-65, July 2004.

[12] Semiconductor Nanowires: fabrication, physical properties and applications, Materials Research Society, Warrendale PA, 2006.

[13] Z. Zhong, D. Wang, Y. Cui, M.W. Bockrath, and C.M. Lieber, "Nanowire Crossbar Arrays as Address Decoders for Integrated Nanosystems," Science, vol. 302, pp. 1377-1379, Nov. 2003.

[14] D. Wang, B. Sheriff, M. McAlpine, and J. Heath, "Development of ultra-high density silicon nanowire arrays for electronics applications," *Nano Research*, vol. 1, pp. 9-21, Jul. 2008.

[15] Sentaurus Device User Guide, *Synopsys, Inc.*, 2007.

[16] A. Marchi, E. Gnani, S. Reggiani, M. Rudan, and G. Baccarani, "Investigating the performance limits of silicon-nanowire and carbon-nanotube FETs," *Solid-State Electronics*, vol. 50, pp. 78-85, Jan. 2006.

[17] S. D. Suk, Ming Li, Yun Young Yeoh, Kyoung Hwan Yeo, Keun Hwi Cho, In Kyung Ku, Hong Cho, WonJun Jang, Dong-Won Kim, Donggun Park, and Won-Seong Lee, "Investigation of nanowire size dependency on TSNWFET," *IEEE Electron Devices Meeting (IEDM) Tech. Dig.*, pp. 891-894, 2007.

[18] P. Stolk, F. Widdershoven, and D. Klaassen, “Modeling statistical dopant fluctuations in MOS transistors,” *IEEE Trans. Electron Dev.,,* vol. 45, pp. 1960-1971, 1998.

[19] P. Narayanan, M. Leuchtenburg, T. Wang, C. A. Moritz, “CMOS Control Enabled Single-Type FET NASIC”, *IEEE Computer Society Intl. Sym. On VLSI*, April 2008.

[20] *HSPICE User’s Manual*, Meta-Software, Inc., Campbell, CA, 1992.

[21] T. Wang, P. Narayanan, and C. A. Moritz, “Heterogeneous 2-level Logic and its Density and Fault Tolerance Implications in Nanoscale Fabrics,” *IEEE Trans. on Nanotechnology*, vol. 8, no. 1, pp. 22-30, January 2009.

[22] C. A. Moritz, T. Wang, P. Narayanan, M. Leuchtenburg, Y. Guo, C. Dezan, and M. Bennaser, “Fault-Tolerant Nanoscale Processors on Semiconductor Nanowire Grids”, *IEEE Trans. on Circuits and Systems I, special issue on Nanoelectronic Circuits and Nanoarchitectures*, vol. 54, iss. 11, pp. 2422-2437, November 2007.

[23] T. Wang, P. Narayanan, and C. A. Moritz, “Combining 2-level Logic Families in Grid-based Nanoscale Fabrics**”**, in *Proceedings of IEEE/ACM Symposium on Nanoscale Architectures 2007*, San Jose, CA, October 2007.

[24] M. Sindhwani, T. Srikanthan and K. Vijayan Asari, “VLSI efficient discrete-time cellular neural network processor”, *IEE Proceedings on Circuits, Devices and Systems,* vol. 149, pp. 167-171, June 2002.

[25] L. O. Chua and L. Yang, “Cellular Neural Networks: Theory”, *IEEE Transactions on Circuits and Systems I,* vol. 35, pp.1257-1272, October 1988.

[26] P. Narayanan, T. Wang, M. Leuchtenburg, and C. A. Moritz, “Comparison of Analog and Digital Naroscale Systems: Issues for the Nano-architect”, *IEEE Nanoelectronics Conference,* March 2008.

[27] P. Narayanan, T. Wang, and C. A. Moritz, “Programmable Cellular Architectures at the Nanoscale”, unpublished.

[28] International Technology Roadmap for Semiconductors, 2007 edition. Available online at http://public.itrs.net/

[29] A. A. Bruen, M. A. Forcinito, “Cryptography, Information Theory, and Error-Correction”, Wiley-Interscience, 2005.

[30] I. L. Sayers and D. J. Kinniment, “Low-cost residue codes and their applications to self-checking VLSI systems”, *IEE Proceedings*, vol. 132, Pt. E, No. 4, July 1985.

[31] H. Krishna and J.D. Sun, “On Theory and Fast Algorithms for Error Correction in Residue Number System Product Codes”, IEEE Transactions on Computers, Vol. 42, No. 7, July 1993.

[32] W. H. Pierce, “Interconnection structure for Redundant Logic, Failure-Tolerant Computer Design”, Academic Press, 1965.

[33] R. E. Lyions and W. Vanderkulk, “The use of triple modular redundancy to improve computer reliability”, IBM Journal of Research and Development, 6(2), 1962.

[34] R. C. Rose and D. K. Ray-Chaudhuri, “On a class of error-correcting binary group codes”, Inform. And Contr., vol. 3, pp 68-79, March 1960.

[35] A. Hocquengham, “Codes correcteurs d’erreurs”, Chiffre, vol. 2, pp. 147-156, September 1959.

[36] T.R.N. Rao, “Error Coding for Arithmetic Processor”, Academic Press, ISBN 0-12-580750-4, 1974.

[37] B. W. Johnson, “Design and Analysis of Fault-Tolerant Digital Systems”, Addison-Wesley Publishing, ISBN 0-201-07570-9, 1989.

[38] D. B. Armstrong, “A general method of applying error correction to synchronous digital systems”, Bell Syst. Tech. J., vol. 40, 1961, pp. 477-593.

[39] P. Narayanan, K. W. Park, C. O. Chui, and C. A. Moritz, “Validating Cascading of Crossbar Circuits with an Integrated Device-Circuit Exploration,” in *Proc. IEEE/ACM Symposium on Nanoscale Architectures(NanoArch'09)*, San Francisco, CA, 2009.

[40] A. DeHon. “Nanowire-based programmable architectures”, ACM Journal on Emerging Technologies in Computing Systems, vol 1, pp. 109-162, 2005.

[41] K. K. Likharev, “CMOL: Devices, Circuits, and Architectures. Introducing Molecular Electronics”, 2004.

[42] D. B. Strukov and K. K. Likharev, “Reconfigurable Hybrid CMOS Devices for Image Processing”, IEEE Transactions on Nanotechnology, vol 6, pp. 696-710, November 2007.

[43] G. S. Snider and R. S. Williams, “Nano/CMOS architectures using a field-programmable nanowire interconnect”, Nanotechnology, vol. 18, pp. 1-11, 2007.

[44] Y. Li, F. Qian, J. Xiang and C.M. Lieber, “Nanowire electronic and optoelectronic devices”, *Materials Today*, vol. 9, pp. 18-27, 2006.

[45] M. Bennaser, Y. Guo, and C. A. Moritz, “Designing memory subsystems resilient to process variations,” in proceedings of the IEEE Computer Society Annual Symposium on VLSI, pp. 357-363, Porto Alegre, Brazil, March 2007.

[46] D. Burnett, K. Erington, C. Subramanian, and K. Baker, “Implications of fundamental threshold voltage variations for high-density SRAM and logic circuits,” Symposium on VLSI Technology, pp. 14-15**,** June 1994.

[47] E. Humenay, D. Tarjan, and K. Skadron, “Impact of parameter variations on multi-core chips,” in proceedings Workshop on Architecture Support for Gigascale Integration, June 2006.

[48] A. Agarwal, B. Paul, H. Mahmoodi, A. Datta and K. Roy, “A Process-Tolerant Cache Architecture for Improved Yield in Nanoscale Technologies,” IEEE Transactions on VLSI Systems, Volume 13 (No. 1), pp.27-38, January 2005.

[49] X. Tang, V. K. De, and J. D. Meindl, “Intrinsic MOSFET parameter fluctuations due to random dopant placement,” IEEE Trans. Very Large Scale Integration Systems, volume 5 (No.4), pp. 369–376, Dec.1997.

[50] E. Garnett, W. Liang, P. Yang, “Growth and Electrical Characteristics of Platinum-Nano-particle-Catalyzed Silicon Nanowires”. *Advanced Materials* vol. 19, pp. 2946-2950, 2007.

[51] Y. Cui, Z. Zhong, D. Wang, W.U. Wang, C.M. Lieber: “High Performance Silicon Nanowire Field Effect Transistors”, *Nano Letters* vol. 3, No. 2, pp. 149-152, 2003.

[52] J. Kim, D. H. Shin, E. Lee, C. Han: “Electrical Characteristics of Singly and Doubly Connected Ni Silicide Nanowire Grown By Plasma-Enhanced Chemical Vapor Deposition”, *Applied Physics Letters* 90, 253103, 2007.

[53] M. Leuchtenburg, P. Narayanan, T. Wang, and C. A. Moritz, “Process Variation and 2-Way Redundancy in Grid-Based Nanoscale Processors” *presented at IEEE Conference on Nanotechnology (NANO’09),* Genoa, Italy, 2009.

[54] P. Joshi, M. Leuchtenburg, P. Narayanan, and C. A. Moritz, “Variation Mitigation versus defect tolerance at the nanoscale”, *presented at NanoDDS 2009,* Fort Lauderdale, FL, 2009.

[55] Z. Chen *et. al.*, “An Integrated Logic Circuit Assembled on a Single Carbon Nanotube,” *Science*, vol. 311, Mar. 2006, p. 1735.

[56] P. Narayanan, K. W. Park, C. O. Chui, and C. A. Moritz, “Manufacturing Pathway and Associated Challenges for Nanoscale Computational Systems,” in *Proc. IEEE Conference on Nanotechnology (NANO’09)*, Genoa, Italy, 2009.

[57] R. He, D. Gao, R. Fan, A. I. Hochbaum, C. Carraro, R. Maboudian, and P. Yang, "Si Nanowire Bridges in Microtrenches: Integration of Growth into Device Fabrication," Advanced Materials, vol. 17, pp. 2098-2102, 2005.

[58] Y. Shan and S. J. Fonash, "Self-Assembling Silicon Nanowires for Device Applications Using the Nanochannel-Guided “Grow-in-Place” Approach," ACS Nano, vol. 2, pp. 429-434, 2008.

[59] A. Ural, Y. Li, and H. Dai, "Electric-field-aligned growth of single-walled carbon nanotubes on surfaces," Applied Physics Letters, vol. 81, pp. 3464-3466, 2002.

[60] O. Englander, D. Christensen, J. Kim, L. Lin, and S. J. S. Morris, "Electric-Field Assisted Growth and Self-Assembly of Intrinsic Silicon Nanowires," Nano Letters, vol. 5, pp. 705-708, 2005.

[61] Y.-T. Liu, X.-M. Xie, Y.-F. Gao, Q.-P. Feng, L.-R. Guo, X.-H. Wang, and X.-Y. Ye, "Gas flow directed assembly of carbon nanotubes into horizontal arrays," Materials Letters, vol. 61, pp. 334-338, 2007.

[62] X. Chen, M. Hirtz, H. Fuchs, and L. Chi, "Fabrication of Gradient Mesostructures by Langmuir-Blodgett Rotating Transfer," Langmuir, vol. 23, pp. 2280-2283, 2007.

[63] D. Whang, S. Jin, and C. M. Lieber, "Nanolithography Using Hierarchically Assembled Nanowire Masks," Nano Letters, vol. 3, pp. 951-954, 2003.

[64] X. Xiong, L Jaberansari, M. G. Hahm, A. Busnaina, and Y. J. Jung, "Building Highly Organized Single-Walled-Carbon-Nanotube Networks Using Template-Guided Fluidic Assembly," Small, vol. 3, pp. 2006-2010, 2007.

[65] . Heo, E. Cho, J.-E. Yang, M.-H. Kim, M. Lee, B. Y. Lee, S. G. Kwon, M.-S. Lee, M.-H. Jo, H.-J. Choi, T. Hyeon, and S. Hong, "Large-Scale Assembly of Silicon Nanowire Network-Based Devices Using Conventional Microfabrication Facilities," Nano Letters, vol. 8, pp. 4523-4527, 2008.

[66] B. J. Jordan, Y. Ofir, D. Patra, S. T. Caldwell, A. Kennedy, S. Joubanian, G. Rabani, G, Cooke, and V. M. Rotello, "Controlled Self-Assembly of Organic Nanowires and Platelets Using Dipolar and Hydrogen-Bonding Interactions," Small, vol. 4, pp. 2074-2078, 2008.

[67] A. Javey, S. Nam, R. S. Friedman, H. Yan, and C. M. Lieber, "Layer-by-Layer Assembly of Nanowires for Three-Dimensional, Multifunctional Electronics," Nano Letters, vol. 7, pp. 773-777, 2007.

[68] T. Martensson, P. Carlberg, M. Borgstrom, L. Montelius, W. Seifert, and L. Samuelson, "Nanowire Arrays Defined by Nanoimprint Lithography," Nano Letters, vol. 4, pp. 699-702, 2004.

[69] S. J. Kang, C. Kocabas, H.-S. Kim, Q. Cao, M. A. Meitl, D.-Y. Khang, and J. A. Rogers, “Printed Multilayer Superstructures of Aligned Single-Walled Carbon Nanotubes for Electronic Applications,” Nano Lett., vol. 7, pp. 3343-3348, 2007.

[70] M. C. McAlpine, H. Ahmad, D. Wang, and J. R. Heath, “Highly Ordered Nanowire Arrays on Plastic Substrates for Ultrasensitive Flexible Chemical Sensors,” Nature Mater., vol. 6, pp. 379-384, 2007.