CONAN - A Design Exploration Framework for Reliable Nano-Electronics Architectures

S. Cotofana[1], A. Schmid[2], Y. Leblebici[2], A. Ionescu[2], O. Soffke[3], P. Zipf[3], M. Glesner[3], and A. Rubio[4]

[1]Delft University of Technology, Delft, The Netherlands. [2]Swiss Federal Institute of Technology, Lausanne, Switzerland. [3]Darmstadt University of Technology, Darmstadt, Germany. [4]Polytechnic University of Catalonia, Barcelona, Spain.

Abstract

In this paper we introduce a design methodology that allows the system/circuit designer to build reliable systems out of unreliable nano-scale components. The central point of our approach is a generic (parametrical) architectural template, COnfigurable Nanostructures for reliAble Nano electronics (CONAN), which embeds support for reliability at various levels of abstractions. Some of the main reliability sources are regular and decentralized structures based on simple basic computation cells designed to be robust against disturbances and noise, fault tolerance based on hardware, time and information redundancy applied at the basic cell level as well as at higher levels, self diagnosis assisted by the dynamic reconfiguration of basic computation cells and interconnect rerouting. Within the CONAN template both technology dependent and independent models co-exists such that the more abstract layers are technology independent while the lower levels can be retargeted to various fabrication technologies. Our proposal is application-oriented and allows the designers to deal with unpredictability, and low reliability, which are unavoidable characteristics of future emerging nano-devices. When combined with the underlying software, the tools supporting the CONAN approach allow the designer to check whether the design constraints are fulfilled before performing a detailed implementation and provides means to trade area, delay, and power consumptions for reliability. As such, this proposal is a call-to-arms to mobilize the efforts of systems designers in order to achieve a systematic design methodology for reliable systems.

1. Introduction

The integrated circuits invention and the manufacturing progress reached nowadays are the fundamental engines for the implementation of all the technologies that support today's information society. The pushing effort behind this progress consists in the miniaturization of devices allowing millions of transistors in a single silicon piece working at frequencies of gigahertz. As the miniaturization trend approaches the physical limits of operation and manufacturing, the characterization of the devices and circuit parameters becomes increasingly hard and even unpractical, with a lack of efficient solutions [10]. Future computing technologies (non silicon based) are envisaged to enable the design of systems with a much higher density of devices than the one today, allowing unprecedented new products and services.

Due to the foreseeable limitations of silicon-based technology and the promising results of new devices of different nature working at nanometer level, there is a worldwide attention to the research and development of new electronic devices that could be the base of this future technology.

The unprecedented amount of computational power these new technologies are expected to permit will be useful only if new design methodologies are available [1]. The main reasons for this are the huge complexity of such systems and the high number of defective components that will unavoidably come along with the introduction of emerging and future technologies.

Consequently, the expected panorama of future electronic system design methodologies corresponds to a massive use of components, orders of magnitude higher than today, with component reliabilities orders of magnitude lower than today. This represents a new, challenging and essential problem.

Nowadays the strategy of design is based on the hierarchical characterization of several levels of abstraction, from device to architectural high level, with intrinsic verification methods and tools for each level. This allows the treatment of large circuits at different abstraction and complexity levels. In this scenario the designer assumes that final systems will be composed of perfect or acceptably correct components. Designers are only aware of a potential defect through the use of design for testability rules, tools and standards, in order to make simple and efficient the last test manufacturing control stage that separates good and bad circuits.

While the vast majority of the recent nanoelectronics-related research efforts are concentrated on the development of new nano-materials and devices, very little has been done into the direction of design methodologies for circuits and systems using such emerging technologies. The main reasons behind this trend are (i) the perception that the novel device technologies are still immature to justify any exploration of design methodologies, and (ii) the assumption that once the new devices are available, one can utilize well-known design paradigms, methodologies and tools in a straightforward manner to develop circuits and systems.

It is a well-known fact, however, that historically, each new device technology has led to the development of new design methodologies that match and exploit the specific characteristics of the corresponding technology. At the same time, some of the novel nano-scale technologies have already reached a sufficiently stable stage at which accurate predictions can already be made about their influence on design, and their system-level exploitation. Up to date there is enough evidence that many of the emerging devices exhibit a behavior that is fundamentally different from that of the traditional (C)MOS devices, and this makes the utilization of current design paradigms not very effective. Moreover the emerging technologies (we include in this category the sub-100-nanometer MOS devices, too) bring a new aspect into design - namely unreliable components that exhibit a certain level of unpredictability. Emerging and future devices exhibit dimensions in the order of the de-Broglie wavelength of electrons. Therefore, their behavior is dictated by quantum physics and these devices will most likely be unreliable by their nature and circuits made of them will be certainly very susceptive to disturbances and noise [9]. Thus it is quite clear that future computers with nanoscale components will certainly contain a number of defects. This reality supports a new approach in which *architectural issues and defect tolerance have to be considered in very early design stages*.

Up to date there are no systematic approaches for designing circuits and systems with the novel nano-scale and sub-100-nanometer CMOS devices. The designer mainly counts on ad-hoc solutions that are mostly based on increasing the pressure on the fabrication technology to produce “perfect” devices. While this might still be an option for CMOS for a while, it does not seem to be feasible in the case of novel technologies. Moreover the “perfect” device can become prohibitivpenely exsive thus not an option for large circuits and systems due to market related reasons.

Given the previously mentioned facts the main objective of the current paper is to introduce a generic methodology, COnfigurable Nanostructures for reliAble Nano electronics (CONAN), which allows the system/circuit designer to build reliable systems out of unreliable components. In this line of reasoning we propose a design paradigm that can deal with device unreliability by inducing fault-, defect- and error-tolerance approaches at various levels of abstraction starting from the device level up to the system architecture level. These approaches are not limited to the classical ones, but include also new solutions, which exploit the characteristics of a given technology. Nevertheless, device physics forms a transparent layer for designers using the proposed design methodology.

While we refrain ourselves from presenting specific results in this paper, our main goal is to explore the feasibility of a complete design framework that will eventually lead to systematic reliable design. In this context, the manuscript is better interpreted as a “call-to-arms” to reach a unified design methodology.

The rest of the presentation is organized as follows: Section 2 presents the basic ideas behind the CONAN methodology. Section 3 describes the hierarchical organization of the abstraction levels at which fault-, defect- and error-tolerance can be induced and Section 4 draws some conclusions.

2. The CONAN Design Methodology

The central point of our proposal is a generic (parametrical) architectural template that embeds support for reliability at various levels of abstraction. Some of the main reliability sources we considered are regular and decentralized structures based on simple basic computation cells designed to be robust against disturbances and noise, fault tolerance based on hardware, time and information redundancy applied at the basic cell level as well as at higher levels, self-diagnosis assisted by the dynamic reconfiguration of basic computation cells and interconnect rerouting.

We note here that the basic computational cells have to be designed in such a way that apart of providing robustness they effectively utilise the potential of the target technology, thus, by definition, they are not standard computational elements such as Boolean gates. Within this hierarchical template both technology dependent and independent models coexists such that the more abstract layers are technology independent while the lower levels can be retargeted to various fabrication technologies.

The underlying idea of the design paradigm we propose is to associate a design methodology and a design exploration framework to a generic architectural template that embeds support for reliability such that given a certain fabrication technology and an application we assist the system/circuit designer in her/his quest for the most appropriate implementation. In this context the designer is given in the top of the standard design tradeoffs also the possibility to trade reliability for area and other performance figures.

Even though in the CONAN paradigm the end user is practically unaware about the particular features of the utilized fabrication technology we do not propose a technology independent approach. The CONAN framework embeds realistic fault models and system-level yields estimations for the technology dependent part. Related to this part, mechanisms performing retargeting of the design exploration framework on a characterized emerging technology are provided.

The design scenario associated with our approach can be sketched as follows:

1. Choose a fabrication technology and retarget the design exploration framework accordingly. This action is not primarily meant to be done by the end user and might imply some major modifications (new models) in the technology dependent part when the technology type change is addressed. However when the technology change is occurring within the same family the end user can operate the retargeting by changing the parameters of the technology specific models.

2. Assume an application and the design constraints in terms of area, power, and reliability requirements associated with it.

3. Instantiate an underlying architecture for the given application and evaluate its potential performance in terms of area, power, and reliability.

4. If the performance is acceptable, proceed with the detailed implementation; otherwise perform some design tradeoffs, and go to step 3.

This procedure allows the designer to check whether the design constraints are fulfilled before performing a detailed implementation and provides means to trade area, delay and power consumptions for reliability. Moreover, if various flavors (different price and different device reliability) of certain technologies are available, the designer can identify the most effective implementation in terms of costs. For the same achieved reliability a larger implementation in a less expensive technology might be in certain circumstances more attractive than a smaller one in a more expensive technology.

The proposed application-oriented methodology forms an original initiative, which allows designers to deal with unpredictability and low reliability, which are unavoidable characteristics of emerging and future devices. Our methodology, together with the underlying software tools supporting the approach, targets a significant improvement of the system reliability, in face of the widely recognized fact that the nano-scale and sub-100-nanometer CMOS device reliability will be dramatically lower than that of current technologies. Our proposal has a number of advantages as follows.

It allows for a systematic design space exploration. In this way given an application, a technology and some design constraints including reliability requirements, various design alternatives can be exercised and evaluated without the need for a complete technology mapping.

The designer is given the opportunity to trade reliability for other performance figures. In this way one can target a certain acceptable error rate and get the corresponding area/delay/power or target an area (price) and get the corresponding error rate. This may be very beneficial as in many applications, e.g., computer graphics, speech processing, where a certain level of error is quite acceptable as it has no visible/audible implications in the quality of the produced picture/sound we sense. Moreover, as the acceptable error rate depends on the price class this kind of tradeoffs are useful for developing solutions for the same application but in different price levels.

The pressure on the fabrication technology to produce “perfect” devices can be relaxed. This has economic implications, as less perfect devices should be less expensive. Additionally, the designer might investigate various design tradeoffs and choose a solution based on an inexpensive process (less reliable devices) when an acceptable reliability can be achieved at the area expense.

The application mappings produced by the design exploration process out of the architectural template has increased fault tolerance and operational robustness as it can deal with permanent faults induced by manufacturing faults as well as with transient faults that may appear during functioning of the system.

The design framework we propose is retargetable thus if the appropriate models are available it can be adapted to any advanced nano-CMOS or other nano-technologies.

3. The CONAN Hierarchy

Our approach follows the conventional method of design hierarchy, embedding however, the reliability concern in the different hierarchical levels. The basis of such a design methodology is presented in Section 2.

In our methodology we utilize a hierarchy of abstraction levels in order to allow coping with the complexity of systems, reliability constrains as well as error tolerance when possible.

Figure 1 shows a graphical representation of the CONAN design methodology. As one can observe in the figure it is a layered approach, i.e., we propose dealing with reliability at different levels of abstraction. The different layers represent different levels of abstraction and are explained in the following sections.

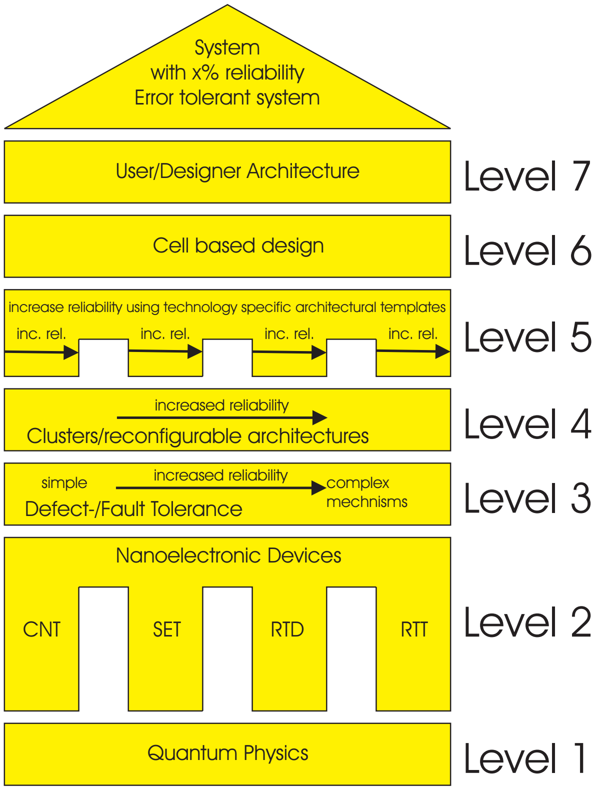


Figure 1: Overview of the CONAN Hierarchy

3.1. Nanoelectronic Devices

Levels 1 and 2 are dedicated to device modeling. It is well understood that emerging and future devices cannot be modeled by the use of classical semiconductor physics anymore. This is because at small dimensions the energy (and therefore also the momentum and the de-Broglie wavelength) of particles, i.e., electrons, is quantized. The device behavior is therefore described by the Schrödinger equation, which is the basis of quantum physics. All nanoelectronic (and very advances CMOS) devices are based on quantum physics and these two lowest levels of the CONAN hierarchy are dedicated to such models.

3.2. Fault and Defect Tolerance at Basic Gate Level

The third level is dedicated to fault and defect tolerance techniques at basic gate level. Several techniques are known to deal with faults at low levels of abstraction. At current technologies, such faults occur most likely due to manufacturing. Therefore, mainly permanent faults are addressed. It is common practice to test the circuits after production and discard faulty ones. The concept of error tolerance, i.e., accepting a certain amount of errors, can significantly increase the yield in these cases. However, another approach is to add spare modules to the circuit, which can be chosen after production. Therefore, circuits need not be discarded, but can be (permanently) configured in order to get circuits which are fault free. Again, the concept of error tolerance is applied leading to even higher yield. There are different kinds of applications, which require different degrees of reliability. So, there is a yield-reliability trade-off, e.g., a microprocessor will exhibit less yield than a digital signal processing unit. In addition, the effort for fault tolerance can be varied for given yield and given reliability, e.g. more spare modules can be used.

Besides the permanent faults, which occur mainly during production, nanoelectronic circuits are very sensitive to disturbances and noise. Current digital circuits do not suffer from noise under normal operating conditions. This looks different for space applications devices which are exposed to radiations outside the protecting ionosphere of the earth. If a particle hits a digital integrated circuit, a register can accidentally change its state. The same effect can also cause the voltage levels at the output of a combinational gate or on interconnects to be changed temporarily. This is usually referred to as *single event upset* (SEU).

The use of very advanced CMOS and/or nanotechnology requires dealing with such effects in every design because the sources of disturbances are not limited to particles any more. In fact, thermal noise at room temperature may even cause an SEU, or prevent the output of a combinational block to be sampled correctly by the subsequent register, at the rising or falling edge of the clock.

The necessity to cope with intrinsic errors at the device and circuit level must be recognized as a key aspect of nano-scale systems design. To implement such robustness and fault tolerance, new circuit design approaches need to be considered at the low level. Many successful logic applications have been reported by mimicking CMOS, but real competing performance with CMOS still remains to be demonstrated [12], [15], [16]. Typically, the widely applied triple modular redundancy with majority voting will fail to guarantee safe operation of nanoelectronic devices, which are expected to suffer from high defect density [17]. New concepts in the design of logic systems will play a dominant role in the development of large nano-systems.

For example graceful degradation of system performance, adaptability of the redundancy factors at several levels of abstraction to the desired probability of correct operation, as well as the application of new design-styles have to be addressed.

A fault tolerant architecture consisting of four layers in which the data is strictly processed in a feed-forward manner has already been considered and is depicted in Figure 2 [18], [19].

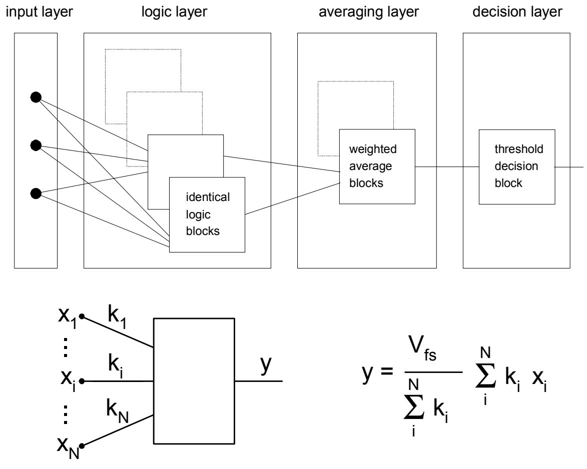


Figure 2: A fault tolerant architecture consisting of four layers. Note that these four layers are all sub-layers of Layer 3 of the CONAN design methodology.

The first layer is denoted as the input layer, accepting conventional Boolean (binary) signal levels. The core operation is performed in the second layer, which consists of a number of identical, redundant units implementing the desired logic function. The fault immunity increases with the number of redundant units, yet the operation is quite different from the classical majority-based redundancy. In contrast to classical n-tuple redundancy, the proposed architecture is expected to be significantly more immune to multiple device failures, in the form of stuck-on or stuck off faults. The third layer receives the outputs of the redundant logic units in the second layer, creating a weighted average with re-scaling. Note that the output of the third layer becomes a multiple-valued logic level. Finally, the fourth layer is the decision layer where a binary output value is extracted using a simple threshold function. It was already shown in the literature that this particular type of weighted-sum functions could be implemented quite easily with SET devices. Similarly, proposals have been made to exploit the particular characteristics of SETs for the implementation of multiple-valued logic functions.

A regular programmable logic array (PLA) of unit building blocks is adapted to provide fault tolerance capability in the second layer using SETs or nanometer CMOS devices [20]-[22]. The PLA is used for performing a programmable NOR Boolean operation of its inputs. The structure of the array is made from one unit cell being replicated in the vertical direction to form the logic function as a slice. A number of slices are appended in the horizontal direction and share the same input variables to be connected to the data inputs. In our case, the Boolean function input variables can be modified via soft programming using programming inputs. Dramatic failures modeled as stuck-on or stuck-off errors can also be simulated using the same programming scheme.

Programmability of the switches granting access to the averaging units allows redundancy factors of two, three or four for each logic function. It has been shown that the proposed four-layer architecture has the capability of absorbing errors which occur with a high-density pattern much more efficiently than majority voting schemes usually applied, even with a low redundancy factor, typically two or three.

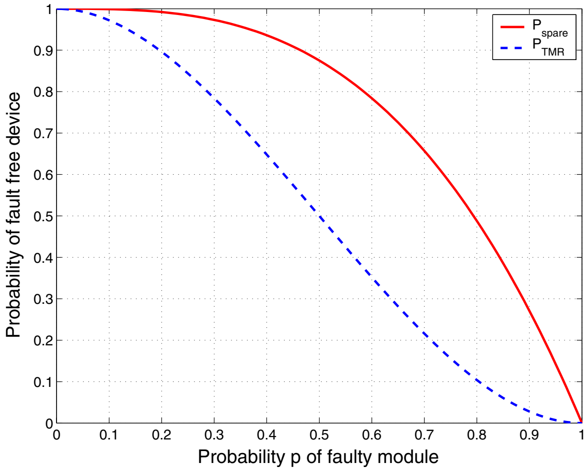
Regular array structures including functional redundancy is coupled to adapted fault tolerant architectures at circuit-level, reprogrammability and reconfigurability offer a very versatile solutions to the reduced yield expected to affect future nanometer-scale devices.

However, some concepts of the classical fault tolerance and information theory are revisited and adapted accordingly in order to come to new concepts usable for future technologies. Nevertheless, the degree of unreliability is much higher for nanoelectronic devices, than for classical ones. Therefore, the existing methods have to be investigated very carefully in order to judge their usability for these future technologies.

Very promising methods are error correcting codes and soft-bits known from channel coding in digital communication systems. By soft-bits, we mean in this case: all signals do not have only the value ‘0’ or ‘1’ but also intermediate values are allowed, e.g., (0.6 NAND 0.2) = 1. These intermediate values can be represented using several bits for each signal. Also a combination of error correcting codes (e.g., Hamming codes) and soft-bits is possible, i.e., each soft-bit is protected by the use of a Hamming code.

It makes sense to use different approaches for permanent and transient faults: at the first glance one could expect, that the use of triple modular redundancy for example (which is not advisable for high defect densities and is only used for clarification purposes here) also protects against permanent faults. This is of course true, but given a certain probability p of a module to be faulty, the probability P to get a fault free device is higher with the use of two spare modules (i.e. three modules in total) than with the use of TMR, although the same number of modules are used in total [5]:

$$\begin{aligned} P_{spare} &= 1 - p^3 \\ P_{TMR} &= (1-p)^3 + 3p(1-p)^2 \end{aligned}$$



Note that P_{spare} is greater than P_{TMR} for all values of p. This analysis also shows clearly why TMR is not applicable for high defect densities: if the probability p exceeds 50%, the probability to get a fault free device is even less than without any redundancy [5].

All fault and defect tolerance approaches require that there is a possibility to test the circuit accordingly. Especially dealing with dynamic faults make on chip runtime testing necessary. These test units can easily be extended in order to achieve also on chip testing of production faults and automatic chip configuration to meet the reliability constraints. Note that the concept of error tolerance is applied here, too, i.e., the ultimate goal is not to build 100% reliable circuits but circuits that exhibit a certain degree of reliability.

3.3. Clusters, Regular Structures and Reconfigurable Architectures

Beyond the classical methods for fault and defect tolerance, the building blocks are clustered in order to get the possibility to reconfigure the circuit in case of permanent faults (either due to manufacturing or due to later damages) or in case of slowly varying faults. Another benefit of clustering is the filtering of fluctuations due to quantum behavior as well as the inherent profit of statistical average parameters when wide process variability appears. This is depicted in Figure 3. Again, a possibility to adjust the effort to the desired error rate is provided.

The use of regular structures allows for reconfiguring the "nanocluster" to react on faults, because the faulty subcell can easily be excluded from the device with simple rerouting. The routing algorithm is comparatively simple because this kind of reconfiguration is performed locally. This simplicity is necessary in order to be able to implement the required algorithms in hardware. With that approach, it is possible to reconfigure the circuit at runtime. With this technique, reliable nanoclusters, with a high isolation of the technology peculiarities, can be used at the cell and architecture levels.

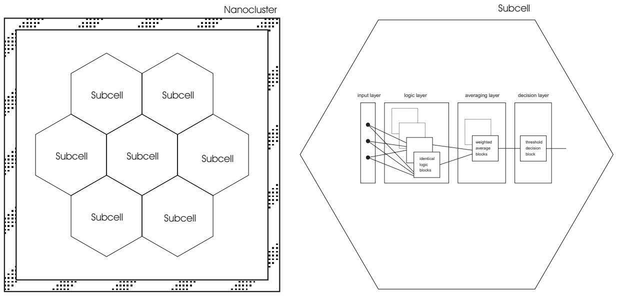


Figure 3: The basic gate cells protected with the approaches described in Level 3 are clustered to regular (locally) reconfigurable (micro-) architectures leading to more reliable "nanoclusters" which can be used in the Level 5 architectures.

An example where this reconfiguration approach can be used concerns the *background charge effect* [12, 15] that is considered one of the main drawbacks of the Single Electron Devices. In fact, the SET characteristics are extremely sensitive to any discrete charge placed in the device proximity, which could result in a very significant shift of their oscillating characteristics (even much-smaller than unit background charge could be very detrimental to Single Electronics). Moreover, it is expected that a random parasitic charge distribution at chip level would result in random I-V characteristics of individual devices. There are different ways to deal with such an effect that could have dramatic consequences for logic or memory applications. A first approach is to envision device-level solutions: for instance, one can design an SET with tunable gate capacitance (resulting in a NEMSET type of device) and/or a feedback loop that could locally compensate for any parasitic background charge. This solution involves extremely high and risky technological developments, as well as complex architectures. On the other hand, a more elegant high-level solution that *reconfigures* the structure in order to compensate for the fabrication defects can be foreseen for such case and is considered among the priorities of the proposed methodology.

3.4. Technology Dependent Architectural Templates

Studying the physical behavior of the nanoelectronic devices one can develop architectural templates for the individual kinds of devices. This can be explained in more detail using an example: the delay t_d of a SET-device depends on the error probability P_{error}, that means the error probability is the probability that a desired tunneling event did not take place after the time t_d. This relationship is given by [11]:

$$t_d = -\frac{\ln(P_{error}) \cdot e \cdot R_t}{|V_j| - V_c}, \text{ with } |V_j| > V_c$$

where e is the unit charge, R_t represents the tunnel resistance, V_j is the voltage across the tunnel junction and V_c is a critical voltage that has to be exceeded for a tunneling event to take place.

Therefore, P_{error}, or in other words, the probability that the desired electron transport did not take place after the time t_d decreases exponentially with t_d, i.e., the longer one waits the less P_{error}. This can lead to the architecture depicted in Figure 4 for SET devices.

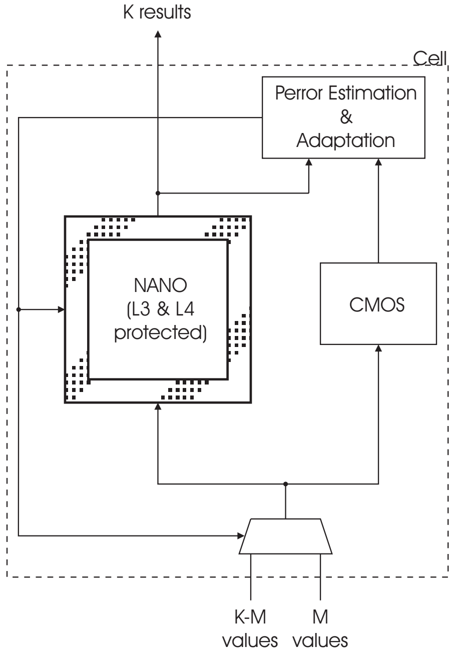


Figure 4: Architecture to deal with the relationship between error probability and delay. Such architecture is used in each computational cell.

Assume a block in the data path of a nanoelectronic system. This block has to do K (>>1) operations with a desired $P_{error\_desired}$. First of all, N times M<<K operations are done using this block and an additional block, built in conventional CMOS, is used to do the same M operations (this takes the same time like the N*M operations done by the nano block take, because the nano block is assumed to be N times faster) in order to estimate P_{error}. This procedure can be repeated i times to find the correct timing for the nanoelectronic part. After this training period, the nano block performs the remaining K-M operations with the speed found by the adaptation process. Assuming that the nano block with optimized timing is still much faster than the corresponding CMOS block, the overall time for the K operations is significantly less than the time the CMOS would need for the K operations if K is great enough.

It is possible to have several such architecture templates for one type of technology: different ranges of P_{error} and different kinds of applications (µP, DSP, etc.) may lead to different architectures. Such sets of architecture templates are specific for the different nanoelectronic technologies like SET/CNT, RTD/RTT, etc. The same applies for different optimization criteria. The use of hybrid devices, i.e., devices that consists of a classical MOSFET as well as a SET, allows for the design of ultra low power systems but the speed is in the same order of magnitude as the speed of CMOS. An architecture similar to the one presented in Figure 4 can be developed to take advantage from the ultra low power capabilities of SET devices in addition to the exploitation of the features of MOS devices. To be crystal clear, the key concept here is to combine classical CMOS and emerging and future devices on a single chip in order to take advantage from the best of both worlds. The example with hybrid devices is a very interesting approach from this point of view: MOS and SET devices are combined at *several levels of abstraction*. The hybrid device itself consists of SET and MOS transistors. In addition the Level 5 architecture, combining hybrid and MOS devices at high levels of abstraction, is used to compensate for the unreliability originated in the use of nanoelectronic devices.

3.5. Classical Design Flow

Using the design flow up to here it is now possible to define cells, which can be used in a standard-cell-design like manner in order to design the system. Here we are back at the well-known classical design flow, which is mostly technology independent, but the system made of these cells has the desired reliability.

Of course the "standard cells" used for this approach are much larger than conventional ones, i.e. they realize much more complex functions. Therefore, the designer has to be able to design his/her own cells. It can be clearly seen from the description above that this is a rather complex task. Thus, the support by CAD tools is needed: developing this CAD tools and algorithms is among many other things future work subject.

4. Conclusions

In this paper we have introduced a design methodology that allows the system/circuit designer to build reliable systems out of unreliable nano-scale components. The central point of our approach is a generic (parametrical) architectural template, COnfigurable Nanostructures for reliAble Nano electronics (CONAN), which embeds support for reliability at various levels of abstractions. Some of the main reliability sources are regular and decentralized structures based on simple basic computation cells designed to be robust against disturbances and noise, fault tolerance based on hardware, time and information redundancy applied at the basic cell level as well as at higher levels, self-diagnosis assisted by the dynamic reconfiguration of basic computation cells and interconnect rerouting. Within the CONAN template both technology dependent and independent models co-exists such that the more abstract layers are technology independent while the lower levels can be retargeted to various fabrication technologies. Our proposal is application-oriented and allows the designers to deal with unpredictability, and low reliability, which are unavoidable characteristics of future emerging nano-devices. When combined with the underlying software tools supporting the CONAN approach, the designer is supported to check whether the design constraints are fulfilled before performing a detailed implementation and provided means to trade area, delay, and power consumptions for reliability.

5. References

[1] G. De Micheli, "Robust system design with uncertain information", Proceedings of the First ACM and IEEE International Conference on Formal Methods and Models for Co-Design, pp. 12, 2003.

[2] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components", Automata Studies, ed. C.E. Shannon and J. McCarthy, Princeton University Press, pp. 43-98, 1956.

[3] W.H. Pierce, "Failure-Tolerant Computer Design", New York Academic Press, 1965.

[4] N. Pippenger, "On networks of noisy gates", Annu. Symp. on Foundations Comput. Sci., IEEE Computer Society Press, pp. 30-38, 1989.

[5] M.A. Breuer, S.K. Gupta and T.M. Mak, "Defect and error tolerance in the presence of massive number of defects", *IEEE Design and Test of Computers*, May-June 2004, pp. 216-227.

[6] International Technology Roadmap for Semiconductors, 2003; http://public.itrs.net/Files/2003ITRS/Home.htm

[7] D. Appello et al., "Understanding yield losses in logic circuits", *IEEE Design and Test of Computers*, May-June 2004, pp. 208-215.

[8] G. Bourianoff, "The future of nanocomputing", *IEEE Computer*, August 2003, pp. 44-53.

[9] N. Pippenger, "Reliable computation by formulas in the presence of noise", *IEEE Tr. on Information Theory*, March 1988, pp. 194-197.

[10] J.R. Heath et al., "A defect tolerant computer architecture: opportunities for nanotechnology" *Science*, 280, p1716, 1998.

[11] C. Lageweg et al., "Single Electron Encoded Latches and Flip-Flops", *IEEE Transactions on Nanotechnolog*y, Vol. 3, No. 2, June 2004.

[12] K. K. Likharev, *Single-electron Devices and Their Applications*, Proceedings of the IEEE, Volume: 87, Issue: 4, pp. 606–632, April 1999.

[13] K. Yano, T. Ishii, T. Sano, T. Mine, F. Murai, T. Hashimoto, T. Kobayashi, T. Kure and K. Seki, *Single-Electron Memory for Giga-to-Tera Bit Storage*, Proceedings of the IEEE, Volume: 87, No. 4, pp. 633-651, April 1999.

[14] A.M. Ionescu, S. Mahapatra, V. Pott, "Hybrid CMOS-SET Architecture with Coulomb Blockade Oscillations and High Current Drive", IEEE Electron Device Letters (EDL), Volume: 25, Issue: 6, pp. 411-413, June 2004.

[15] C. Wasshuber, *Computational Single Electronics*, Springer Verlag, New York, 2002.

[16] M. Kirihara, N. Kuwamura, K. Taniguchi, and C. Hamaguchi, "Monte Carlo study of single-electronic devices," in *Ext. Abst. Int. Conf. on Solid State Devices and Materials*, Yokohama, Japan, 1994, pp. 328–330.

[17] K. Nikolic, A. Sadek, M. Forshaw, "Architectures for Reliable Computing with Unreliable Nanodevices," *Proc. 1st IEEE Conference on Nanotechnology IEEE-NANO*, 2001.

[18] A. Schmid and Y. Leblebici, "Robust Circuit and System Design Methodologies for Nanometer-Scale Devices and Single-Electron Transistors," accepted for publication in the Special Issue on Nanoelectronics - IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 12, No. 11, pp.1156-1166, Nov. 2004.

[19] A. Schmid and Y. Leblebici, "A Highly Fault Tolerant PLA Architecture for Failure-Prone Nanometer CMOS and Novel Quantum Device Technologies," The 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Cannes, France, October 2004.

[20] A. DeHon, "Array-Based Architecture for FET-Based, Nanoscale Electronics," Proc. IEEE Nanotechnology, Vol. 2, No. 1, March 2003, pp. 23-32.

[21] M. M. Ziegler, M. R. Stan, "CMOS/Nano Co-Design for Crossbar-Based Molecular Electronic Systems," Proc. IEEE Nanotechnology, Vol. 2, No. 4, December 2004, pp. 217-230.

[22] A. Schmid and Y. Leblebici, "Array of Nanometer-Scale Devices Performing Logic Operations with Fault-Tolerant Capability," Fourth IEEE Conference on Nanotechnology IEEE-NANO, München, Germany, August 2004.