Physical Synthesis onto a Sea-of-Tiles with Double-Gate Silicon Nanowire Transistors

Shashikanth Bobba[1], Michele De Marchi[1], Yusuf Leblebici[2], Giovanni De Micheli[1]
[1]LSI, EPFL, Lausanne, Switzerland [2]LSM, EPFL, Lausanne, Switzerland

Abstract

We have designed and fabricated double-gate ambipolar field-effect transistors, which exhibit p-type and n-type characteristics by controlling the polarity of the second gate. In this work, we present an approach for designing an efficient regular layout, called *Sea-of-Tiles* (SoTs). First, we address gate-level routing congestion by proposing compact layout techniques and novel symbolic-layout styles. Second, we design four logic tiles, which form the basic building block of the SoT fabric. We run extensive comparisons of mapping standard benchmarks on the SoT. Our study shows that SoT with $Tile_{G2}$ and $TileG_{1h2}$, on an average, outperforms the one with $Tile_{G1}$ and $Tile_{G3}$ by 16% and 10% in area utilization, respectively.

Categories and Subject Descriptors

B.7.1 [Integrated Circuits]Design Styles: Advanced technologies

General Terms

Design, Layout, Performance, Regular

Keywords

Ambipolar devices, Regular layouts, Silicon Nanowire FET, Tile

1. Introduction

Layout regularity is one of the key features required to increase the yield of ICs at advanced technology nodes [Tejas 07]. Hence, design styles based on regular layout fabrics have the advantage of higher yield as they maximize the layout manufacturability. Various regular fabrics have been proposed throughout the evolution of semiconductor industry, where some recent approaches are discussed in [Lin 09] [Ran 06] [Taylor 07]. In gate arrays fabric style, a sea of prefabricated transistors is customized to obtain a desired logic gate. The flexibility of building generic logic gates comes at a huge cost of area as well as routing overhead, thereby increasing the performance gap between ASICs and gate arrays. With the advent of via programmable gate arrays [Ran 06] and logic-bricks [Taylor 07], the performance gap is minimized. On the other hand, strict design rules, at 22nm technology node and beyond, has led to cell layouts with arrays of gates with a constant gate pitch, which resemble a sea-of-gates layout style. In this work, we define a regular logic *tile* that has an array of prefabricated transistor-pairs grouped together. A desired logic function can be mapped onto an array of logic tiles.

FinFET transistors are successfully replacing planar CMOS transistors beyond 22nm technology node [Hisamoto 00]. Intel has showcased 37% faster chips with low static and dynamic power consumption with their tri-gate transistor technology,

DAC 2012, June 3-7, 2012, San Francisco, California, USA.

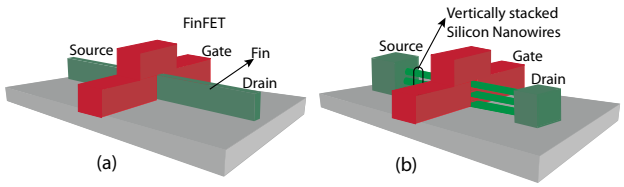

Figure 1. (a) FinFET providing increase in channel area between the source and drain regions (b) Vertically stacked SiNWFET with multiple parallel nanowire channels, each with Gate-All-Around (GAA) control [Saccheto 09].

[Doyle 03], at 22nm node when compared to 32nm planar technology. Following the trend to one-dimensional (1-D) structures, *Vertically Stacked Silicon Nanowire Field Effect Transistors* (SiNWFETs) are a promising extension to the tri-gate FinFETs [Suk 05]. The superior performance of these 1-D channel devices (nanowire FET) comes from a high *Ion*/*Ioff* ratio, due to the gate-all-around structure, which improves the electrostatic control of the channel, thereby reducing the leakage current of the device. Figure 1 shows a tri-gate FinFET transistor and a vertically stacked SiNWFET. In addition, SiNWFET exhibit enhanced electrostatics properties, such as polarity control, which are electrically impossible to planar- and Fin- FETs.

Our methodology takes advantage of the electrostatics of these devices, which can be built to be ambipolar, i.e. to exhibit n- and p-type characteristics. By engineering of the source and drain contacts and by constructing independent double-gate structures, the device polarity can be electrostatically forced to either n- or p- type by polarizing one of the two gates. The in-field polarizability of these devices enables the development of new logic architectures, which are intrinsically not implementable in CMOS in a compact form [Jamaa 08]. However, the routing complexity at the device level increases due to the presence of an extra gate, called *polarity gate* (PG).

Typical CMOS layout techniques involve transistors with a single gate. In the traditional approach for CMOS, compact layouts are realized by optimal transistor chaining of p- and n-type transistors [Uethara 81] [Hwang 90]. However, in the case of ambipolar gates, the polarity of the transistor (p-type or n-type) changes with the input signals. Motivated by these observations, we propose compact layout techniques for *Double Gate Silicon Nanowire FET* (DG-SiNWFET). In order to facilitate this, we propose novel symbolic layouts for ambipolar logic with *Dumbell-Stick* diagrams.

As a second contribution, we design an efficient regular layout brick (called as *tile*), which forms the basic building block for *Sea-of-Tiles* (SoT) design methodology. The basic tile for SoT is optimized for area and regularity. Technology mapping, with logic synthesis tools, on various tiles helped us in choosing an efficient tile for realizing SoT. We show that hybrid tile $Tile_{G1h2}$ and $Tile_{G2}$, on an average, outperforms $Tile_{G1}$ and $Tile_{G3}$ by 16%, and 10% in total area utilization, respectively.

Finally, we demonstrate $Tile_{G1h2}$ (and $Tile_{G2}$) as a basic building block for the future ambipolar logic circuits.

The remainder of this paper is organized as follows. In Section 2, we present the technology background of ambipolar SiNWFET. In Section 3, we introduce novel symbolic-layouts for ambipolar devices and explain the layout techniques based on dumbbell-stick diagrams. In Section 4, we introduce logic tiles for SiNWFETs and perform technology mapping to find an optimal tile for sea-of-tiles design. We conclude in Section 5.

2. Ambipolar SiNWFET

The advantage of SiNWFETs over other one-dimensional devices such as carbon nanotube transistors, is that SiNWs can be fabricated with a top-down silicon process [Ng 07]. Moreover, SiNWs can be built in vertical stacks, thereby giving highly dense array of nanowire transistors [Sacchetto 09].

Figure 2 show a SiNWFET device structure with SiNWs suspended between source and drain pillars. This SiNW is divided into three sections, which are in turn polarized by two gate-all-around gate regions. The center gate region works as in a conventional MOSFET, switching conduction in the device channel by means of a potential barrier. The side regions are instead polarized by a polarity gate, which controls Schottky barrier thicknesses at the S/D junctions and selects the majority carrier type, thus forcing the device to be either n- or p-type.

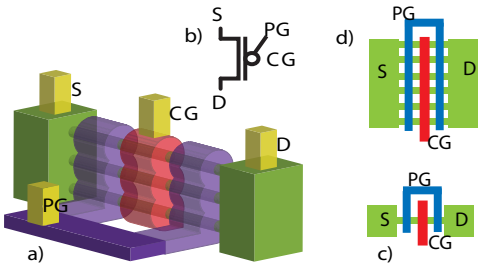

Figure 2. Conceptual structure of the ambipolar DG-SiNWFET: a) 3D view of the device. b) Circuit symbol for the device. c) Top view of the device showing one stack of nanowires forming the channel. d) Large transistor.

A SEM image of an array of vertically-stacked SiNWs, suspended between pillars, before patterning the gates, is shown in Fig. 3a. Figure 3b shows the double-gate SiNWFET after patterning the control and polarity gates. The measured electrical characteristic of the fabricated device is shown in Fig. 4. *Vpg* and *Vcg* correspond to the voltages applied to the polarity gate and control gate respectively. Further device optimization is envisaged for a balanced p- and n- type device.

In order to exploit the unique feature of this device of being polarized electrostatically, a static ambipolar logic family was introduced in [O'Connor 07]. Figure 5a shows a basic logic gate, which can be built with this methodology. An ambipolar transistor constitutes the pull-down network of a Pseudo-CMOS logic gate, having two logic inputs connected to the polarity and the control gate, respectively. In the case of a positive polarity gate input, the transistor behaves as a n-FET, thus producing the output of a classical pseudo-logic inverter. Alternatively, if the polarity gate has a low bias voltage, the transistor behaves as p-type, producing a degraded buffer output characteristic. If we consider this gate as a black box, and see both input signals as logic values, we can see that the gate calculates the XNOR logic function.

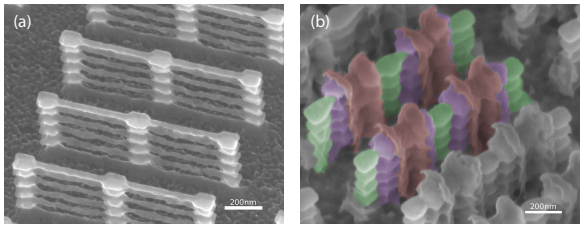

Figure 3. SEM images of a double-gate vertically stacked silicon nanowire FET (a) before the gate patterning; (b) after the gate patterning; Control gate (red); Polarity gate (violet); Active area (green);

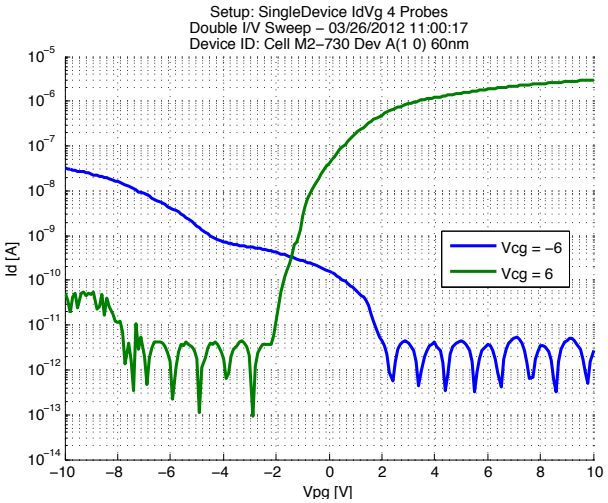

Figure 4. Electrical characteristics of a SiNWFET

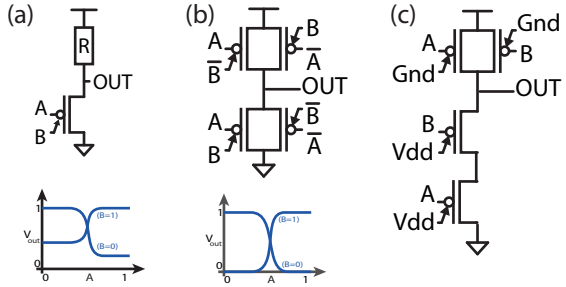

Figure 5. (a) Pseudo-CMOS logic gate with a double gate ambipolar CNTFET in the PDN. (b) Fully complementary XNOR logic gate with opposite polarity transistor pairs in PUN and PDN. (c) NAND logic gate by biasing the polarity gate to either Vdd or Gnd.

In order to obtain a gate featuring full-swing output, a pull-up network substitutes the pull-up resistor, making the logic complementary and each transistor is coupled with another transistor of opposite polarity [Jamaa 09], as in the case of CMOS pass-transistor gates. Figure 5b shows the complete gate, together with its conceptual output characteristic. A two-input XOR function with just 4 transistors portrays the high expressive nature of the double gate ambipolar transistors. CMOS static logic gates (*negative-unate* functions like NAND, NOR, INV, AOI, etc) can be realized by appropriately biasing the polarity gate of the double gate devices. Figure 5c shows a two input NAND gate realized with double gate transistors. The PGs of all the transistors in the pull up network are connected to ground (Gnd), whereas the PGs of all the transistors in the pull down network are connected to supply (Vdd).

3. Layout Technique for Ambipolar Logic Gates

In this section, we first introduce novel symbolic-layouts for ambipolar logic gates, *dumbell-stick diagrams*, based on which we present a layout technique to design complex gates.

3.1 Symbolic Layouts for Ambipolar Logic: Dumbell-Stick Diagrams

Similar to the CMOS stick diagrams, *dumbell-stick diagrams* are proposed for ambipolar devices (in our case DG-SiNWFET) for designing compact layouts by minimizing the cell routing complexity. Figure 6a shows the top view of a DG-SiNWFET (see Fig. 2). The suspended silicon nanowires between the source and drain contacts form the basic dumbell. The control gate and the polarity gate constitute the sticks. Based on this basic building block we present a dumbell-stick diagram of an inverter in Fig. 5d. *A* is the input of the inverter. The nodes *V*, *G*, and *Y* correspond to *Vdd*, *Gnd* and *output* ($\overline{A}$). Transistor pairing, shown in Fig. 5d, is an important transistor placement technique used for layout area reduction. By transistor pairing, two inter-connected pFET and nFET are placed on the same column to minimize the routing complexity as well as to ensure more layout regularity. In Fig. 5e, we show transistor grouping, where the polarity gates of the stacked transistor are connected together. Transistor grouping is unique to ambipolar double-gate devices. In the following section we show the importance of grouping transistors for minimizing the routing overhead introduced by polarity gates.

3.2 Layout Technique for *Unate* and *Binate* Logic Gates

Unate logic functions (e.g. NAND, NOR, AOI, etc) with ambipolar devices are obtained by biasing the polarity gates (PGs) of the *pull-up-network* (PUN) and *pull-down-network* (PDN) to Gnd and Vdd respectively. Hence, all the transistors in the PUN (and PDN) can be grouped together (i.e. PGs of the stacked transistors are connected together), thereby forming one PG for each PUN and PDN. After biasing the PGs, CMOS layout style with transistors aligned according to the Euler paths can be employed [Uethara 81]. The transistors are placed in two parallel rows where all transistors in the PUN are in one row while all the transistors in the PDN are in the other. The main objective is to place transistors in such a way that the gate signals are aligned and drain/source regions of adjacent transistors are abutted. Figure 7a shows an example of a 2-input NAND gate with the PGs biased to either Gnd or Vdd. Figure 7(b,c) shows the final layout of the NAND gate and the dumbell-stick diagram.

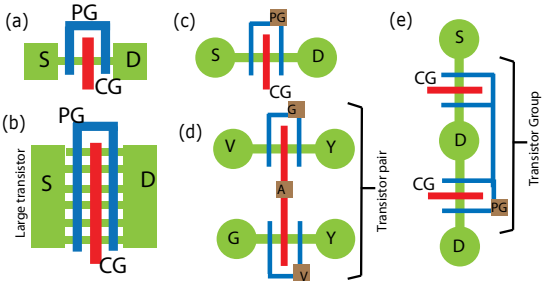

Figure 6 (a) A top view of the DG-SiNWFET shown in Fig. 1. (b) Large transistor. (c) Equivalent dumbell-stick diagram. (d) Dumbell-stick diagram of an Inverter with a transistor pair. (e) Grouping transistor with similar polarity gates.

*3-input XOR gate with 4 transistors is shown in Fig. 11.

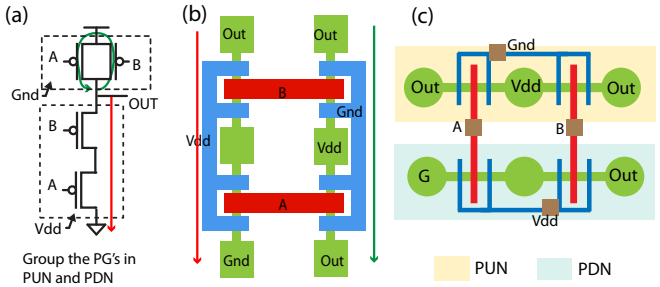

Figure 7. a) Schematic of a static NAND-2 gate by polarizing the ambipolar FET. b) Layout of the NAND gate. c) Dumbell-Stick diagram.

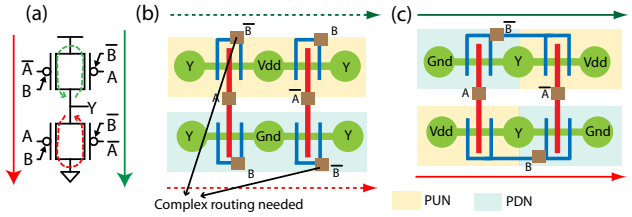

Figure 8. a) Schematic of a 2-input XOR [Jamaa 09] with various Euler paths (b) Dumbell-stick diagram drawn with CMOS style Euler paths (c) Dumbell-stick diagram drawn with Euler paths optimized for transistor grouping.

Efficient implementation of binate logic functions (e.g. XOR and XNOR) is possible by using the polarity gates of the ambipolar FETs as logic inputs. Using the transmission-gate transistor structure of [Jamaa 09], a 2-input (or a 3-input*) XOR gate can be constructed using only 4 transistors. An example of a 2-input XOR gate is shown in Fig. 7a, where all the polarity gates are either connected to logic input B or $\overline{B}$. Unlike for *unate* functions, the polarity gates in the PUN (and PDN) cannot be grouped. In Fig. 7b, we show a dumbell-stick diagram for a CMOS style layout. Since the adjacent transistors cannot be grouped, extra routing effort is needed to connect similar polarity gates together. An efficient implementation is shown in Fig. 7c, where similar polarity gates are grouped together. From the dumbell-stick diagram, we can observe that the PUN and PDN are placed next to each other, which is possible with DG-SiNWFET technology as the transistors are field controlled to make them p-type or n-type.

3.3 Layout Technique for Complex Logic Gates with an embedded XOR/XNOR

Several novel circuit designs and architectures have been proposed which leverage upon embedded XOR functionality of ambipolar logic [Jamaa 09], [De Marchi 10], [Zukoski 11]. In [De Marchi 10], authors have presented the idea of regular logic fabrics and evaluated various complex gates (combination of AND-XOR-OR-INV) based on the number of sub-functions each gate can implement. A key observation is that 2-input XOR/XNOR gates form the main building block of most logic cells. Hence in this work, we focus on layout techniques for complex functions with 2-input embedded XOR function.

Existing CMOS layout techniques have been devised for single-gate transistors and are not applicable to ambipolar transitor network as their polarity (p-type or n-type) changes with the input signals. Hence, modeling a complex gate by two

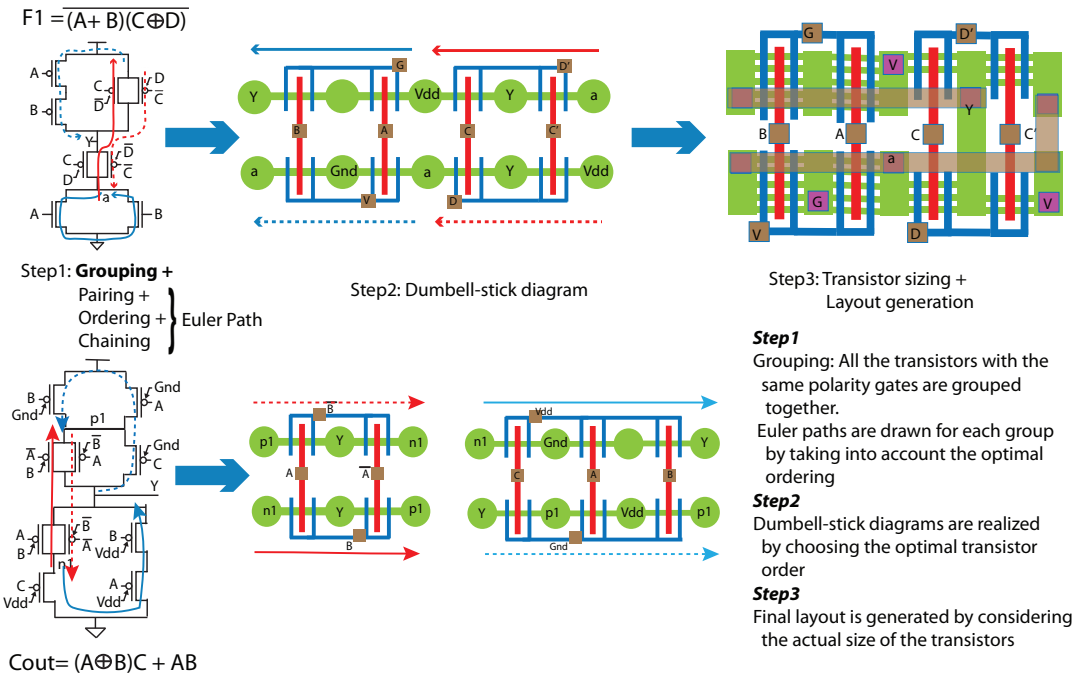

Figure 9. Layout topology generation of complex gates with embedded XOR operation.

graphs, one for p-type devices and the other for n-type devices, is not feasible anymore.

Figure 9 illustrates the layout topology generation for a *carry-out* logic of a *full-adder* and a configurable regular fabric, *F1* [De Marchi 10]. The procedure is summarized here.

Step1: All the transistors with the same logic input on their polarity gates are grouped together. Dual groups are formed based on the complementary signals on the polarity gates. In the example of *carry-out* logic (Cout in Figure 9), the groups formed by polarity gates B and $\overline{B}$ are dual. Similarly the groups formed by polarity gates Vdd and Gnd are dual.

For each dual group, transistors are chained along the Euler paths. In Figure 9 Euler paths in red are shown for the dual group formed by B and $\overline{B}$, whereas the Euler paths in blue are related to the dual group Vdd and Gnd.

Step2: Dumbell-stick diagrams are derived from the Euler paths, as shown in the Fig. 8. In the case of the regular logic fabric, F1, a dumbbell-stick diagram without any discontinuity in the active area is achieved.

Step3: The final layout of the complex gate is generated from the dumbell-stick diagram and by extracting the actual size of the transistors from the schematic.

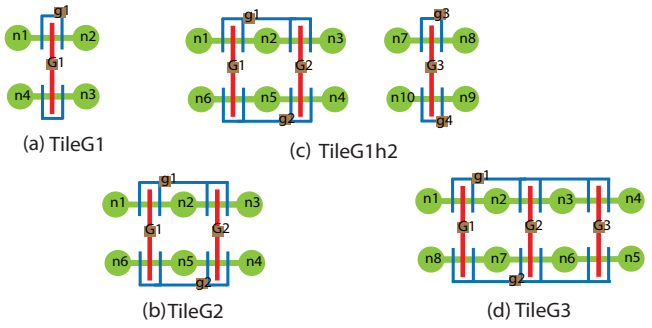

Figure 10. Dumbell-stick diagrams of various logic tiles considered for SoTs (a) TileG1 (b) TileG2 (c) TileG1h2 (d) TileG3.

4. Sea-of-Tiles (SoTs)

Regular layout fabrics have an advantage of higher yield as they maximize the layout manufacturability. In this work we propose a configurable *sea-of-tiles* (SoTs) architecture, in which an array of logic tiles are uniformly spread across the chip. Four different tiles, shown in Fig. 9, are considered in this work. $Tile_{G1}$, $Tile_{G2}$ and $Tile_{G3}$ are regular logic tiles, where $Tile_{G1h2}$ (hybrid tile) is a combination of $Tile_{G1}$ and $Tile_{G2}$.

4.1 Logic Tiles as Building Blocks

In the previous section, we have discussed on ensuring fine-grain regularity in the layouts by transistor pairing and transistor grouping. Transistor pairing helps in aligning the control gates of the complementary transistors in the PUN and PDN, whereas with transistor grouping polarity gates of adjacent transistors are connected together. By grouping the polarity gates of the adjacent transistors we can reduce the number of *input* pins to the connected fabric, *tile*.

We define a logic tile as an array of transistors, which are paired and grouped together. A $Tile_{Gn}$, is an array of *n* transistor-pairs grouped together. All the polarity gates of the top/bottom dumbell are connected together. This is the first step towards minimizing the intra-cell routing congestion. In the example of carry-out logic gate of a full-adder (see Fig. 8), $Tile_{G2}$ and $Tile_{G3}$ are employed to realize the gate. Similarly in the case of NAND and XOR (see Fig. 6 and Fig. 7) $Tile_{G2}$ forms the basic building block. Moreover, the technology facilitates in realizing these tiles with a high yield as the silicon nanowires are fabricated in groups.

Table 1. Various logic gates that can be realized by configuring the $Tile_{G2}$

Logic	n1	n2	n3	n4	n5	n6	G1	G2	g1	g2
XOR2	Gnd	Out	Vdd	Gnd	Out	Vdd	A	A'	B'	B
XNOR2	Gnd	Out	Vdd	Gnd	Out	Vdd	A	A'	B	B'
NAND2	Out	Vdd	Out	Out	-	Gnd	A	B	Gnd	Vdd
NOR2	Vdd	-	Out	Out	Gnd	Out	A	B	Gnd	Vdd
INV	Vdd	Out	Vdd	Gnd	Out	Gnd	A	A	Gnd	Vdd
BUF	O1	Vdd	Out	Out	Gnd	O1	A	O1	Gnd	Vdd

Table 2. Various logic gates that can be mapped by configuring the contacts and the input signals of the four tiles (#N – Number of tiles, and #UF – Utilization factor).

Gates	$Tile_{G1}$		$Tile_{G2}$		$Tile_{G1h2}$		$Tile_{G3}$	
	#N	#UF	#N	#UF	#N	#UF	#N	#UF
AND2	3	0.6	2	0.6	1	0.75	1	1
AND3	4	0.57	2	0.8	1.38	0.67	2	0.57
AOI21	3	0.6	2	0.6	1	0.75	1	1
AOI221	5	0.56	3	0.625	1.62	0.71	2	0.71
AOI222	6	0.54	3	0.75	2	0.67	2	0.86
AOI22	4	0.57	2	0.8	1.38	0.67	2	0.57
AOI321	6	0.54	3	0.75	2	0.67	2	0.86
BUF	2	0.66	1	1	0.62	1	1	0.67
INV	1	0.66	1	1	0.38	1	1	0.67
NAND2	2	0.66	1	1	0.62	1	1	0.67
NAND3	3	0.6	2	0.6	1	0.75	1	1
NAND4	4	0.57	2	0.8	1.38	0.67	2	0.57
NOR2	2	0.66	1	1	0.62	1	1	0.67
NOR3	3	0.6	2	0.6	1	0.75	1	1
NOR4	4	0.57	2	0.8	1.38	0.67	2	0.57
OAI21	3	0.6	2	0.6	1	0.75	1	1
OAI22	4	0.57	2	0.8	1.38	0.67	2	0.57
OR2	3	0.6	2	0.6	1	0.75	1	1
OR3	4	0.57	2	0.8	1.38	0.67	2	0.57
XNOR2	8	0.57	2	0.8	1.38	0.67	2	0.57
XNOR3	9	0.56	3	0.625	1.62	0.71	2	0.71
XOR2	8	0.57	2	0.8	1.38	0.67	2	0.57
XOR3	9	0.56	3	0.625	1.62	0.71	2	0.71

Figure 10b shows an un-mapped (not configured) $Tile_{G2}$. Various logic functions can be realized by connecting the nodes (*n1-n6*) and gates (*g1*, *g2*, *G1* and *G2*) to appropriate inputs. Table 1 lists various logic functions that can be realized with a single $Tile_{G2}$. However, complex logic functions can be obtained by considering an array of $Tile_{G2}$.

Figure 10 shows four tiles that we consider for the sea-of-tiles architecture. $Tile_{G1}$ is the simplest tile with only one pair of transistors. An array of $Tile_{G1}$ is similar to sea-of-gates. Any Boolean logic function can be mapped on to an array of $Tile_{G1}$. The flexibility of building generic logic gates comes at a cost of area. Moreover, providing access to each and every polarity gate increases the intra-cell routing (Metal1 and Metal2 routing) complexity. $Tile_{G2}$ and $Tile_{G3}$ include two and three transistor pairs, respectively, grouped together. A hybrid tile $Tile_{G1h2}$ is a combination of $Tile_{G1}$ and $Tile_{G2}$, whose polarity gates are not connected. This gives the flexibility of utilizing a part of a tile, when remained un-mapped, by functions with low area utilization. For example, a NAND2 gate when mapped onto a $Tile_{G1h2}$ requires only the segment of a tile with gates G1 and G2. The unmapped part of the tile with gate G3 can be employed either to map an *inverter* or to increase the drive strength of the gate. In Table 2, we report various logic gates that can be configured with the 4 tiles we have considered. The number of tiles required for each gate and their respective area utilization is also presented. It has to be noted that we also consider extra logic needed for generating inverted inputs. For example in the case of XOR2/XNOR2, we have discussed in section 3.2 about realizing with only one $Tile_{G2}$. In the case we use single-rail logic, we take an extra $Tile_{G2}$ for generating the two negated inputs ($\overline{A}$ and $\overline{B}$).

4.2 Optimal Tile: Simulations and Result

In this work we compare four tiles for an efficient implementation of the SoT architecture. Our main objective is to find the best tile, which gives highest area utilization for various benchmarks. Though the techniques presented in this paper are linked to the ambipolar SiNWFETs, the concepts can be extended to all the technologies contending for ambipolar logic.

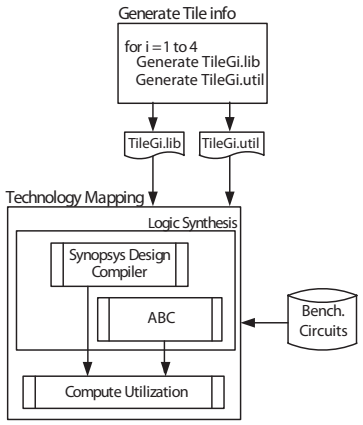

Figure 11. Design flow for finding the best Tile for SoT.

Figure 11 shows our design flow. As a first step, for every tile ($Tile_{Gi}$) we generate a list of logic gates that can be mapped on to it (TileGi.lib) and their respective utilization factor (TileGi.util). Utilization factor takes only the active area into account. For example *NAND2* when mapped onto a $Tile_{G1}$ has a utilization factor of 0.66, whereas when mapped onto a $Tile_{G2}$ it has a utilization factor of 1. It has to be noted that the number of logic gates that can be mapped to different tiles vary. For technology mapping, we used Synopsys design compiler [DC] and ABC [ABC] synthesis tools to benchmark various circuits.

Table 3 summarizes the results of various benchmark circuits after technology mapping. We report total area utilization for each benchmark when mapped onto four different tiles ($Tile_{G1}$, $Tile_{G2}$, $Tile_{G1h2}$, and $Tile_{G3}$). Technology mapping only uses the cells that are associated with each tile (shown in Table 2). Both the synthesis tools were run with different delay constraints. Area utilization for a benchmark circuit is calculated from the total count of each cell and their respective utilization factors. We did not run simulations to study power and delay, as we assume all the tiles with SiNWFETs. Since we map the same netlist onto four different SoTs, it is reasonable to believe that they have the same delay characteristics.

Examining the results for the four logic tiles, we see that SoT with tiles $Tile_{G1h2}$ (and, $Tile_{G2}$) have a higher area efficiency, 10% (8%) and 16% (14%), when compared to SoT with $Tile_{G1}$ and $Tile_{G3}$, respectively. Though $Tile_{G3}$ and $Tile_{G1h2}$ have the same number of transistors per tile, the hybrid tile outperforms $Tile_{G3}$ with 10% improvement in area efficiency.

Embedded XOR functionality is one of the key features of ambipolar logic gates. With a transmission-gate transistor structure [Jamaa 09], a 2-input and a 3-input XOR/XNOR gate can be constructed using only 4 transistors. In Fig. 11, we show how $Tile_{G2}$ can be the most effective layout possible. In Figure

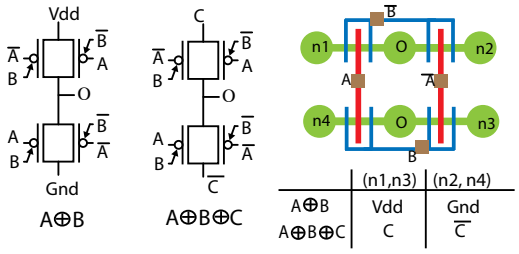

Figure 12. Schematic of a 2-input and 3-input XOR along with the mapping on to a $Tile_{G2}$.

Table 3. Normalized area of various benchmarks when mapped onto a SoT with $Tile_{G1}$, $Tile_{G2}$, $Tile_{G1h2}$, and $Tile_{G3}$ using design compiler [DC] and [ABC].

Bench.	$Tile_{G1}$		$Tile_{G2}$		$Tile_{G1h2}$		$Tile_{G3}$	
	DC	ABC	DC	ABC	DC	ABC	DC	ABC
Dalu	1968	2558	1728	2235	1689	2115	1808	2548
Add64	3946	3004	3693	2664	3483	2483	3560	2740
C5315	4072	5404	3465	4791	3422	4477	3984	5088
C7552	4914	5606	4188	5001	4150	4653	4752	5456
i10	5964	6350	5034	5634	4790	5286	5452	6232
C1908	1132	1778	936	1518	942	1469	1116	1692
C3540	2940	3436	2517	3033	2486	2859	2756	3184
C6288	8462	9336	7227	8253	7373	7744	7580	8000
Des	9392	12482	8142	10623	7910	10323	9016	11912
Average	1	1	0.86	0.87	0.85	0.83	0.94	0.94

13 we show dumbell-stick diagrams of both the *sum* (Sum) and *carry-out* (Cout) logic of a *full-adder*, mapped onto a SoT with $Tile_{G1h2}$. The Sum, which is a 3-input XOR of inputs A, B and C, is mapped on to a *Tile-(i+1,j)* of the entire array. The unmapped part of the *Tile-(i+1,j)* can be employed for realizing either an inverter logic gate or can be a part of the neighboring logic gate. Similarly the Cout is mapped on to 2 tiles *Tile-(i,j)* and *Tile-(i,j+1)*.

Several novel reconfigurable blocks have been proposed which leverage upon embedded XOR functionality of ambipolar logic In Figure 14, we demonstrate how a computational fabric (F1) [De Marchi 10] and a universal logic module (3,2-ULM) [Zukoski 11] can be mapped onto a SoT of $Tile_{G2}$. Inverted inputs, for a 2-input XOR functions, are generated with a single tile (*Tile-(i,j)* for 3,2-ULM and *Tile-(i,j+2)* for F1).

With all the three examples, we demonstrate how tiles, $Tile_{G1h2}$ and $Tile_{G2}$, can be the fundamental building blocks for future ambipolar logic circuits.

5. Conclusion

Double-gate SiNWFETs, with an extra polarity gate, are promising contenders for efficient implementation of ambipolar logic [Jamaa 09]. In this work, we present an approach for designing an efficient regular layout fabric, called Sea-of-Tiles. In order to facilitate design, we propose a compact layout technique and novel symbolic-layout styles for ambipolar logic gates. We show that SoT with tiles $TileG_{1h2}$ and $TileG_2$, on an average, outperform the one with $Tile_{G1}$ and $Tile_{G3}$ by 16% and 10% in area utilization, respectively. We envisage $TileG_2$ or $TileG_{1h2}$ to be the basic building block for the future ambipolar logic circuits.

Acknowledgment

This research was supported by ERC-2009-AdG-246810. The authors would like to thank Davide Sacchetto for his help in fabrication and characterizing the double-gate SiNW transistors.

6. References

[ABC] "Abc system for logic synthesis,"

[DC] "Synopsys design compiler," 2010.

[De Marchi 10] De Marchi, M., *et al.*, "Synthesis of regular computational fabrics with ambipolar CNTFET technology," *IEEE Proc. ICECS,* pp.70-73, Dec. 2010

[Doyle 03] Doyle, B.S., *et al.*, "High performance fully-depleted tri-gate CMOS transistors," *Electron Device Letters, IEEE* , vol.24, no.4, pp. 263- 265, April 2003.

[Hwang 90] Hwang, C.-Y., *et al.*, "A fast transistor-chaining algorithm for CMOS cell layout," *IEEE Trans. CAD*, pp.781-786, Jul 1990.

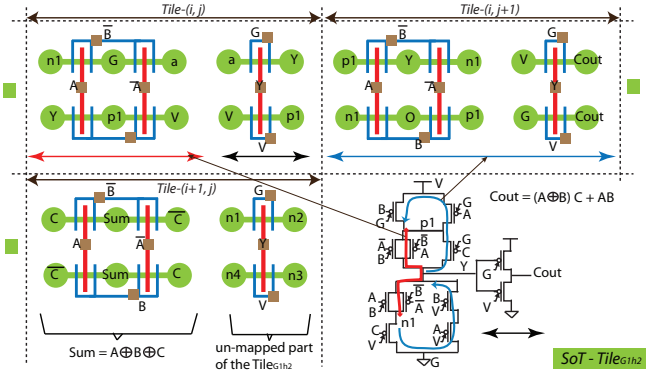

Figure13. A Full-adder mapped on to a Sea-of-Tiles with the hybrid tile *$Tile_{G1h2}$* as the basic building block.

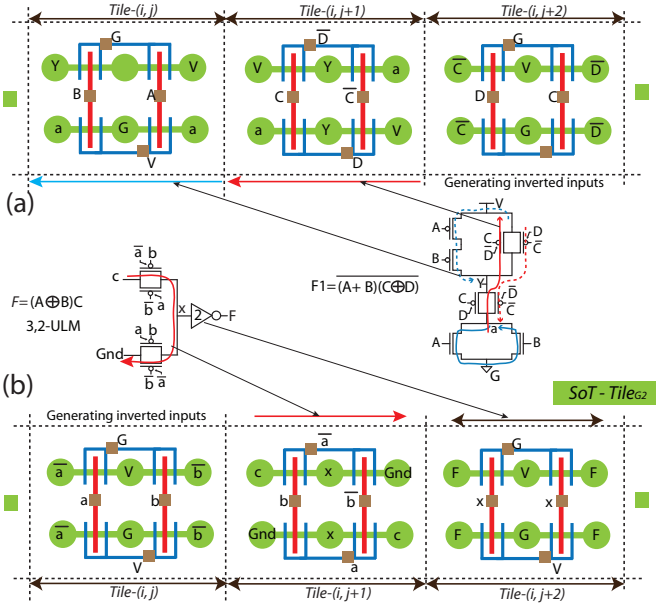

Figure14. Reconfigurable fabrics mapped on to SoT with TileG2(a) Regular computation fabric [Demarchi 10] (b) Universal logic module (3,2-ULM) [Zukoski 11].

[Hisamoto 00] Hisamoto, D., *et al.*, "FinFET-a self-aligned double-gate MOSFET scalable to 20 nm," *IEEE Trans. Electron Devices,* pp. 2320- 2325, Dec 2000.

[Jamaa 08] Jamaa, M.H.B., *et al.*, "Programmable logic circuits based on ambipolar cnfet," *ACM/IEEE Proc. DAC*, 2008.

[Jamaa 09] Jamaa, M.H.B., *et al.*, "Novel library of logic gates with ambipolar cntfets: opportunities for multi-level logic synthesis," IEEE *Proc. DATE '09*.

[Lin 05] Y.-M. Lin, *et al.*, "High-performance carbon nanotube field-effect transistor with tunable polarities," *IEEE Trans*, pp. 481–489, September 2005.

[Lin 09] Yi-Wei Lin, Malgorzata Marek-Sadowska, and Wojciech Maly. "Transistor-level layout of high-density regular circuits", *Proc. ISPD, 2009.*

[Ng 07] Ng, R., Wang, T., and Chan, M., "A new approach to fabricate vertically stacked single-crystalline silicon nanowires," *IEEE Proc. EDSSC*, 2007.

[O'Connor 07] O'Connor, I., *et al.*, "Ultra-fine grain reconfigurability using cntfets*," Proc. ICECS*, dec 2007.

[Ran 06] Y. Ran and M. Marek-Sadowska, "Designing via-configurable logic blocks for regular fabric," *IEEE Trans. VLSI*, vol. 14, no. 1, pp. 1 –14, jan. 2006.

[Sacchetto 09] Sacchetto, D., *et al.*, "Fabrication and Characterization of Vertically Stacked Gate-All-Around Si Nanowire FET Arrays," *Proc. ESSDERC*, 2009.

[Suk 05] Suk, S. D., *et al.*, "High performance 5nm radius twin silicon nanowire mosfet (tsnwfet): fabrication on bulk si wafer, characteristics, and reliability," *IEEE Proc.*, IEDM, dec. 2005, pp. 717 –720.

[Taylor 07] B. Taylor and L. Pileggi, "Exact combinatorial optimization methods for physical design of regular logic bricks," *Proc. DAC,* 2007, pp. 344–349.

[Tejas 07] Tejas, J., *et al.*, "Maximization of layout printability/manufacturability by extreme layout regularity," J. Micro/Nanolith. MEMS, 2007.

[Uethara 81] Uehara, T., and Vancleemput, W., "Optimal layout of cmos functional arrays," *IEEE Trans. CAD*, pp. 305 –312, may 1981.

[Zukoski 11] Zukoski, A., Xuebei Yang., Mohanram, K., "Universal logic modules based on double-gate carbon nanotube transistors," *Proc. DAC*, pp.884-889, 2011.