Chapter 6

TECHNOLOGICAL DISPERSIONS TOLERANT ARCHITECTURES

The probability that the technology dispersions is to increase, for both the ultimate CMOS and the electronic components derived from molecular electronics, appears to be very high [NSF01]. The purpose of the methodology we present in this chapter is to suggest ideas as a starting point to be further discussed. By proposing an intermediate method between the standard CMOS design-flow and the revolutionary molecular-electronics bottom-up approach (Section 2), the aim of this methodology is to transfer some of the complexity and the cost needed to control a state of the art technology, toward the development of new architectures and algorithms compensating intrinsic technological dispersions, hopefully at a lower cost.

To achieve this goal, we have adopted a pragmatic approach, combining facts about today’s state of the art nanotechnology fabrication and fairly probable working hypothesis. This reasoning leads to the study of technological dispersion tolerance in highly regular architectures based on molecular-electronics devices. We propose a granular modelization strategy based on high level nano-device modeling. The complexity of the model results of a trade off between reasonable simulation time and sufficient accuracy. The objective is to ensure both the viability of system analysis and the simulation of architectures where each nano-device characteristic can be individually modified. This methodology makes use of a Matlab and VHDL-AMS platform environment developed on purpose.

In the following chapter, we present and compare six architectures implementing different defect tolerant circuits and strategies. Some of these are inspired from existing techniques, others are original. Prior to that, we present the basis of our development platform, based on Matlab and VHDL-AMS, that can generate custom regular memory arrays implementing the previously presented memory cell modeling.
Each cell’s set of parameters is uniquely tailored thanks to a randomly generated variation map that usually follows a Gaussian law. The control blocks surrounding this memory plan are developed with a modular approach in mind and can therefore be adapted to the characteristics of the studied memory map: memory size, word size, number of bits stored per cell, and in some case, number of correction bits per word. Finally, the six evaluated architectures are compared based on various criteria like different error rates, relative speed of operation and relative silicon area.

1. Architecture generation tool

The tool developed under Matlab can be splitted in two independent parts. The first one helps the user to bring in the \( I_{DS}(V_G) \) curve characteristic of a molecular device used in the presented memory cell structure. The second part generates VHDL-AMS code implementing an array of memory cells. It also displays the resulting dispersions of this memory map. Finally, it allows the user to save these information for post-processing. This graphical interface associated to the VHDL-AMS code generator provide a convenient way to quickly evaluate the integration of different types of molecular transistors.

\( I_{DS}(V_G) \) curve characteristic input

The first part of the tool automatizes, to some extend, the input of a \( I_{DS}(V_G) \) electrical characteristic and allows its visual representation (Figure 6.1). Most of the needed information have to be extracted from experimental curves. The real-time drawing of the display allows to verify or to modify accordingly the different parameters characterizing a molecular transistor (Section 1).

The \( K_1 \) and \( K_2 \) constants used in the equations of the molecular NWFET transistor model (See Chapter 4 p. 99) are automatically calculated. Figure 6.1 shows this interface filled with data extracted from literature. The displayed hysteresis cycle is assumed to be the broadest one measured on a specific molecular transistor.

The tool also offers the ability to save and to load files (with a special *.car extension) containing the complete set of fields related to a specific molecular transistor as well as all parameters of the generated memory (Options button). The Generate button starts the process of generating VHDL-AMS code instantiating a memory map where electrical variations are randomly spread over both the access and the memory transistors.

Thanks to this approach, one can actually really start concentrating on the circuit and on the architecture level. At some point in time, all figures can be refined when extra experimental data become available.