Asynchronous Nano - electronics

Prof. Katke Sharda P.
Department of Electrical Engineering
BMIT Solapur , India

Prof. Rangdal Sachin M.
Department of ENTC
VVPIET Solapur, India

ABSTRACT: This paper is an implementation of asynchronous QDI logic in molecular nano-electronics. The main building blocks of QDI logic can be successfully implemented; we have illustrated the approach with the layout of an adder stage. The proposed techniques is to improve the reliability of QDI by applying nano-CMOS...

Keywords: Molecular Nanotechnology, QDI Logic

I. INTRODUCTION

Currently, molecular nano-electronics is considered a plausible successor to CMOS. Although immense fabrication difficulties still lie ahead, at least experimental devices are feasible today. Molecular nano-electronics is a mostly non-lithographic bottom-up fabrication technology whose main advantage over CMOS is to break the limit of lithography in terms of feature size and device density. All technologies at the nanoscale level, including nano-CMOS, will face great fabrication challenges that will translate into important parameter variations and decreased reliability. Timing will be difficult to control and predict. Wires will be short, by necessity and by choice: the technology does not allow to grow long wires reliably and wire delay is quadratic in the length. Consequently, it will be impossible to build a useful global clocking network with those technologies. For those two main reasons—difficulty in controlling and predicting timing and impossibility of building a clock network—asynchronous logic seems an ideal and perhaps unavoidable choice for digital circuits in this technology. QDI, which among all types of asynchronous methods, relies on the weakest timing assumption (isochronic fork), seems particularly well suited. This paper is a preliminary investigation in implementing asynchronous (QDI) logic in molecular nanotechnology. It is first and foremost an assessment, and further improvement, of QDI's robustness to extreme parameter variations and restricted geometry. The paper is organized as follows. First, we briefly describe the main features of this new technology. Among several possible alternatives we choose a somewhat idealized one based on complementary FETs. We define its main layout rules. Next, we show how to design basic combinational gates. State-holding elements like C-elements and precharge function-blocks require some attention because the traditional "staticizer" implementation must be avoided. We give an implementation for those cells that avoid staticizers. We show how the logic family of the Caltech Asynchronous Microprocessor (CAM) and the logic family of the MiniMIPS can be both implemented without use of staticizers. We illustrate the design style with the implementation of a ripple-carry adder. Finally, the implementation of isochronic forks with its implied timing assumption is analyzed.

II. MOLECULAR NANOTECHNOLOGY

Our target technology is a somewhat advanced version of the devices produced in the Heath Lab at Caltech, the Lieber Group at Harvard and the HP group, [5, 6, 1]. Chemists are able to grow silicon nano wires (NW) with diameters around 5nm and up to ten microns in length. These wires are aligned in one direction, either by a flow process or by nano-print. Because of the difficulty in arranging the wires in a regular pattern, the feasible geometry is restricted to a crossbar: a collection of parallel wires in one direction superimposed with a collection of parallel wires in the orthogonal direction. However, the grid is not perfectly regular and several wires may be broken. Interesting things may be made to happen at the intersection of two orthogonal wires. Special molecules can be placed at the cross point between two wires to connect the two wires. Such a junction is originally of high enough resistance that the two wires are not electrically connected. But, if a high voltage is applied between the two wires, the molecule will begin inducting and continue to do so when the voltage is lowered. In other words, by applying voltage at selected junctions, the junctions can be programmed to conduct. The junctions can also be made to rectify, i.e., we can program both resistors and diodes. Resistors and diodes are sufficient to build a complete logic family and have been used to build PLAs and memories, but they have no gain (amplification) and therefore signal degradation is unavoidable, leading to failure in any computation containing a significant number of transitions in series. In order to implement general computations, transistors are needed to provide amplification, which in turn requires being able to make semi-conductors out of silicon NWs. NWs can

be doped during the growth in order to control their conductivity. Heavily doped regions of a wire are conducting; some regions can be kept lightly doped so as to control their conductivity via an electrical wires, one has effectively created a FET at the junction. Technologies with one type of transistors, diodes, and resistive pullups or pulldowns have been announced and demonstrated. The transistors have enough gain to build restoring logic. Recently, all three groups mentioned have announced that they will soon (or already do) fabricate FETs of both n- and p-types, with appropriate threshold voltages and acceptable gain, making it possible to design complementary logic circuits. Although the characteristics of the transistors are still shrouded in mystery, we take as working hypothesis those such devices will be built. We also assume the availability of low-resistance metal wires for the grid direction that provides the gates of transistors. (Such metal wires are obtained either by coating silicon wires with metal during Crystal growth, or by imprinting pure metal wires using the nano-imprint technique.) Let us summarize the components of our (slightly precursory) target technology. As already mentioned, we believe that most properties of this technology will apply to nano-CMOS. The basic building block is the tile. A tile is a crossbar array of nanowires: the (top) horizontal wires are metal conducting wires that are used as gates of transistors; the (bottom) vertical wires are semiconducting silicon
nanowires (NW) used as channels of transistors. Wire length is strictly limited; say around 10μ m, with a wire-to-wire pitch of approximately 10 nm. (Metal wires can be packed more densely and can be longer as they are less resistant. But we will ignore those differences.) We assume that we can implement tiles of 100x100 wires. A tile may be a routing tile or a compute tile. A routing tile contains only programmable junctions and is used to connect two compute tiles. A compute tile consists of an *nplane*, a *p-plane*, and a *routing plane*. An n-plane contains n-transistors only; a p-plane contains p-transistors only; a routing plane contains programmable connections only. An important advantage of this technology is that the different active cross points (transistors, resistors, diodes) fit exactly under the area of the cross point and therefore do not increase the area of the array, hence contributing to the density advantage of the technology. A few important restrictions and properties must be mentioned. (1) It is not possible to mix the different components (n-transistor, p-transistor, and resistor) inside the same plane. Each plane is homogeneous. (2) It is not possible to connect wires end-to-end in one direction. Connections are made through orthogonal wires. (3) field created by applying a voltage on the crossing metal wire. If a dielectric can be placed at the intersection of (and between) the two

Connection resistance is high (100K_) and highly variable. (4) The drive of transistors is not well characterized. We assume that the limit to the number of transistors in series is similar to CMOS. We have chosen 3 as a hard limit. (5) Up to 10% of the wires and connections may be broken or stuck open. (6) Finally, the collection of nano-tiles is placed on top of a standard silicon layer. Power distribution and input/output are implemented in the silicon layer. All vertical nano wires in a p-plane are connected to a micro-wire distributing Vdd, and all vertical nanowires in an n-plane are connected to a micro-wire distributing GND.

III. IMPLEMENTING QDI LOGIC

Among all asynchronous logic families, QDI is the most robust to parameter variations because it relies on the weakest timing assumption: the *isochronic fork*. The main question addressed in this paper is the following. In the presence of extreme PVT (process, voltage, temperature) parameter variations, how will a QDI circuit (implemented in molecular nano) fail, and what can be done to avoid the failure or at least significantly improve the circuit's robustness?

A. Combinational Cells

The implementation of combinational cells does not present any particular problem beside the usual restriction on the length of transistor chains, but it gives us the opportunity to fix a general layout scheme. Figure 1 shows two alternative implementations for the inverter and the nand2. (The nor2 gate is derived from the nand2 by exchanging Vdd and GND, and p- and n-transistors.) A dot at the intersection of two wires indicates a connection; a diagonal bar (/) indicates an n-transistor. A diagonal bar (5) indicates a p-transistor. In the first type of layout, the n- and p-planes are vertically aligned w.r.t. the metal wires; in the second type, they are horizontally aligned w.r.t. the metal wires. It is clear that the second choice of layout presents an important area advantage: the area of the NAND-gate is 6x3 in the first case and 2x3 in the second case. As shown in Figure 2, the area penalty of the first layout for an n-input combinational gate is$2n^{2.}$ But there is another, more important, advantage to the second layout choice. If an input is shared by several transistor gates, then in the second layout choice, the input signals to the different gates are carried by the same metal wire, resulting in very similar RC

characteristics for the different paths. In the first layout choice, the paths from one input to several transistor gates are very dissimilar, with the paths to one type of transistors going through two resistive contacts and the paths to the other type of transistors being a single metal wire. As we shall see, this difference is important for the implementation of isochronic forks.

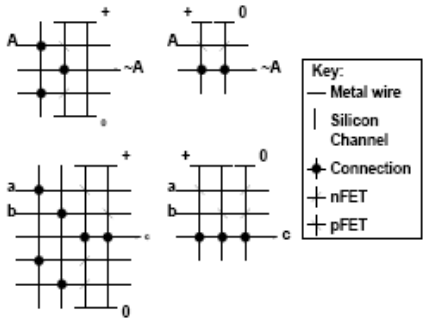
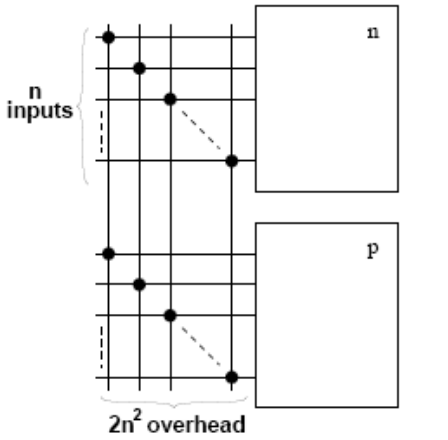


Figure 1. Two possible layouts for the inverter and nand2



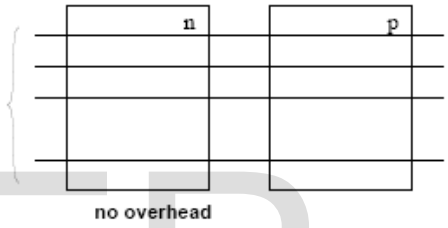


Figure 2. Two possible layouts for an n-input cell

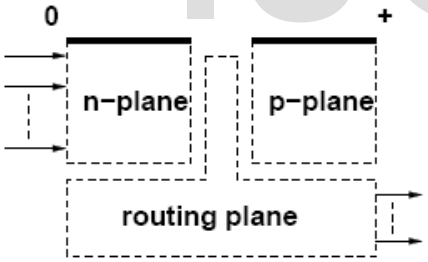


Figure 3. General layout scheme for a compute tile

B. General layout scheme

The general scheme for a compute tile is shown in Figure 3. The two transistor planes are side by side with the input signals running horizontally on metal wires. Each vertical semiconductor wire of the transistor planes can be used (inside a transistor plane) as a pullup or pull down transistor chain if transistors are placed at some of the cross-points.

Because of the restricted geometry, parallel chains cannot share transistors, and therefore all boolean expressions are implemented in the disjunctive normal form: The expression $a \wedge (b \vee c)$ can only be implemented as $a \wedge b \vee a \wedge c$. The general shape of the routing plane is that of an inverted T as shown on the figure. The part of the routing plane between the two transistor planes is used only for the feedback connections needed in the implementation of state-holding gates. (We will give examples shortly.) The bottom part of the routing plane connects the pull up and pulls down chains to the output(s) of the logic gate. (A similar scheme has been proposed in [1].) A single compute tile contains several cells.

IV. Implementing an Adder

We describe the implementation of an n-bit adder stage in nano. The adder is designed in the same style as the one used in the CAM. The basic QDI pipeline

stage $[L?x; R!F(x)]$ can be implemented as in Figure 10 using the Control/data decomposition approach of the CAM. (See, for instance, [7].) As shown in Figure 11, the control part can be implemented as a simple half-buffer (which can be as simple as a C-element in this case); the data path consists of a dual-rail register per bit of input with the use the derived transformation. The isochronicity assumption is that the delay for the valid inputs to propagate from the outputs of the registers to the input of the function blocks should be less than the delay of the adversary path consisting of the *wacks* (in parallel), the completion tree, and the C-element in the control. Furthermore, in order to avoid propagating the control signal en (which is also r0 in this case) to all bits of the data path, we can replace en with ct V cf in all cells but the least significant one, where ct and cf is the pair of bits implementing the carry-in. The nano-layouts for the carry and sum functions are shown in Figure 12 and Figure 13. The productions rules describing the pull up and pull downs of the sum and carry cells are as follows. For the sum:

$$\begin{aligned} ct \wedge (a = b) \vee cf \wedge (a \neq b) &\rightarrow st\_\downarrow \\ cf \wedge (a = b) \vee ct \wedge (a \neq b) &\rightarrow sf\_\downarrow \\ \neg ct \wedge \neg cf \vee \neg sf\_ &\rightarrow st\_\uparrow \\ \neg ct \wedge \neg cf \vee \neg st\_ &\rightarrow sf\_\uparrow \end{aligned}$$

For the carry-out:

$$\begin{aligned} cf \wedge at \wedge bt \vee ct \wedge at \vee ct \wedge bt &\rightarrow dt\_\downarrow \\ ct \wedge af \wedge bf \vee cf \wedge af \vee cf \wedge bf &\rightarrow df\_\downarrow \\ \neg ct \wedge \neg cf \vee \neg df\_ &\rightarrow dt\_\uparrow \\ \neg cf \wedge \neg ct \vee \neg dt\_ &\rightarrow df\_\uparrow \end{aligned}$$

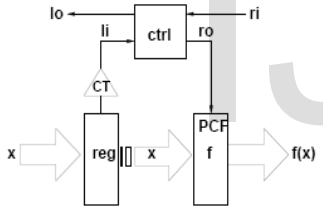


Figure 10. Control/data decomposition of a pipeline stage

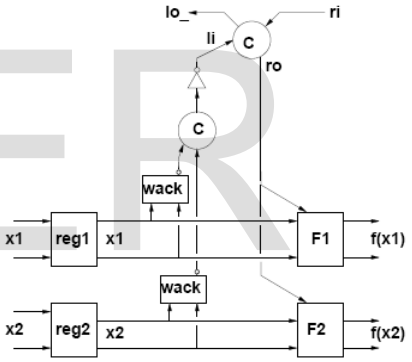


Figure 11. A control/data pipeline stage with half-buffer control

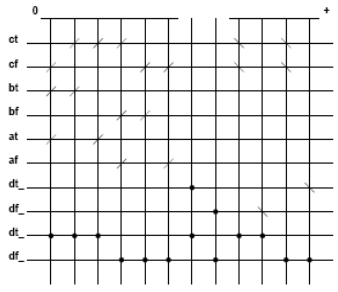


Figure 12. Carry-out computation

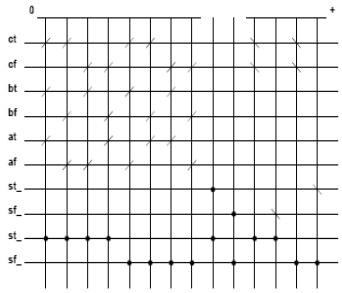


Figure 13. Sum computation

V. CONCLUSION

The version of this template is V2. Most of the formatting instructions in this document have been compiled by Causal Productions from the IEEE LaTeX style files. Causal Productions offers both A4 templates and US Letter templates for LaTeX and Microsoft Word. The LaTeX templates depend on the official IEEEtran.cls and IEEEtran.bst files, whereas the Microsoft Word templates are self-contained. Causal Productions has used its best efforts to ensure that the templates have the same appearance.

VI. REFERENCES

1. Alberto Trombetta and Wei Jiang (2011), 'Privacy-Preserving Updates to Anonymous and Confidential Databases', IEEE Transactions on Knowledge and Data Engineering, Vol. 22, pp. 578-568. **(JOURNAL)**

2. Anderson R (2006), 'A security policy model for clinical information systems', Proceedings of the IEEE Symposium on Security and Privacy, pp. 30-43. **(CONFERENCE)**

3. Charu C. Aggarwal and Philip S. Yu (2008), 'A Survey of Association Rule Hiding Methods for Privacy', A hand book of Privacy- preserving data mining: models and algorithms, Kluwer Academic Publishers, London, pp.32-39. **(TEXT BOOK)**

4. Chen M.S and Yu P.S. (2004), 'Data Mining: An Overview from a Database Perspective', IEEE Transactions on Knowledge and Data Engineering, Vol. 8, 866-883.

5. Dimitris Sacharidis, Kyriakos Mouratidis and Dimitris Papadias (2010), 'k-Anonymity in the Presence of External Databases', IEEE Transactions on Knowledge and Data Engineering, Vol. 22, pp. 392-404.

6. Luo Yongcheng and Le Jiajin (2009), 'Research on Book Circulation Information Publishing Based on Anonymity Techniques', Proceedings of the 1st International Web Information Systems and Mining, pp. 534-537.

7. Machanavajjhala A. and Gehrke J. (2006), 'l-diversity: Privacy beyond k-anonymity', Proceedings of the International Conference on Data Engineering, pp. 56-72.

8. Martin D.J., Kifer D., Machanavajjhala A., and Gehrke, J. (2007), 'Worst-Case Background Knowledge for Privacy-Preserving Data Publishing', Proceedings of the IEEE International Conference on Data Engineering, pp. 126-135.

9. Xiaojun Ye, Yawei Zhang and Ming Liu (2009), 'A Personalized (α, k)-Anonymity Mode', Proceedings of the IEEE International Conference on 9th International Conference on Web-Age Information Management, pp.122-131.

10. Saygin Y., Verykios V. S., and. Clifton C. (2007), 'Using unknowns to prevent discovery of association rules,' Proceedings of the Special Interest Group On Management of Data Record, pp.45-54.