Towards Nanocomputer Architecture

Paul Beckett, Andrew Jennings
School of Electrical & Computer Systems Engineering
RMIT University
PO Box 2476V Melbourne, Australia
pbeckett@rmit.edu.au, ajennings@rmit.edu.au

Abstract

At the nanometer scale, the focus of micro-architecture will move from processing to communication. Most general computer architectures to date have been based on a "stored program" paradigm that differentiates between memory and processing and relies on communication over busses and other (relatively) long distance mechanisms. Nanometer-scale electronics – nano-electronics - promises to fundamentally change the ground-rules. Processing will be cheap and plentiful, interconnection expensive but pervasive. This will tend to move computer architecture in the direction of locally-connected, reconfigurable hardware meshes that merge processing and memory. If the overheads associated with reconfigurability can be reduced or even eliminated, architectures based on non-volatile, reconfigurable, fine-grained meshes with rich, local interconnect offer a better match to the expected characteristics of future nanoelectronic devices.

Keywords: computer architecture, nanocomputer architecture, micro-architecture, nanoelectronic technology, device scaling, array architecture, future trends, QCA, SIMD, MIMD.

1 Introduction

Computer designers have traditionally had to trade the performance of a machine for the area occupied by its component switches. However, when the first practical "nano" scale devices - those with dimensions between one and ten nanometers (10 to 100 atomic diameters) - start to emerge from research laboratories within two or three years, they will mandate a new approach to computer design. Montemerlo et al (1996) have described the greatest challenge in nanoelectronics as the development of logic designs and computer architectures necessary to link small, sensitive devices together to perform useful calculations efficiently. Ultimately, the objective is to construct a useful "Avogadro computer" (Durbeck 2001) - one with an architecture that makes efficient use of in the order of 10^{23} switches to perform computations. In the more immediate term, it is forecast that by 2012 a CMOS (or possibly SiGe) chip will comprise almost 10^{10} transistors and will operate at speeds in the order of 10 - 15GHz (IST 2000).

The design challenges will be formidable. For example, amongst a long list of major technical difficulties, the SIA roadmap (which refers particularly to CMOS) identifies the following major issues (SIA 1999):

- power management at all levels;
- new architectures to overcome bottlenecks at interconnects;
- ultimate short channel limitations (e.g. at 30nm) requiring more complex gate structures such as SOI or dual-gate transistors;
- the spiralling costs of both lithography and fabrication.

In addition to the fundamental problems caused by high power density (Borkar 1999), physical problems such as leakage, threshold voltage control, tunnelling, electro-migration, high interconnect resistance, crosstalk and the need for robust and flexible error management become significant as device features shrink (Montemerlo et al 1996). These problems, in turn, affect the way that devices may be connected together and will ensure that the performance of future architectures will come to be dominated by interconnection constraints rather than by the performance of the logic (Ghosh et al 1999, Timp, Howard and Swami 1999).

It is likely, therefore, that the physics of nanoelectronic devices will conspire to eliminate the classical stored-program (von-Neumann) architecture as a contender at nanoelectronic device densities. This organisation, which has driven the development of computer architecture for almost 50 years, differentiates between the functions of memory and logic processing and tends to be built around system-wide constructs such as busses and global control signals. It is hard to imagine how any form of globally connected stored-program architecture could be built in a technology where communication even between adjacent switches is difficult.

Nevertheless, if the progress implied by Moore's Law is to continue (Borkar 2000), nanocomputer architectures must eventually supersede conventional, general-purpose microprocessor machines. They will therefore need to perform the same functions as their predecessors as well as sharing many of their overall characteristics. They will (ideally) need to be small, fast, cheap and robust, work at room temperature and run code from a standard compiler, including legacy code. This legacy requirement is often overlooked. It is likely that computing functions will

continue to be described in terms of software with its inherently linear control flow. General purpose computing is dominated by control dependencies and tends to rely on dynamic data structures (Mangione-Smith and Hutchings 1997). How the temporal "control-flow" and dynamic data allocation of such a software description might be mapped efficiently onto the hardware circuits of a nanocomputer is not yet clear. Margolus (1998) offered one vision when he forecast that *"…our most powerful large-scale general purpose computers will be built out of macroscopic crystalline arrays of identical … elements. These will be the distant descendants of today's SIMD and FPGA computing devices: … architectural ideas that are used today in physical hardware will reappear as data structures within this new digital medium"*.

This paper will discuss the major issues that will influence computer architecture in the nanoelectronic domain. The paper is organised as follows: section 2 covers the problems of device scaling and how the characteristics of nanoelectronic devices will constrain future architectural development. In Section 3 we look at a small selection of novel architectures that have been developed to deal with these constraints. Finally we speculate on some paths forward for nanocomputers that can accommodate the legacy code requirements.

2 Scaling Limits of CMOS

CMOS has been the work-horse technology in commercial VLSI systems for about 10 years, after superseding nMOS in the early 1990's. During that time, transistor channel lengths have shrunk from microns down to today's typical dimensions of 150 to 180nm (Gelsinger 2001) and are certain to further scale to 70-100nm in the near future. Such devices have already built on research lines – for example by Asai and Wada (1997), Taur et al (1997) and Tang et al (2001) - and these experiments have demonstrated that mass-production is possible.

In order to contain an escalating power-density and at the same time maintain adequate reliability margins, traditional CMOS scaling has relied on the simultaneous reduction of device dimensions, isolation, interconnect dimensions, and supply voltages (Davari 1999). However, FET scaling will be ultimately limited by high fields in the gate oxide and the channels, short channel effects that reduce device thresholds and increased sub-threshold leakage currents (McFarland 1997). As a result, Davari has suggested that gains in FET device performance will eventually stall as the minimum effective channel length approaches 30nm at a supply voltage of 1.0V and a gate oxide thickness of about 1.5nm. Beyond this point, any further performance growth will need to rely on increased functional integration with an emphasis on circuit and architectural innovations.

2.1 Defect and Reliability Limits

The probability of failure for transistors in current CMOS manufacturing processes range from 10^{-9} to 10^{-7} (Forshaw, Nikolic and Sadek 2001) and it appears certain that currently available processes will not be suitable for providing defect-free device structures at sub-100nm scales (Parihar, Singh and Poole 1998). Thus any architecture built from large numbers of nanoscale components will necessarily contain a significant number of defects. An understanding of the role of these defects and how they affect yield will be important to future architectures. Novel low-temperature, 3-D integrated manufacturing technologies such as that proposed by Ohmi et al (2001) might eventually result in reliable, defect-free, high-performance gigahertz-rate systems. However, given the investment in current silicon processing lines, there is no reason to expect that these will be available soon, or that defect rates on typical process lines will improve more than an order of magnitude moving into the nanometre region. Thus, defects are guaranteed to remain a major technical issue at the architectural level.

A closely related problem is the longer term reliability of nanoelectronic technology. The reliability curve developed for ULSI logic by Shibayama et al (1997) (Figure 1) indicates that at gate densities in the order of 10^7 almost half of systems can be expected to have failed within 10 years (based on the assumption that a single gate failure results in the failure of the entire system). Extrapolating these curves for transistor densities in the order of 10^9 (the IST forecast for 2006) would imply a 90% failure rate within about 1.3 years[1]. To maintain the same reliability as a 1 million gate chip would require an error rate in the order of 10^{-16}/hour-gate, four orders of magnitude better than current technology. How these curves might eventually be extended to a system with 10^{23} devices is unclear. What is clear, however, is that nanocomputer architectures will certainly need to be dynamically defect tolerant - with an ability to find defects as they develop and to reconfigure around them (Koren and Koren 1998).

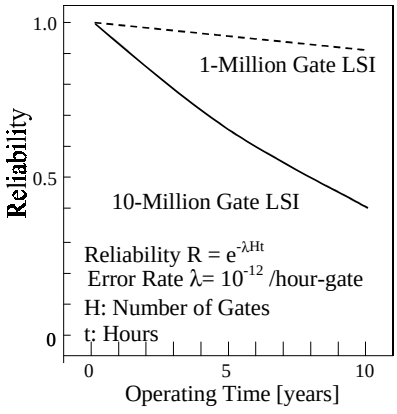

Figure 1 ULSI Reliability Curves - from (Shibayama et al 1997)

As a result, testing will represent a major issue in nanoelectronic systems. Currently, testing can account for up to 60% of the total costs of production for an ASIC - even for 250nm CMOS (SIA 1999), and this figure will

[1] Using the IST average of 5 transistors per gate.

become worse at higher densities. Run-time self-test regimes will therefore be increasingly important in the nanocomputer domain.

2.2 Wiring Delay

At a basic level, the wiring delay problem is simple to articulate: as interconnection width and thickness decrease, resistance per unit length increases, while as interconnections become denser (and oxide layers thinner), capacitance also tends to increase (Borkar 1999). For example, if the RC delay of a 1mm metal line in 0.5μm technology is 15ps then at 100nm (in the same materials) the delay would be 340ps (Sylvester and Keutzer 2001).

Ho, Mai and Horowitz (2001) have performed a detailed analysis of the performance of wires in scaled technologies and have identified two distinct characteristics. For short connections (those that tend to dominate current chip wiring) the ratio of local interconnection delay to gate delay stays very close to unity - i.e. interconnection delay closely tracks gate delay with scaling. For metal interconnections, this will be true down to approximately 10nm at which point the simple resistance relationship breaks down and the resistance increases due to quantum effects (Hadley and Mooij 2000).

On the other hand, global wiring tends to increase in length with increasing levels of integration, implying that the interconnection delay of these wires will increase relative to the basic gate delay. Sylvester and Keutzer (2001) conclude that the scaling of global wires will not be sustainable beyond about 180nm due to the rising RC delays of scaled-dimension conductors. However, as interconnect delay will be tolerably small in blocks of 50 – 100K gates, they argue for hierarchical implementation methodologies based on macro-blocks of this size.

In addition, at future gigahertz operating speeds, signal "time-of-flight" and attenuation will become significant limiting factors. As both of these depend on the dielectric constant of the propagation material, solving them will require significant changes to processing technology. For example, Ohmi et al (2001) have developed processes that use a gas-isolated, high-k gate dielectric, metal-gate, metal-substrate SOI scheme with thermally conducting through-holes to reduce temperature variations and increase interconnection reliability. These complex, aggressive fabrication schemes contrast markedly with the intrinsic self-assembly mechanisms proposed by Merkle (1996) and others.

2.3 Emerging Devices

The problems associated with the scaling of CMOS devices have led to a search for alternative transistor and circuit configurations. Proposals for silicon-based technologies include silicon-on-insulator (Taur et al 1997), single electron devices (Nakajima et al 1997), resonant-tunnelling devices (RTDs) (Capasso et al 1989), (Frazier et al 1993), double layer tunnelling transistors (Geppert 2000) and Schottky Barrier MOSFETS (Tucker 1997). Of these, RTDs appear to hold the most promise as a short to medium-term solution although most of the implementations in the literature to date are based on relatively complex heterostructure technologies - predominately based on GaAs.

RTDs are inherently fast and have been known and used for more than a decade. Their negative differential resistance (NDR) characteristics directly support multi-value logic styles (Waho, Chen and Yamamoto 1996) that can result in significantly simpler circuit designs. RTD circuits are typically based on one or more tunnelling diodes integrated with a conventional (often hetero-structure) FET (Mohan et al 1993). The main problem with this approach has been the need to match the current of the FET and the peak current of the diode(s) although a recent configuration avoids this problem by surrounding a vertical resonant diode structure with a Schottky control gate (Stock et al 2001).

Ultimately, electronic devices may simply cease to be an option at the scale of 1 or 2 nm. A number of molecular based technologies have been suggested as potential alternatives (Goldhaber-Gordon et al 1997, Reed et al 1999) as well as some computing architectures that might exploit them (Ellenbogen 1997, Ellenbogen and Love 1999). There have even been suggestions for nano-mechanical devices - somewhat reminiscent of Shannon's original (1949) relay logic (Drexler 1992, Merkle and Drexler 1996) as well as computational DNA systems (Young and Sheu 1997).

Finally, semiconductor behaviour has recently been demonstrated within very narrow carbon nanotube (fullerene) based structures (Wilson et al 2000). Nano-tube technology may eventually support the construction of non-volatile RAM and logic functions at integration levels approaching 10^{12} elements/cm^2, operating frequencies in excess of 100GHz (Rueckes et al 2000) and, as electron flow can be ballistic in short nanotube wires, supporting current densities in the order of 10^9A/cm^2 - well above the figure that would vaporize traditional interconnect metals. Simple logical operations with nanotubes have just been demonstrated (Liu et al 2001). Rueckes et al (2000) have built a bistable bit and designs for electromechanical logic and memory have been proposed (Ami and Joachim 2001). Further, high band-gap materials such as boron nitride (Chen et al 1999) may also offer interesting nanotube building blocks capable of working at significantly higher temperatures than carbon.

Although some of these emerging device technologies have been demonstrated in the laboratory, it is not at all clear which of them have the potential for robust, high-speed operation at room temperature - at least in the near future.

3 Nanocomputer Architecture Candidates

To date, architecture research has responded to the opportunities and challenges offered by device scaling in two ways. The first approach simply increases existing machine resources - more or larger caches; more on-chip processors, often including local DRAM (Kozyrakis and Patterson 1998), direct multi-threading support (i.e.

exploiting parallelism between concurrently running tasks rather than within a single algorithm) and other similar techniques. While being effective for some applications, these can quickly run into all of the physical limitations outlined previously, especially the wire-length problems that can result in unacceptable memory and I/O latency although the 50 to 100K-gate hierarchical design blocks suggested by Sylvester and Keutzer (2001) are certainly large enough to contain a small RISC processor or other quite sophisticated processing elements. Durbeck and Macias (2000) put it this way: "*... there is no clear way for CPU/memory architectures to tap into the extremely high switch counts … available with atomic-scale manufacture, because there is no clear way to massively scale up the (CPU) architecture. … there is no such thing as "more" Pentium®. There is such a thing as more Pentiums®, however.*"

The second approach uses modular and hierarchical architectures to improve the performance of traditional single-thread architectures (Vajapeyam and Valero 2001). Table 1, reproduced from Fountain et al (1998), compares the three main classes of parallel architectures in terms of characteristics applicable to the nanocomputer domain. They conclude that highly regular, locally connected, peripherally interfaced, data-parallel architectures offer a good match to the characteristics of nanoelectronic devices. However, it is worth noting that data-parallel architectures represent only a small portion of the interesting problems in computer architecture and are a poor match for most general purpose computing problems.

Future computer architectures may well be market application driven (Ronen et al 2001), with the characteristics of each market segment resulting in its own optimised parallel microarchitecture. Ronen et al, like Durbeck and Macias, clearly rule out the possibility of today's high-end microprocessor being tomorrow's low-power/low-cost solution.

Parameter	Data	Function	Neural
Degree of parallelism	High	Low	high
Processor Complexity	Low	High	medium
Interconnect Density	Low	High	high
Amount of Interfacing	Low	High	low
Extensibility	High	Low	low

Table 1 A Comparison of Three Parallel Architecture Classes (Fountain et al 1998)

3.1 Quantum Cellular Array Architectures

Cellular Arrays (CAs) have been known and studied for almost 40 years (von Neumann 1966). Their architecture is based on the replication of identical processing elements with nearest neighbour connection. The fundamental idea behind the operation of Quantum Cellular Automata (QCA) devices is that the energy state of a suitable assembly of electrons, initially in a specific ground state, will alter as a result of changed boundary conditions (Maccuci et al 1999).

Lent et al (1993) and more recently Porod (1998) have proposed specific realizations of this idea using two-electron cells composed of four quantum dots in which the polarization of one cell induces a polarization in a neighbouring cell through Coulomb interaction in a very non-linear fashion. If left alone, the two electrons will seek the configuration corresponding to the ground state of the cell by tunnelling ("hopping") between the dots. Lent et al have demonstrated that AND gates, OR gates, and inverters can be constructed and interconnected. Fountain et al (1998) comment that circuits built from QCA elements would form extremely coherent computing systems, although some concerns remain about their theoretical validity, and the optimum implementation of memory.

As the coulomb interactions in QCA are based on a small number of electrons (as low as one) they tend to be swamped by thermal noise unless they are operated at very low temperatures (in the milliKelvin range). This will very likely prevent them having a serious impact on the mainstream computing domain. An interesting variation on the QCA - based on magnetism - is described by (Cowburn and Welland 2000). In the Magnetic QCA (MQCA), networks of interacting submicron magnetic dots are used to perform logic operations and propagate information. As MQCA energies are in the order of 1eV they will work well at room temperature. Cowburn and Welland suggest that MQCA technology may eventually offer active device densities in the order of $2.5 \times 10^{11}/cm^2$ with a power-delay product that is 10^4 times less than current CMOS.

3.2 Synthetic Neural Systems

Synthetic Neural Network (SNN) systems, also called artificial neural networks, connectionist networks, or parallel distributed processing networks, are concerned with the synthesis, design, fabrication, training and analysis of neuromorphic (i.e. brain-inspired) electronic systems (Ferry, Grondin and Akers 1989). These systems achieve high performance via the adaptive interconnection of simple switching elements that process information in parallel. Arrays of simple neural processing elements show features such as association, fault tolerance and self-organisation. However, while the complexity of neural processing is low, the interconnection density is high (see Table 1) so there is still a question as to their applicability in the nanocomputer domain.

So far, most of the work in neural networks relates to static networks - classifier systems or associative networks (Glösekötter, Pacha and Goser 1998) that learn to map data by modifying their internal configuration. For example, in addition to employing QCA cells to encode binary information, Porod (1998) has proposed an analogue Quantum-Dot Cellular Neural Network (Q-CNN) in which each cell is described by appropriate state variables, and the dynamics of the whole array is given by the dynamics governing each cell plus the influence exerted by its neighbours.

The alternative approaches - time dependent, biologically inspired networks that process data using a dynamical systems approach - exhibit more interesting emergent behaviour. They require vast numbers of devices to implement but these are likely to be available in the

nanocomputing domain. However, as in all CNN systems, each neural node has to be connected to at least 10 to 100 synapses for useful computation, so it is questionable whether the low drive capability of nanoelectronic devices will be suitable building blocks for these systems.

3.3 Locally Connected Machines

A common example of regular, locally connected, data-parallel architectures is the Single Instruction Multiple Data machine. SIMD machines exploit the inherent data parallelism in many algorithms - especially those targeting signal and image processing (Gayles et al 2000). Fountain et al (1998) identify the characteristics that may make the SIMD topology suited to nanocomputer architecture as:

- a regular and repetitive structure;
- *local* connections between all system elements;
- all external connections made at the array edge;
- the existence of feasible strategies for fault tolerance.

However, SIMD architecture still suffer from two major problems - global instruction issue as well as global control and clock signals. Global clocking is required by SIMD machines not only to drive each individual (synchronous) element but also to manage inter-element synchronisation.

It is clear from the analysis of Fountain et al (1998) that the interconnection costs of SIMD in the nano-domain are very high - with the majority of the die area in their experiments being taken up by control signal distribution. Numerous asynchronous design techniques (e.g. Hauck 1995) have been proposed to overcome the need for a global clock in SIMD machines. While it is still unclear whether, in practice, these asynchronous techniques actually offer improved performance, they are at least as good as the conventional synchronous approach and may offer the only means to overcome global communication constraints in the nanocomputer domain.

The same considerations appear to constrain other multi-processor architectures such as MIMD. Crawley (1997) has performed a series of experiments on various MIMD architectures and concluded that inter-processor communications will be limited by the availability of wider metal tracks on upper layers (called "fat" wiring by Crawley). The tradeoff here is between track resistance (and therefore delay) and interconnection density. Crawley also notes that more complex computational structures such as carry look-ahead begin to lose their advantages over simpler and smaller structures once wiring delays are factored in.

3.3.1 Propagated Instruction Processor

The Propagated Instruction Processor was proposed by Fountain (1997) as a way of avoiding the interconnection problem in SIMD arising from its global instruction flow characteristics. In the PIP architecture, instructions are pipelined in a horizontal direction such that the single-bit functional units can operate simultaneously on multiple algorithms. The technique shares many of the characteristics of SIMD, pipelined processors and systolic arrays. One of the primary advantages of the architecture is its completely local interconnection scheme that results in high performance on selected applications.

However, the architecture is still basically SIMD and thus will work best with algorithms from which significant data parallelism can be extracted - e.g. Fountain's examples of point-wise 1-bit AND of two images, an 8-bit local median filter, 32-bit point-wise floating point division and an 8-bit global matrix multiplication (Fountain 1997). In addition, the fault tolerance of the PIP may ultimately depend of an ability to bypass faulty processors without upsetting the timing relationship between propagating instructions - something that has not been reported to date.

3.3.2 Merged Processor/Memory Systems - IRAM and RAW

The structure and performance of memory chips are becoming a liability to computer architecture. There are two basic problems: firstly the so-called "memory wall" (or gap) resulting from a divergence in the relative speed of processor and DRAM that is growing at 50% per year (Flynn 1999). Secondly, while DRAM size is increasing by 60% per year, its fundamental organisation – a single DRAM chip with a single access port - is becoming increasingly difficult to use effectively. This observation has led to the development of a number of merged memory/processor architectures. Two notable examples of this approach are the Intelligent RAM (IRAM) system (Patterson et al 1997), and the Reconfigurable Architecture Workstation (RAW) (Waingold et al 1997).

The IRAM system merges processing and memory onto a single chip. The objective is to lower memory latency, increase memory bandwidth, and at the same time improve energy efficiency. The IRAM scheme revives the vector architecture originally found in supercomputers and implements it by merging at least 16MB of DRAM, a 64-bit two-way superscalar processor core with caches, variable width vector units, and a high-performance memory switch onto a single chip.

The RAW microprocessor chip comprises a set of replicated tiles, each tile containing a simple RISC like processor, a small amount of configurable logic, and a portion of memory for instructions and data. Each tile has an associated programmable switch which connects the tiles in a wide-channel point-to-point interconnect. The compiler statically schedules multiple streams of computations, with one program counter per tile. The interconnect provides register-to-register communication with very low latency and can also be statically scheduled. The compiler is thus able to schedule instruction-level parallelism across the tiles and exploit the large number of registers and memory ports.

3.4 Reconfigurable and Defect Tolerant Hardware

Reconfigurable hardware can be used in a number of ways: to provide reconfigurable functional units within a host processor: as a reconfigurable coprocessor unit; as an attached reconfigurable processor in a multiprocessor

system; or as a loosely coupled external standalone processing unit (Compton and Hauck 2000). One of the primary variations between these architectures is the degree of coupling (if any) with a host microprocessor. For example, the OneChip architecture (Carrillo and Chow 2001) integrates a Reconfigurable Functional Unit (RFU) into the pipeline of a superscalar Reduced Instruction Set Computer (RISC). The reconfigurable logic appears as a set of Programmable Function Units that operate in parallel with the standard processor. The Berkeley hybrid MIPS architecture, Garp, (Hauser and Wawrzynek 1997) includes a reconfigurable coprocessor that shares a single memory hierarchy with the standard processor, while the Chimaera system (Hauck et al 1997) integrates reconfigurable logic into the host processor itself with direct access to the host's register file.

3.4.1 Reconfigurable Logic and FPGAs

When FPGAs were first introduced they were primarily considered to be just another form of (mask programmed) gate array - albeit without the large start-up costs and lead times. Since then FPGAs have moved beyond the simple implementation of digital (glue) logic and into general-purpose computation. Although offering flexibility and the ability to optimise an architecture for a particular application, programmable logic tends to be inefficient at implementing certain types of operations, such as loop and branch control (Hartenstein 2001). In addition, there is a perception that fine-grained architectures (those with path widths of one or two bits) exhibit high routing overheads and poor routability (Hartenstein 1997). It is probably true that field-programmable gate arrays (FPGAs) will always be slower and less dense than the equivalent function in full custom, standard cell or mask programmed gate arrays as the configuration nodes take up significant space as well as adding extra capacitance and resistance (and thus delay) to the signal lines. The challenge will be to find new organisations that optimise the balance between reconfigurability and performance.

FPGAs exhibit a range of block granularity. Very fine-grained logic blocks have been applied to bit level manipulation of data in applications such as encryption image processing and filters (Ohta et al 1998) while coarse-grained architectures are primarily used for word-width datapath circuits. Again, the tradeoff is between flexibility and performance - as coarse-grained blocks are optimized for large computations, they can be faster and smaller overall than a set of smaller cells connected to form the same type of structure. However, they also tend to be less flexible, forcing the application to be adapted to the architecture just as for conventional processors.

New techniques are required that maintain the flexibility of the FPGA structure while minimising the effects of its configuration overheads. In particular, the serial re-configuration mechanisms of current FPGAs will clearly not scale indefinitely - a device with 10^{23} programmable elements could take a few million years to configure! One of the few current proposals to directly address this issue is the Cell Matrix architecture (Macias 1999).

3.4.2 Defect Tolerant Hardware

As identified previously, defect tolerant architectures will be the only way to economically build computing systems with hundreds of billions of devices because any system using nanoscale components will contain significant numbers of defects. One example of an existing defect tolerant custom configurable system is the Teramac (Heath et al 1998). The basic idea was to build a system out of cheap but imperfect components (FPGAs in this case), find the defects and configure the available good resources using software. The high routability of the Teramac is based on the availability of excessive interconnections - due to its "fat-tree" routing configuration. However, it is possible that current methods for detecting defects such as those used in Teramac will not scale to devices with 10^{10} configuration bits (Goldstein 2001). Thus, novel parallel defect mapping techniques will need to be developed - most probably built-in, and coupled with self-configuration mechanisms of the type suggested by Macias (1999) or Gericota et al (2001).

"Embryonics" (Mange et al 2000) is a biologically inspired scheme that aims to produce highly robust integrated circuits with self-repair and self-replication properties. In this case, the reconfiguration algorithm is performed on-chip in the form of an "artificial genome" containing the basic configuration of the cell. Its fault tolerance relies on fault detection and location via built-in self-test plus an ability to bypass faulty cells and to substitute spare cells in their place. However, the simplistic system employed – substituting entire columns of cells if just one cell is faulty - has too many limitations to scale successfully, not the least of which is the need to estimate the number of standby logic cells that might be required in a typical implementation.

While the various demonstration systems have their limitations, they do illustrate that it is possible to build a computer system that contains defective components as long as there is sufficient communication bandwidth to support the discovery and use of working components plus the capacity to perform such a rearrangement of working components. An ability to perform self-test will be critical. It is possible that the most important components in nanocomputer architecture might turn out to be its configuration switches and controls.

4 Nanocomputer Architecture

Having surveyed the current challenges and opportunities in the nanoelectronic domain, it is now possible to make some predictions about likely characteristics of future nanocomputer architectures. As has been seen, these characteristics lead to: the need for extremely localised interconnect; the use of homogenous arrays that are able to support heterogenous processing structures; the ability to exploit parallelism at multiple levels (e.g. instruction level, multi-threaded etc.); a requirement for dynamic reconfigurability with low reconfiguration overheads (in both space and time) as well as defect and/or fault tolerance - at both the commissioning/configuration stage and at run-time.

4.1.1 Reconfiguring the Memory Gap

Although the fabrication of RAM and digital logic are completely separated at present, and there is a vast and expensive infrastructure supporting both, the functions of logic and memory must eventually merge if the increasing gap in performance between the two is to be overcome.

Non-volatility will be the key. When DRAM is finally superseded by non-volatile memory, it will be possible to envisage a computing system in which all storage – disk, main memory and caches – merges into the processing mesh. Figure 2 illustrates one reason why this would be a good idea. In the memory hierarchy of a conventional processor, it is possible for code and data items to be duplicated in more than five places in the system (e.g. disk, disk cache, memory, memory cache(s), registers). It is fairly easy to argue that this is not a good use of the available machine resources (Flynn 1999).

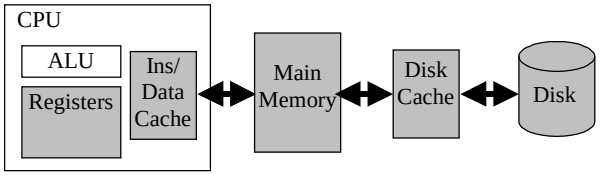

Figure 2 Conventional Memory Hierarchy

At present, the two main contenders for non-volatile technology are floating-gate structures and magnetics. The roadmap for non-volatile Magnetic RAM (MRAM) shows it reaching a density of 1Gbit by 2006 (Inomata 2001) and "nano-magnetic" technology (Cowburn and Welland 2000) may eventually support densities of 10^{12} bits.

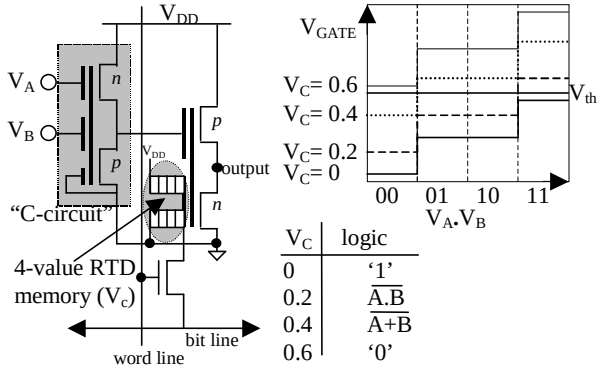

Figure 3 Reconfigurable Threshold Logic Cell

Although floating-gate devices have been under development for around 30 years (Ohnakado and Ajika 2001), they are unlikely to reach the same densities as magnetic based systems. Figure 3 illustrates an example of a non-volatile reconfigurable logic cell that merges the variable threshold νMOS logic of Kotani, Shibata and Ohmi (1992) with the multi-valued memory cell of Wei and Lin (1992). In this cell, the initial "c-circuit" acts as a simple D/A converter, producing a voltage that is proportional to the two input values. The RTD-based 4-valued memory is used to adjust the offset of this voltage, shown as V_{GATE} in Figure 3, thereby reconfiguring the function of the (two-input) logic gate.

The low-overhead reconfigurability offered by this type of circuit – or by alternatives such as nano-magnetics – may eventually support the creation of a merged memory/processing structure, in which the idea of "mass storage" is replaced by "mass reconfiguration" as program and data become indistinguishable from the processing mesh.

4.1.2 Coding Around the Defects

As outlined previously, all nanocomputer systems will contain faulty components. Defect/fault tolerance supporting the ability to detect and avoid defects at both the commissioning/ configuration stage and at run-time, will therefore be of critical importance. Forshaw et al (2001) have shown that it is theoretically possible to produce working systems with defect rates as high as 10^{-5} to 10^{-4} if reconfigurable techniques are used to bypass the defects.

Existing static fault mapping techniques (such as are used in hard disk systems, for example) may represent a good starting point, but it is likely that built-in self test (BIST) will be necessary to maintain system integrity in the presence of soft-errors and noise. There have been some initial studies into how to optimally configure BIST in an extremely large cellular array (Goldstein 2000) but no general solutions have been developed as yet.

4.1.3 "Grain Wars"

At least in the short term, the outcome of the coarse-grain vs. fine-grain argument is difficult to predict as there are strong arguments for both styles. Eventually, however, all nanocomputer architectures will be formed from arrays of simple cells with highly localised interconnect. This will be an inevitable outcome of shrinking geometries as devices evolve towards molecular and quantum/ single electron technologies.

At present, the tendency towards course-grained architectures (e.g. multiple CPU blocks, ALU arrays etc.) is being driven by the high overheads imposed by reconfiguration techniques in devices such as FPGAs. If this can be reduced, for example by the use of multi-value techniques such as was illustrated previously, then fine-grained structures offer a much more general solution to the creation of flexible computing platforms.

4.1.4 A Processor for Every Process

It appears, then, that the ultimate computing fabric will be an homogenous, fine-grained, non-volatile, fault tolerant, reconfigurable, processing array, exhibiting adjacent or nearest neighbour interconnect only and supporting heterogenous structures that are derived by compiling a HLL program. The processing fabric will be reconfigurable in a way that maximises the system's ability to exploit parallelism - consisting of as many individual processing meshes as are necessary, each configured in an optimal manner for the particular function.

This scheme takes advantage of the future abundance of processing with a scarcity of interconnect. Instead of a

large number of constructed programs, we may instead try to store (close to) all possible programs in the device. In this organisation, programs would be continuously configured within the non-volatile memory/logic - available to respond to an input stimulus by generating an output whenever required. The concept of memory hierarchy would be completely eliminated – if the logic structure is large enough to store and execute all "programs" for that machine. In the more immediate term, configuration "context switching" (Kearney 1998) will replace the loader of conventional operating systems.

As this architecture effectively merges processor logic, RAM and disk into one structure, the only remaining potential performance bottleneck will be the input/output channel. Although I/O bandwidth tends not to be as great a problem as the memory/processor interface, current processors working in domains such as multimedia already have some difficulty maintaining high data throughputs and this will continue to be an issue (for example with 3D multimedia). The challenge will be to develop flexible parallel I/O configurations that will allow the internal processes to operate at peak performance.

4.1.5 Legacy Software

General purpose computing is largely sequential, dominated by control dependencies and tends to rely on dynamic data structures that currently do not map well to array architectures (Mangione-Smith 1997). However a nanocomputer will inherit a vast quantity of legacy software that cannot be ignored (it could be said that the Y2K issue revealed just how extraordinarily long-lived are some types of software).

There is no doubt that the translation from a source program to systems with billions of gates will be an extremely complex task. But, ironically, the very availability of a large number of gates makes the task easier. In this case, the synthesis process has access to the resources necessary to create all possible computation paths in the "program" and then simply select the single correct result at the end. This aggressive form of speculation is the basis of the synthesis process for the PipeRench architecture (Goldstein et al 2000).

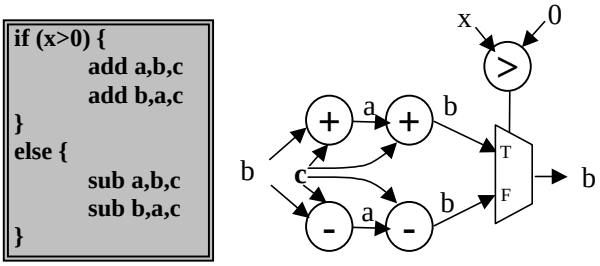

Figure 4 Processing Graph Fragment

The graph fragment in Figure 4 illustrates this point. All arithmetic functions are duplicated as required, as are the intermediate variables - without concern for the hazards that would occur in a typical pipelined system. Note that in this simplified diagram, no data synchronisation mechanism is shown. A number of ideas have been proposed that would be applicable to the nanocomputer domain, from Asynchronous Wave Pipelines (Hauck, Katoch and Huss 2000) through to gate-level nano-pipelined computation using RTDs (Mazumder et al 1998).

5 Conclusions

We argue that future nanocomputer architectures will be formed from non-volatile reconfigurable, locally-connected hardware meshes that merge processing and memory. In this paper, we have highlighted the characteristics of nanoelectronic devices that make this most likely - primarily the severe limitations on the length interconnection lines between devices. It appears that the current trend towards coarse-grained structures may not be supportable in the long term. If the overheads associated with reconfigurability can be reduced or even eliminated, architectures based on fine-grained meshes with rich, local interconnect offer a better match to the characteristics of nanoelectronic devices.

Of course, having access to a vast, reconfigurable computing platform is only the first step. The question still remains as to what use such an architecture might be put. Will it be necessary to own an "Avagadro computer" in order to run Windows® 2030? Moravec (1998) has suggested that, if the power of the human brain is in the synapses connecting neurons, then it would take the equivalent of 10^{14} instructions/sec. to mimic a brain with an estimated 10^{13} - 10^{15} synapses. Might the power of nanocomputer architecture finally release the ghost in the machine?

6 References

Ami, S., Joachim, C. (2001). Logic Gates and Memory Cells Based on Single C60 Electromechanical Transistors. *Nanotechnology* **12** (1):44-52.

Asai, S., Wada, Y. (1997). Technology Challenges for Integration Near and Below 0.1um. *Proceedings of the IEEE* **85** (4):505-520.

Borkar, S. (1999). Design Challenges of Technology Scaling. *IEEE Micro* **19** (4):23-29.

Borkar, S. (2000). Obeying Moore's Law Beyond 0.18 Micron. *Proc. ASIC/SOC Conference, 2000. Proceedings. 13th Annual IEEE International*, IEEE, pp:26-31.

Capasso, F., Sen, S., Beltram, F., Lunardi, L., Vangurlekar, A. S., Smith, P., Shah, N. J., Malik, R. J., Cho, A. Y. (1989). Quantum Functional Devices: Resonant-Tunneling Transistors, Circuits with Reduced Complexity, and Multi-Valued Logic. *IEEE Transactions on Electron Devices* **36.** (10).

Carrillo, J. E., Chow, P. (2001). The Effect of Reconfigurable Units in Superscalar Processors. *Proc. Ninth International Symposium on Field Programmable Gate Arrays*, FPGA 2001, Monterey, CA, USA, ACM, February 11-13, 2001, pp:141-150.

Chen, Y., Chadderton, L.T., Fitz Gerald, J., Williams, J.S. (1999). A Solid State Process for Formation of Boron Nitride Nanotubes. *Applied Physics Letters*, **74** (20):2960-2962.

Compton, C., Hauck, S. (2000). An Introduction to Reconfigurable Computing. *IEEE Computer* (April 2000).

Cowburn, R. P., Welland, M. E. (2000). Room Temperature Magnetic Quantum Cellular Automata. *Science* **287**:1466-1468.

Crawley, D. (1997). *An Analysis of MIMD Processor Node Designs for Nanoelectronic Systems*. Internal Report. Image Processing Group, Department of Physics & Astronomy, University College. London.

Davari, B. (1999). CMOS Technology: Present and Future. *Proc. IEEE Symposium on VLSI Circuits, Digest of Technical Papers*, IEEE, 1999, pp:5-9.

Drexler, K. E. (1992). Nanosystems: Molecular Machinery, Manufacturing and Computation, Wiley & Sons. Inc.

Durbeck, L. J. K. (2001). An Approach to Designing Extremely Large, Extremely Parallel Systems. Abstract of a talk given at The Conference on High Speed Computing, Salishan Lodge, Gleneden, Oregon, U.S.A., April 26 2001. Conference sponsored by Los Alamos, Lawrence Livermore, and Sandia National Laboratories. Cell Matrix Corporation, http://www.cellmatrix.com/entryway/products/pub/SalishanAbstract2.html, accessed: 12 August, 2001.

Durbeck, L. J. K., Macias, N. J. (2000). *The Cell Matrix: An Architecture for Nanocomputing*. Cell Matrix Corporation, http://www.cellmatrix.com/entryway/products/pub/publications.html, accessed: 12 August, 2001.

Ellenbogen, J. C. (1997). *Matter as Software*. The Mitre Corporation. http://www.mitre.org/technology/nanotech.

Ellenbogen, J. C., and Love, J. C. (1999). Architectures for Molecular Electronic Computers: 1. Logic Structures and an Adder Built from Molecular Electronic Diodes,. The Mitre Corporation. http://www.mitre.org/technology/nanotech

Ferry, D. K., Grondin, R. O., Akers, L. A. (1989). Two-Dimensional Automata in VLSI. In *Sub-Micron Integrated Circuits*. R. K. Watts (ed), John Wiley & Sons.

Flynn, M. J. (1999). Basic Issues in Microprocessor Architecture. *Journal of Systems Architecture* **45** (12-13):939-948.

Forshaw, M. R. B., Nikolic, K., Sadek, A. (2001). *3rd Annual Report, Autonomous Nanoelectronic Systems With Extended Replication and Signalling, ANSWERS*. Technical Report. University College London, Image Processing Group. London, U.K. http://ipga.phys.ucl.ac.uk/research/answers/reports/3rd_year_UCL.pdf

Fountain, T. J. (1997). The Propagated Instruction Processor. *Proc. Workshop on Innovative Circuits and Systems for Nanoelectronics*, Delft, pp:69-74.

Fountain, T. J., Duff, M. J. B. D., Crawley, D. G., Tomlinson, C. and Moffat, C. (1998). The Use of Nanoelectronic Devices in Highly-Parallel Computing Systems. *IEEE Transactions on VLSI Systems* **6** (1):31-38.

Frazier, G., Taddiken, A., Seabaugh, A., Randall, J. (1993). Nanoelectronic Circuits using Resonant Tunneling Transistors and Diodes. *Proc. Solid-State Circuits Conference, 1993. Digest of Technical Papers. 40th ISSCC., 1993*, IEEE International, 24-26 Feb. 1993, pp:174 - 175.

Gayles, E. S., Kelliher, T. P., Owens, R. M., Irwin, M. J. (2000). The Design of the MGAP-2: A Micro-Grained Massively Parallel Array. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **8** (6):709 - 716.

Gelsinger, P. P. (2001). Microprocessors for the New Millennium: Challenges, Opportunities, and New Frontiers. *Proc. International Solid-State Circuits Conference ISSCC2001*, San Francisco, USA, IEEE.

Geppert, L. (2000) Quantum Transistors: Toward Nanoelectronics. IEEE Spectrum: 46 - 51, September 2000

Gericota, M. G., Alves, G.R., Silva, M.L., Ferreira, J.M. (2001). DRAFT: An On-line Fault Detection Method for Dynamic & Partially Reconfigurable FPGAs. *Proc. Seventh International On-Line Testing Workshop*, IEEE, pp:34 -36.

Ghosh, P., Mangaser, R., Mark, C., Rose, K. (1999). Interconnect-Dominated VLSI Design. *Proc. Proceedings of 20th Anniversary Conference on Advanced Research in VLSI*, 21-24 March 1999, pp:114 - 122.

Glösekötter, P., Pacha, C., Goser, K. (1998). Associative Matrix for Nano-Scale Integrated Circuits. Proc. Proceedings of the Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems, IEEE.

Goldhaber-Gordon, D., Montemerlo, M. S., Love, J. C., Opiteck, G. J., Ellenbogen, J. C. (1997). Overview of Nanoelectronic Devices. *Proceedings of the IEEE* **85** (4):521-540.

Goldstein, S. C. (2001). Electronic Nanotechnology and Reconfigurable Computing. *Proc. IEEE Computer Society Workshop on VLSI*, Orlando, Florida, IEEE.

Goldstein, S. C., Schmit, H., Budiu, M., Cadambi, S., Moe, M., Taylor, R. R. (2000). PipeRench: A Reconfigurable Architecture and Compiler. *IEEE Computer* (April 2000):70-77.

Hadley, P., Mooij, J. E. (2000). *Quantum Nanocircuits: Chips of the Future?* Internal Report. Delft Institute of Microelectronics and Submicron Technology DIMES and Department of Applied Physics. Delft, NL. http://vortex.tn.tudelft.nl/publi/2000/quantumdev/qdevices.html

Hanyu, T., Teranishi, K., Kameyama, M. (1998). Multiple-Valued Logic-in-Memory VLSI Based on a Floating-Gate-MOS Pass-Transistor Network. *Proc. IEEE International Solid-State Circuits Conference*, pp:194 -195, 437.

Hartenstein, R. (1997). The Microprocessor is No Longer General Purpose: Why Future Reconfigurable Platforms Will Win. *Proc. Second Annual IEEE International Conference on Innovative Systems in Silicon*, pp:2 -12.

Hartenstein, R. W. (2001). Coarse Grain Reconfigurable Architectures. *Proc. Proceedings of the ASP-DAC 2001 Design Automation Conference*, 30 Jan.-2 Feb. 2001, pp:564-569.

Hauck, O., Katoch, A., Huss, S.A. (2000). VLSI System Design Using Asynchronous Wave Pipelines: A 0.35µ CMOS 1.5GHz Elliptic Curve Public Key Cryptosystem Chip. *Proc. Sixth International Symposium on Advanced Research in Asynchronous Circuits and Systems, ASYNC 2000*, 2-6 April 2000, pp:188 -197.

Hauck, S. (1995). Asynchronous Design Methodologies: An Overview. *Proceedings of the IEEE* **83** (1):69 - 93.

Hauck, S., Fry, T.W., Hosler, M.M., Kao, J.P. (1997). The Chimaera Reconfigurable Functional Unit. *Proc. IEEE Symposium on Field-Programmable Custom Computing Machines, FCCM'97*, pp:87 - 96.

Hauser, J. R., Wawrzynek, J. (1997) Garp: A MIPS Processor with a Reconfigurable Coprocessor, *Proc. IEEE Symposium on FPGAs for Custom Computing Machines, FCCM'97*, pp: 12-21.

Heath, J. R., Kuekes, P. J., Snider, G. S., Williams, S. (1998). A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology. *Science* **280** (12 June 1998):1716-21.

Ho, R., Mai, K.W., Horowitz, M.A. (2001). The Future of Wires. *Proceedings of the IEEE* **89** (4):490 -504.

Inomata, K. (2001). Present and Future of Magnetic RAM Technology. *IEICE Transactions on Electronics* **E84-C** (6):740 - 746.

IST (2000). *Technology Roadmap for Nanoelectronics*, European Commission IST Programme - Future and Emerging Technologies. Compano, R., Molenkamp, L., Paul, D. J. (eds).

Kearney, D., Keifer, R. (1998). *Hardware Context Switching in a Signal Processing Application for an FPGA Custom Computer*. Advanced Computing Research Centre, School of Computer and Information Science, University of SA.

Koren, I., Koren, Z. (1998). Defect Tolernce in VLSI Circuits: Techniques and Yield Analysis. *Proceedings of the IEEE* **86** (6):1819 - 1836.

Kotani, K., Shibata, T., Ohmi, T. (1992). Neuron-MOS Binary Logic Circuits Featuring Dramatic Reduction in Transistor Count and Interconnections. Proc. International Electron Devices Meeting, 1992, 13-16 Dec. 1992, pp:431 -434.

Kozyrakis, C. E., Patterson, D. A. (1998). A New Direction for Computer Architecture Research. *IEEE Computer* **31**(11):24-32.

Lent, C. S., Tougaw, P. D., Porod, W., Bernstein, G. H. (1993). Quantum Cellular Automata. *Nanotechnology* **4** (1):49-57.

Liu, X., Lee, C., Zhou, C, Han, J. (2001). Carbon Nanotube Field-Effect Inverters. *Applied Physics Letters* **79** (20):3329-3331.

Maccuci, M., Francaviglia, S., Luchetti, G., Iannaccone, G. (1999). *Quantum-Dot Cellular Automata Circuit Simulation*. Technical Report. University of Pisa, Department of Information Engineering.

Macias, N. J. (1999). The PIG Paradigm: The Design and Use of a Massively Parallel Fine Grained Self-Reconfigurable Infinitely Scalable Architecture. *Proc. First NASA/DoD Workshop on Evolvable Hardware*, 1999.

Mange, D., Sipper, M., Stauffer, A., Tempesti, G. (2000). Toward Robust Integrated Circuits: The Embryonics Approach. *Proceedings of the IEEE* **88** (4):516-543.

Mangione-Smith, W. H., Hutchings, B. L. (1997). Configurable Computing: The Road Ahead. In *Reconfigurable Architectures: High Performance by Configware*. R. Hartenstein, V. Prasanna (ed). Chicago, IT Press**:**81 - 96.

Margolus, N. (1998). Crystalline Computation. In *The Feynman Lecture Series on Computation, Volume 2*. A. Hey (ed), Addison-Wesley.

Mazumder, P., Kulkarni, S., Bhattacharya, M., Jian Ping Sun, Haddad, G.I. (1998). Digital Circuit Applications of Resonant Tunneling Devices. *Proceedings of the IEEE* **86** (4):664 -686.

McFarland, G. W. (1997): CMOS Technology Scaling and its Impact on Cache Delay. PhD Thesis. Stanford University,

Merkle, R. C. (1996). Design Considerations for an Assembler. *Nanotechnology* **7** (3):210-215.

Merkle, R. C., Drexler, K. E. (1996). Helical Logic. *Nanotechnology* **7** (4):325-339.

Mohan, S., Mazumder, P., Haddad, G. I., Mains, R. K., Sun, J. P. (1993). Logic Design Based on Negative Differential Resistance Characteristics of Quantum Electronic Devices. *IEE Proceedings-G: Electronic Devices* **140** (6):383-391.

Montemerlo, M., Love, C., Opiteck, G., Goldhaber-Gordon, D., Ellenbogen, J. (1996). *Technologies and Designs for Electronic Nanocomputers*. The Mitre Corporation. http://www.mitre.org/technology/nanotech/

Moravec, H. (1998). *When Will Computer Hardware Match the Human Brain?* Journal of Transhumanism, Vol. 1, http://www.transhumanist.com/volume1/moravec.htm, accessed: 4 March 2001.

Nakajima, A., Futatsugi, T., Kosemura, K., Fukano, T., Yokoyama, N. (1997). Room Temperature Operation of Si Single-electron Memory with Self-aligned Floating Dot Gate. *Applied Physics Letters*, **70** (13):1742 - 1744.

Ohmi, T., Sugawa, S., Kotani, K., Hirayama, M., Morimoto, A. (2001). New Paradigm of Silicon Technology. *Proceedings of the IEEE* **89** (3):394 - 412.

Ohnakado, T., Ajika, N. (2001). Review of Device Technologies of Flash Memories. *IEICE Transactions on Electronics* E84-C (6):724-733.

Ohta, A., Isshiki, T., Kunieda, H. (1998). A New High Logic Density FPGA For Bit-Serial Pipeline Datapath. *Proc. IEEE Asia-Pacific Conference on Circuits and Systems, APCCAS 1998*, 24-27 Nov. 1998, pp:755 - 758.

Parihar, V., Singh, R., Poole, K. F. (1998). Silicon Nanoelectronics: 100nm Barriers and Potential Solutions. *Proc. IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, IEEE, 1998, pp:427-121.

Patterson, D., Anderson, T., Cardwell, N., Fromm, R., Keeton, K., Kozyrakis, C., Thomas, R., Yelick, K. (1997). A Case for Intelligent RAM. *IEEE Micro* **17** (2):34 -44.

Porod, W. (1998). Towards Nanoelectronics: Possible CNN Implementations Using Nanoelectronic Devices. *Proc. 5th IEEE International Workshop on Cellular Neural Networks and their Applications*, London, England, 14-17 April 1998, pp:20 - 25.

Reed, M. A., Bennett, D.W., Chen, J., Grubisha, D.S., Jones, L., Rawlett, A.M., Tour, J.M., Zhou, C. (1999). Prospects for Molecular-Scale Devices. *Proc. IEEE Electron Devices Meeting*, IEEE.

Ronen, R., Mendelson, A., Lai, K., Shih-Lien Lu, Pollack, F., Shen, J.P. (2001). Coming Challenges in Microarchitecture and Architecture. *Proceedings of the IEEE* **98** (3):325 - 340.

Rueckes, T., Kim, K., Joselevich, E., Tseng, G. Y., Cheung, C-L., Lieber, C. M. (2000). Carbon Nanotube-Based Nonvolatile Random Access Memory for Molecular Computing. *Science* **289**:94 - 97.

Shibayama, A., Igura, H., Mizuno, M., Yamashina, M. (1997). An Autonomous Reconfigurable Cell Array for Fault-Tolerant LSIs. *Proc. IEEE International Solid State Circuits Conference, ISSCC97*, IEEE, February 7, 1997, pp:230 - 231, 462.

SIA (1999). International Technology Roadmap for Semiconductors, Semiconductor Industry Association.

Stock, J., Malindretos, J., Indlekofer, K.M., Pottgens, M., Forster, A., Luth, H. (2001). A Vertical Resonant Tunneling Transistor for Application in Digital Logic Circuits. *IEEE Transactions on Electron Devices* **48** (6):1028 -1032.

Sylvester, D., Keutzer, K (2001). Impact of Small Process Geometries on Microarchitectures in Systems on a Chip. *Proceedings of the IEEE* **89** (4):467 - 489.

Tang, S. H., Chang, L., Lindert, N., Choi, Y-K., Lee, W-C., Huang, X., Subramanian, V., Bokor, J., King, T-J., Hu, C. (2001). FinFET — A Quasi-Planar Double-Gate MOSFET. *Proc. IEEE International Solid State Circuits Conference, ISSCC 2001*, San Francisco, USA, February 2001.

Taur, Y., Buchanan, D.A., Wei Chen, Frank, D.J., Ismail, K.E., Shih-Hsien Lo, Sai-Halasz, G.A., Viswanathan, R.G., Wann, H.-J. C., Wind, S.J., Hon-Sum Wong (1997). CMOS Scaling into the Nanometer Regime. *Proceedings of the IEEE* **85** (4):486 - 504.

Timp, G. L., Howard, R. E. and Mankiewich, P., M. (1999). Nano-electronics for Advanced Computation and Communication. In *Nanotechnology*. G. L. Timp (ed). New-York:, Springer-Verlag Inc.

Tucker, J. R. (1997). Schottky Barrier MOSFETS for Silicon Nanoelectronics. *Proc. Proceedings of Advanced Workshop on Frontiers in Electronics, WOFE '97*, 6-11 Jan. 1997, pp:97-100.

Vajapeyam, S., Valero, M. (2001). Early 21st Century Processors. *IEEE Computer* **34** (4):47-50.

von Neumann, J. (1966). *Theory of Self-Reproducing Automata*, University of Illinois Press.

Waho, T., Chen, K.J., Yamamoto, M. (1996). A Novel Multiple-Valued Logic Gate Using Resonant Tunneling Devices. IEEE Electron Device Letters 17 (5):223-225.

Waingold, E., Tayor, M., Srikrishna, D., Sarkar, V., Lee, W., Lee, V., Kim, J., Frank, M., Finch, P., Barua, R., Babb, J., Amarasinghe,S. Agarwal, A. (1997). Baring It All to Software: Raw Machines. *IEEE Computer* **30**:83 - 96.

Wei, S.-J., Lin, H.C. (1992). Multivalued SRAM Cell Using Resonant Tunneling Diodes. *IEEE Journal of Solid-State Circuits* **27** (2):212-216.

Wilson, M., Patney, H., Lee, G., Evans, L. (2000). New Wave Electronics and the Perfect Tube. *Monitor, March-May 2001*:14-17.

Young, W. C., Sheu, B. J. (1997) Unraveling the Future of Computing. IEEE Circuits and Devices Magazine **13**: 14 - 21, November 1997.