Logic cells and interconnect strategies for nanoscale reconfigurable computing fabrics

I. O’Connor, K. Jabeur, D. Navarro, N. Yakymets
Lyon Institute of Nanotechnology
University of Lyon, Ecole Centrale de Lyon
Ecully, France
ian.oconnor@ec-lyon.fr

P.E. Gaillardon, M.H. Ben Jamaa, F. Clermidy
CEA-LETI-MINATEC
Grenoble, France

Abstract—The back-gate terminal on double-gate ambipolar transistors can be used as a powerful vector to achieve fine-grain logic reconfigurability. This paper describes ways of exploiting this property to improve on standard cell logic techniques, and to build logic gates with tunable functionalities. Given the vastly reduced transistor count, conventional use of reconfigurable interconnect at the cell level would lead to large overhead, and new interconnect strategies are required. We explore two types of interconnect architectures (island-style and cell-matrix) and develop a mapping method to evaluate trade-offs between matrix occupation, cluster size and switch requirements.

Keywords—nanotechnology; logic design; reconfigurable architectures

I. INTRODUCTION

Performance scaling beyond the endpoint of the CMOS roadmap is likely to be achieved through the use of emerging field effect devices (such as FinFETs, NWFTFs, CNTFETs), where the bulk silicon channel may be replaced with silicon nanowires (or SiGe or GaAs) or carbon (nanotubes or graphene nanoribbons). The resulting level of integration density and potential lack of reliability requires physically regular structures to simplify the process flow, improve robustness to process variability and limit delays. Regular structures allow the design of regular fabrics (gates, memory, interconnect…).

In this work, we show how double-gate ambipolar transistors can be used to build either regular dynamic-logic standard cells with maximal flexibility at the physical (layout) level, or fine-grain reconfigurable logic blocks which can realize any one of fourteen basic binary operation modes. The proposed structures achieve better integration density over CMOS equivalent gates due to a lower transistor count and improved overall dynamic logic gate delay. Such performance levels render conventional architectures inefficient – some static interconnect can be introduced into the overall computing fabric to build a matrix- or island-based approach.

This paper is organized as follows: we begin by presenting the technological hypotheses (section II). In section III, we propose a novel family of DG-CNTFET logic standard cells.

The reconfigurable logic cells are presented in section IV, while the architectural concerns are discussed in section V.

II. TECHNOLOGICAL HYPOTHESES

In this paper, we consider in-field programmable devices exhibiting controllable (double-gate) ambipolarity. While we consider the existing double gate CNT field effect transistors (DG-CNTFET) [1], the circuit techniques described in this paper are also valid using other devices with this property, such as double-gate transistors based on silicon nanowires [2]. In such devices, it is possible to control electrically the channel polarity to three states (N-type, P-type or off) with a polarity gate (PG). For carbon electronics, several techniques to manufacture such devices have been proposed, where ambipolar DG-CNTFETs can have one top- and one back-gate [1], or, by applying the self-alignment technique [3], two top-gates [4]. Here, we consider the latter technique. We also suppose that the technology is based on aligned semiconducting CNT [5], with metal interconnect defined by CMOS-compatible lithography steps.

A sample device cross-section is shown in fig. 1a with the layout drawn in fig. 1b. The gate G turns the device on or off, in the same way as the regular gate of a MOSFET, while the polarity gate PG controls the device polarity setting to N- or P-type with a positive (+V) or negative (-V) voltage, respectively. The device is in the off-state (whatever the voltage on gate G) if the polarity gate is set to 0V. The symbol for this in-field programmable CNTFET is shown in fig. 1c.

Figure 1. DG-CNTFET device cross-section (a), top view (b), symbol (c)

In all electrical simulations in this paper, we use the compact model presented in [6], written in Verilog-A and

This work was funded by the French National Research Agency under the program ANR-08-SEGI-012 “Nanograin”.

---

Bulk Si | SiO₂ | Al₂O₃ | CNT
---

HfO₂ | Al | CNT

Polarity Gate | Control Gate | CNT

D/S G PG

(a) (b) (c)
 extending a unipolar model to double-gate functionality to reflect the ambipolar characteristics of DG-CNTFETs.

III. STANDARD LOGIC CELLS USING AMBIPOLARITY

The logic cells described in this work are derived from dynamic-logic, i.e., based on the precharge and evaluation of a capacitive node. In the conventional approach, this leads to a transistor count of \(n+2\) (where \(n\) represents the fan-in), rather than \(2n\) in a functionally equivalent static-logic implementation. For standard logic functions, we can use the extra DG-CNTFET terminal (PG) to represent a free variable and further reduce complexity. The extra off-state allows the combination of the functionality of the evaluation transistor (dynamic-logic function) with the function transistors, while the programmable polarity also allows us to eliminate input inversion circuitry. In this way, the designed cells achieve a transistor count of \(n+1\).

We devised three approaches to implement logic functions. The resulting structures [7] are defined with a double clock-signal (DCK, fig. 2a), with multiple clock-signals (MCK, fig. 2b) and with a single clock-signal (SCK, fig. 2c).

The DCK-cell (fig. 2a) is composed of a precharge transistor, a function path consisting of one or more DG-CNTFETs and incorporating the evaluation action, and supply rails (+V and ground). In this approach, the complementary function of any given initial function can be realized simply by inverting the supply voltages, or by flipping the layout if the cell connections are symmetrical for an axis parallel to the power lines. The pull-up network (or the pull-down network in the case of an inverted function) is driven not only by data inputs but also by evaluation signals on the back gates. DG-CNTFETs can be placed in series branches (for AND functions), parallel branches (for OR functions) or any combination of series and parallel branches, as with conventional CMOS. The evaluation clock signal is required to control the back-gate of at least one DG-CNTFET for each path from +V or ground to the output node. In operation, the output node is firstly discharged to zero when \(PC=+V\) and \(EV=0V\). PC is subsequently set to 0V (such that the precharge transistor is in the off-state) and EV is set to +V (such that all the transistors in the function path are of N-type). Hence if we consider the function path as a multivariable Boolean function \(f(I_{n1}, I_{n2}, ..., I_{nn})\), the output can be written:

\[
Out = ¬PC \lor (EV \land f(I_{n1}, I_{n2}, ..., I_{nn}))
\]

where we use the logical connective notation (¬ NOT, \(\lor\) OR, \(\land\) AND). The function path transistors front gate inputs are non-complemented. In the case of complemented inputs, DG-CNTFETs can also be configured to P-type using the corresponding voltage on their back-gate (i.e. with \(EV=\{0,-V\}\)). We used N-type devices in the pull-up network in order to keep \(EV=\{0,+V\}\) compatible with the cell input and precharge voltages.

The MCK-cell structure (fig. 2b) is similar to the DCK-cell structure, but here it is possible to use complementary evaluation signals for the back gate control. In this configuration, the EV+ signal dynamic is comprised between \(\{0, +V\}\) and the complementary signal EV- dynamic is comprised between \(-V, 0\). This allows the function path to contain both N- and P-type switching behaviour during the evaluation phase and to achieve complex functions with fewer transistors.

The SCK-cell structure (fig. 2c) uses only a single clock signal. In this approach, the function path is realized in a pull-down network rather than a pull-up network as in the previous structures. The clock signal is applied to the front gate for the P-type precharge transistor for a PC phase at Clk=0, and to the back gates in the function path for an evaluation through an N-type switching network during the EV phase at Clk=+V. The structure is inherently more suited to inverted functions (Inverter, NOR, NAND…) and does not suffer from any logic level degradation (such as that observed in the DCK- and MCK-gate structures) because a P-type transistor is used to precharge the output to +V, and N-type transistors are used to conditionally discharge the output to 0. In this way, we guarantee a better immunity to output signal degeneration and avoid any eventual use of supplementary level-restorer circuits. Further, the complementary function of any given initial function can be realized (as with the DCK-gate structure) by simply inverting the supply voltages. However, the output signal voltages will degenerate both for logic levels “1” (+V-\(m\)\(V_{TN}\), where \(m\) represents the maximum number of N-type transistors in series in the function path) and “0” (+V\(T_{TH}\)). Hence if we consider the function path as a multivariable Boolean function \(f(I_{n1}, I_{n2}, ..., I_{nn})\), the output can be written:

\[
Out = ¬(Clk \land f(I_{n1}, I_{n2}, ..., I_{nn}))
\]

Various performance metrics such as power consumption, transistor count and delay of the new family of standard cells were evaluated to highlight the characteristics of the new cell structures. We compared the proposed standard cells to their equivalent gates working with conventional CMOS-like dynamic logic (denoted CDL), with the same devices.

Simulations were carried out with the same CNTFET transistor model and the same operating conditions. We used a clock frequency of 3GHz with equal rise and fall times (20ps) and capacitive load of 150fF. The supply voltage was 1V and clock and data inputs were full rail (i.e. \(+V=1V, -V=1V\)). Cyclic simulations were carried out to establish mean power consumption and worst-case time delay over all data combinations. Tab. 1 shows the results of simulations in terms of average power and worst-case time delay simulations. All data was averaged for a range of logic function structures.
We can see from these results that the xCK-cell library achieves an overall improvement of 2x concerning the worst-case time delay, compared to the CDL structure family. This decrease is due to the absence of an evaluation transistor in the function path, reducing equivalent channel resistance and associated time constant with load capacitance accordingly. However, while using fewer transistors, the average power consumption increases slightly (10%).

IV. FINE-GRAIN RECONFIGURABLE LOGIC CELLS

The impact of the polarity gate on the transistor channel transport characteristics also enables the construction of dynamically reconfigurable logic cells, such as DRLC_7T which can be configured to any one of fourteen basic binary operation modes (fig. 3).

![Dynamically reconfigurable logic cell (DRLC_7T)](image)

**TABLE II. 3-INPUT CONFIGURATIONS FOR DRLC_7T WITH 3 LOGIC LEVELS (+V, 0, -V) AND CORRESPONDING 14 BASIC BINARY LOGIC FUNCTIONS**

<table>
<thead>
<tr>
<th>V_{ba}</th>
<th>V_{bb}</th>
<th>V_{bc}</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V</td>
<td>+V</td>
<td>+V</td>
<td>A\lor B</td>
</tr>
<tr>
<td>+V</td>
<td>+V</td>
<td>-V</td>
<td>A\lor B</td>
</tr>
<tr>
<td>V</td>
<td>0</td>
<td>+V</td>
<td>~A</td>
</tr>
<tr>
<td>+V</td>
<td>0</td>
<td>-V</td>
<td>A</td>
</tr>
<tr>
<td>-V</td>
<td>-V</td>
<td>+V</td>
<td>A\land B</td>
</tr>
<tr>
<td>-V</td>
<td>-V</td>
<td>-V</td>
<td>A\land B</td>
</tr>
<tr>
<td>+V</td>
<td>-V</td>
<td>+V</td>
<td>B\land A</td>
</tr>
<tr>
<td>+V</td>
<td>-V</td>
<td>-V</td>
<td>B\land A</td>
</tr>
<tr>
<td>0</td>
<td>+V</td>
<td>+V</td>
<td>~B</td>
</tr>
<tr>
<td>0</td>
<td>+V</td>
<td>-V</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>-V</td>
<td>0</td>
</tr>
<tr>
<td>-V</td>
<td>+V</td>
<td>+V</td>
<td>A\lor B</td>
</tr>
<tr>
<td>-V</td>
<td>+V</td>
<td>-V</td>
<td>A\lor B</td>
</tr>
</tbody>
</table>

DRLC_7T is made up of 7 CNTFETs arranged in two logic stages: the first stage performs an elementary logical operation and the second stage works either in follower or inverter mode.

- A and B are boolean data inputs (voltages at A and B vary between 0V and 1V);
- V_{ba}, V_{bb}, V_{bc} are control inputs which configure the circuit to a basic binary logic function according to tab. II (control bias voltages may take one of three values, as indicated previously);
- PC_1, PC_2 (pre-charge) and EV_1, EV_2 (evaluation) are four non-overlapping clocking inputs with pre-charge and evaluation periods as in classical CMOS dynamic logic gates;
- Y is the circuit output.

It is possible to derive other logic circuit structures using correlation techniques and zero-suppressed BDDs, both in dynamic-logic and static-logic styles. While this is outside the scope of this paper, such techniques are useful both to tune the function set to specific requirements, and to achieve the theoretical maximum number of functions. In the following section, we will explore how such families of fine-grain reconfigurable logic cells can be assembled to build nanocomputing architectures.

V. MATRIX-BASED NANOCOMPUTER ARCHITECTURES

To assemble logic cells in matrix-based nanocomputer architectures, it is necessary to establish multiple levels of hierarchy. The lowest level is the fine-grain logic cell that can be configured to one of many available logic functions, as discussed in the previous section. With clear and modular hierarchy, there is however no restriction on the cell type. The next level is a matrix composed of logic cells (fig. 4a), where the interconnect network between cells is fixed and the cells as drawn have two Boolean data inputs A and B and one duplicated data output Y (control and clock inputs not shown). A thorough analysis of matrix interconnections shows that the utilization of small matrices where cells are connected via the “Cross Cap” topology leads to the highest mapping success rate.

![Matrix of nxw (w=width, d=depth) cells with fixed interconnect topology](image)

(a) Cell Matrix-based architecture: a cluster of 2\times2 matrices (b)

At the highest level, the matrices can be assembled to implement a homogeneous Cell Matrix [8] that appears to be suitable for the fine-grained logic cells (fig. 4b), or connected into a cluster (fig. 5a) using one of the conventional FPGA architectures [9],[10]. In popular island-style FPGAs, the primary inputs and outputs of cells link to the routing tracks via the connection boxes (CBs), and different tracks are connected via the switch boxes (SBs). SBs consist of bidirectional programmable switches connecting terminals on different sides. The best routability for such architectures can be
achieved with a SB connectivity factor $F_S = 3$ (this represents the maximum number of switches connected to a terminal in the SB) and with the full connectivity of CBs (i.e. the maximum number of tracks a pin can be connected to) [11] (fig. 5b).

Several parameters have significant impact on the efficiency of the architecture, such as cluster size and the percentage of switch connectivity. Concerning the cluster size, $4 \times 4$ and $8 \times 8$ clusters are the most efficient in terms of routability [12]. However, more than 1000 switches will be used in a $4 \times 4$ cluster of $2 \times 2$ matrices if the wire width is equal to 4 and $F_S = 3$. To reduce the switch overheads without any impact on cluster routability, it is possible to connect SBs and CBs linked via the wire segments that terminate together at the edge of a cluster [9]. The combination of partially connected CBs (fig. 5c) with an SB «Step» architecture (fig. 5d) leads to a 54% reduction in the SB and CB size. Thus, the total number of switches in a $4 \times 4$ cluster of $2 \times 2$ matrices decreases to 520.

To explore the potential of nanocomputer architectures, new approaches and synthesis tools must supplement conventional techniques [13][14]. We implemented a methodology that supports the automatic mapping of Boolean functions onto nanoscale circuits. Using this methodology with a randomly generated set of Boolean functions, we conducted a comparative analysis of both island-style and Cell Matrix-based architectures, and assessed the design area in terms of the number of transistors involved in the final mapping. The results are shown in tab. III. We found that the Cell Matrix-based architecture involves less switches, but requires a significant number (up to 58%) of logic cells for buffers to propagate data from the primary inputs to the primary outputs within a cluster. Thus, the application of SBs and CBs in an island-style architecture helps to reduce the transistor overheads by 44%. Almost all logic cells in the Cell Matrix-based cluster are mapped, whereas more than half of the SB/CB-based cluster is free and potentially can be used to implement additional functions.

VI. Conclusion

In this paper, we have described both circuit-level and architectural approaches to the use of ambipolar DG-CNTFETs in logic architectures. We demonstrated three generic structures to build a library of clocked standard cells, which require only $n+1$ transistors rather than $n+2$ typically required for conventional dynamic logic cells, for comparable average power consumption figures, and with reduced (up to 2X) worst-case delay. The same property was also used in a reconfigurable logic cell, which served as the basis for exploration of nanocomputer architectures. We considered cell-matrix and island-style architectures, and showed that the application of switch and connection boxes can help in reducing the transistor overheads up to 44% in an island-style architecture, as well as in leading to better cluster routability compared to the Cell Matrix-based architecture.

**REFERENCES**


---

**TABLE III. ANALYSIS OF SB/CB- AND CELL MATRIX-BASED 4X4 CLUSTERS**

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Island-style</th>
<th>Cell Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of transistors involved in mapping, F</td>
<td>758</td>
<td>1350</td>
</tr>
<tr>
<td>Percentage of mapped matrices in a cluster, %</td>
<td>44</td>
<td>100</td>
</tr>
<tr>
<td>Number of switches added to connect matrices</td>
<td>520</td>
<td>0</td>
</tr>
</tbody>
</table>