

Contents

1	Preface	2
2	Performance comparison of CMOS, RTD, SET and QCA devices against a standard computing task	3
2.1	Benchmark Architecture	3
2.2	Results	4
2.3	Discussion	6
3	Fault-tolerant techniques for nanocomputers	7
3.1	<i>R</i> -Modular Redundancy	7
3.2	Cascaded Triple-Modular Redundancy	8
3.3	NAND Multiplexing	9
3.3.1	Multiplexing	10
3.3.2	NAND Multiplexing	12
3.3.3	Theory	13
3.3.4	Reliability of Multiplexed Systems	13
3.4	Reconfigurable Computers	15
3.4.1	Theory of reconfigurable fault-tolerant chips	17
3.5	Discussion	20
4	Current status of nanoscale devices for information processing and computing	21
4.1	About the Status Table	21
4.2	Short descriptions and references for the emerging devices status table	23
	References	32

1 Preface

Device scaling, combined with the scaling of interconnects, has allowed more components to be placed on a single chip. At the same time devices with reduced dimensions have shorter delays. These trends have so far produced an exponential increase in the performance and functionality of computers. However, more and more research is being devoted in pursuit of alternatives to metal oxide semiconductor field effect transistor (MOSFET) devices for computing applications, since MOSFET technology will inevitably approach its scaling limits [1]. In addition, some new concepts and devices offer possible advantages in terms of reduced circuit complexity and power dissipation. Some of the more promising ideas include resonant tunnelling diodes (RTDs), electronic quantum cellular automata (electronic QCAs), single electron transistors (SETs) and molecular electronic devices [2].

We have concentrated on digital circuits and architectures, because almost all existing computers use binary digital logic. We chose a memory-adder combination as a representative example of such circuits. In the First Annual Report (July 1999) we concentrated on developing and testing an HSPICE complementary metal oxide semiconductor (CMOS) model and a QCA memory-adder model. The CMOS model was developed in order to obtain 'benchmark' values for CMOS circuit performance to compare with the newer nanoelectronic circuits. During the period December 1999 – May 2000, we focused our research on RTD memory systems, because RTD adder circuit models were developed by the U. Dortmund group and were presented in the Second Annual Report. We investigated the main operational parameters of RTDs when they are integrated into very large scale computing systems. In particular we compared the scaling behaviour of III-V RTD based memory-adder systems with those designed in the dominant technology, silicon CMOS.

In the third year, effort was concentrated on SETs and on fault-tolerance. The work reported in this and previous documents has largely assumed that the devices and structures are perfect. However, there are numerous sources of both hard and soft errors, including manufacturing defects, thermal fluctuations, background charge effects, variations in electron statistics, cosmic rays and radioactive materials. Some preliminary work was carried out into the effects of radioactive materials and cosmic rays on nanoelectronic structures, but this is as yet incomplete. Here we report on the more general topic of how fault-tolerant computers can be designed to cope with various levels of defects or errors that might arise during manufacturing or in service.

2 Performance comparison of CMOS, RTD, SET and QCA devices against a standard computing task

Most of the work reported in this sub-section has been described in greater detail in the Second Annual Report. Since this work is a necessary part of this deliverable, we repeat here the main concepts and results, together with some new results on SETs.

2.1 Benchmark Architecture

The maximum operational frequency of a computational architecture is determined by the signal distribution along some critical data path. In conventional microprocessors this critical path is almost invariably register to register arithmetic, but with large on-chip cache memories, memory access may prove the limiting factor. We set up a benchmark architecture (see Fig. 3.2 in the Final Report, or Fig. 5 in Second Annual Report), that we have modelled in silicon CMOS and simulated with HSPICE to provide a standard for comparison with equivalent RTD/HFET, QCA and SET circuits. This is a simplified model of part of a microprocessor, where two integer words of length L_{word} are collected successively from memory, added together and then returned to memory. Each word is collected in a latch at the base of the memory and then sent to the adder. The reverse process occurs for writing the final answer back to memory.

The benchmark memory structure chosen is shown in the Second Annual Report, Fig. 1. The memory consists of blocks, each containing N_{word} bits. There are L_{word} blocks laid out in a square, each representing one bit of the integer word, so that extracting a word from the memory involves each of the blocks being accessed in parallel. The word is collected in a latch at the base of the memory and then sent to the adder. The reverse process occurs for writing the final answer back to memory. The maximum operational frequency is determined by the signal distribution along some critical data path. We have identified and analyzed the potential signal delay through such a data path. We ignore all details of address generation.

- The *CMOS* memory cell is the standard six-transistor cell, and the adder is the Carry Lookahead Adder (CLA) - see First Annual Report.
- For the *RTD* system, we used the low-power, tunnelling-based SRAM (TSRAM) cell design described in the Second Annual Report, and the results of the Dortmund group for a pipelined N -bit CLA [4].
- For the *SET* system we assumed the same memory organisation as for CMOS and RTD/HFET, but with an SET memory cell. The adder design follows the cellular automata concept. See Final Report for detailed descriptions of the Yano-cell and basic block for the adder - semicell - of the cellular automata concept, shown in Fig. 1.
- For the *QCA* system, two circuit designs for memory and adder, shown in Fig. 2, were used.

For CMOS, RTD, HFET and interconnects we used appropriate circuit models of these devices and then simulated the circuit response in HSPICE [9]. The implementation of the QCA memory-adder system required a few changes of approach. First, we explicitly modelled the address decoding, as this has a significant dependence on N_{word} . Addresses are carried as a pipelined binary string to each of the SRAM units, and are decoded and distributed within the unit by circuitry at the unit's corner. Second, the addition is performed by a simple ripple carry adder, which comprises a serial string of full adders. A carry lookahead adder would be inappropriate, due to the micro-pipelined nature of QCA data flow. The QCA circuitry is micro-pipelined using the adiabatic clocking scheme laid out in [7].

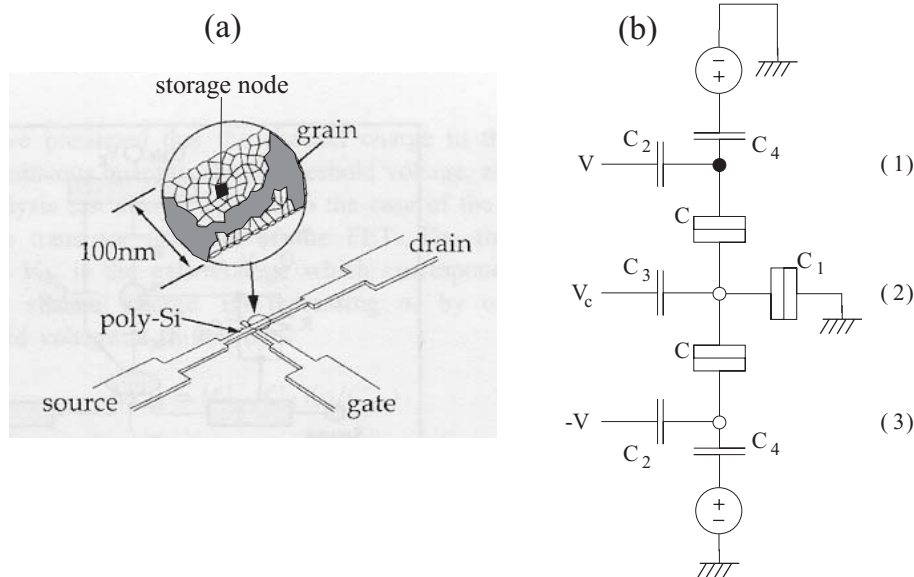


Figure 1: (a) Fabricated Yano memory cell and enlarged active part of poly-Si ultra-thin layer (3nm) with a storage dot (roughly up to 10nm in 'diameter') - after [5]. (b) Semicell (from the Universita di Pisa) - three islands are connected to one another by tunneling junctions and to external voltage sources via capacitors.

For QCAs the number of phase regions the data have to cross will govern the time for signal propagation, see Fig. 2. This contrasts with the CMOS implementation, where memory access and addition can each be performed within a single clock transition. The length of a clock transition will depend on the largest value for the number of cells in a clock region. The limiting case for the value for N_{cells} is taken as the longest path in a region. All of the control signals for the SRAM block require four clock transitions to cross a memory cell, and the addition process requires three clock transitions.

In order to estimate the computational times involved in accessing the memory and carrying out the addition operation, an analytical model was developed. The model assumes four-phase adiabatic clocking of four-well QCAs: full details are provided in the First Annual Report. The model establishes a general relationship which gives an estimate of the time necessary to allow a signal to propagate across a phase region of a wire consisting of N_{cell} cells:

$$t_{\text{clock}} = T_{\text{switch}} N_{\text{cell}}^{1.16} K . \quad (1)$$

This equation is obtained by examining the numerical results of Lent and Tougaw [6]. The largest clock region is the second phase of the memory design (indicated in Fig. 2 by the region shaded in white) and so the limiting choice for N_{cells} is 32. The value taken for K is 3.5, which corresponds to an acceptable level of non-adiabaticity in the system of $\Delta P/P \approx 0.1$.

2.2 Results

The simulations were performed with a sweep over values of N_{word} between 10 and 100,000, with $L_{\text{word}} = 64$ bits. For CMOS and RTD/HFET, the simulations were repeated for minimum feature sizes $\lambda = 0.25 \mu\text{m}$ and $\lambda = 0.05 \mu\text{m}$. We took the same values for the interconnect parameters.

For SETs we chose the Yano memory cell as the basis for calculation (because it is the only one which has been manufactured in large numbers). For QCAs we considered two types of cells. The first were solid-state with inter-dot separation 20nm, inter-cell separation 60nm and switching time (the

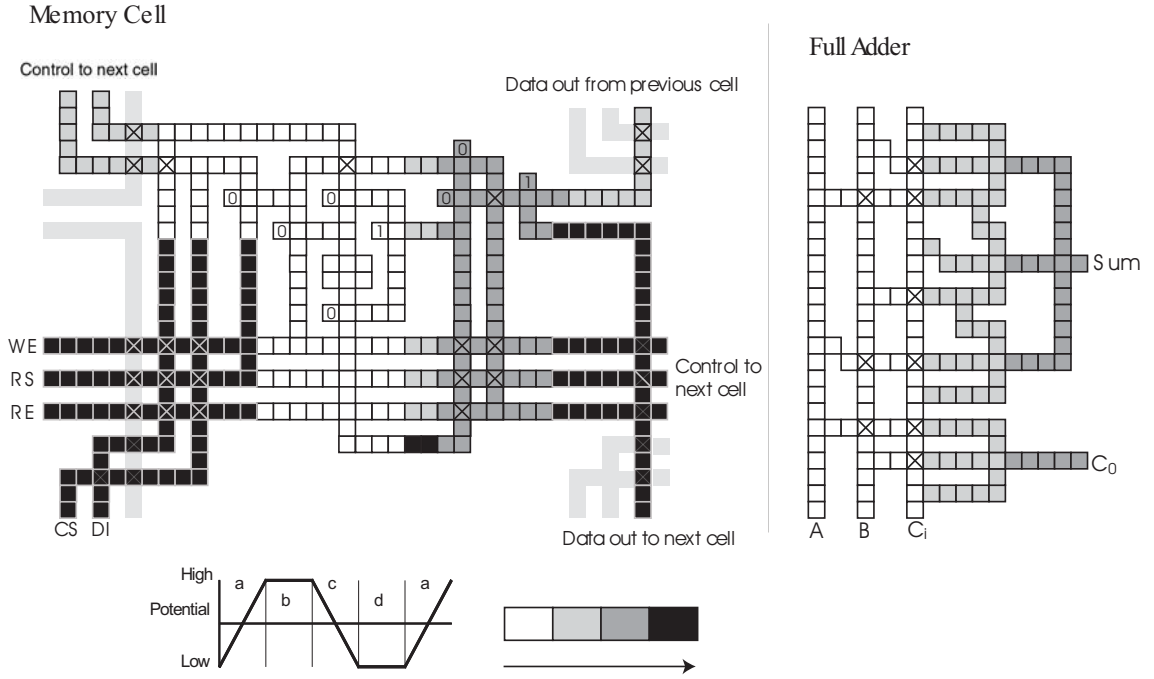


Figure 2: The two circuit designs used to implement the memory-adder model in QCAs. The four phase clock is shown at the bottom of the figure. The grey levels represent the four phases of the adiabatic clock system. WE=Write Enable, RE=Read Enable, RS=Row Select, CS=Column Select, DI=Data In. Cells labelled X represent a wire crossover.

time necessary to allow a signal to propagate across the phase region) $T_{sw} = 2$ ps. Such cells would have to be operated at very low temperature. The second were macro-molecular cells (2nm, 6nm, $T_{sw} = 0.02$ ps), which could in principle operate at room temperature. The operational frequency for CMOS and RTD/HFET systems is defined here as $f = 1/\max\{t_{\text{memory-delay}}, t_{\text{adder-delay}}\}$. In order to compare like with like we have divided the QCA total delay time by four, as the full (read-read-add-write) operation requires four conventional clock cycles. The actual clocking in the case of the RTD adder and the QCA systems ensures proper, synchronised operation of the system and represents a different concept from the operational frequency, although the two are proportional. For both the CMOS and RTD/HFET circuits the results in Fig. 3 show two different regimes. For small memory sizes (less than about 5K words) the speed is limited by circuit delays in the adder, and is independent of memory size.

As the memory size increases, capacitive loading effects in the memory block start to affect the speed. The shape of the curves is different for the QCA circuit because the addition time is almost negligible in comparison with the time to propagate signals to and from the memory. The computational speed is therefore approximately proportional to $N_{\text{word}}^{0.5}$.

The SET analysis differed from that for RTDs and QCAs. A full simulation was not possible (see ANSWERS Report deliverable no. 16). We found that the memory access time, $\sim 10\mu\text{s}$, dominated all other factors. Thus the curve for SETs in Fig. 3 is flat.

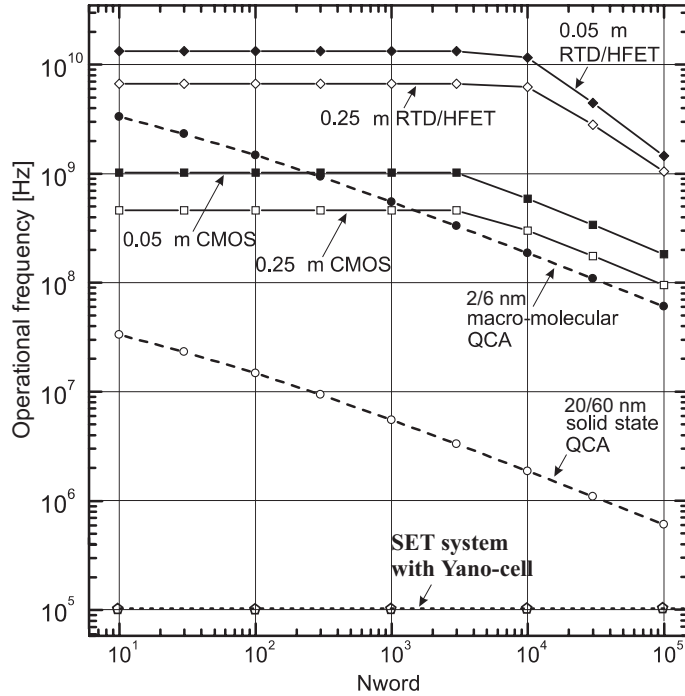


Figure 3: **Operational speed vs. N_{word} for CMOS, RTD/HFET, SET and QCA memory-adder model, for $L_{\text{word}} = 64$ bits. The minimum feature size for CMOS and RTD/HFET is: $\lambda = 0.25 \mu\text{m}$ and $\lambda = 0.05 \mu\text{m}$. Following Lent and Tougaw [6] two types of QCAs were assumed. The first were solid-state with inter-dot separation 20nm, inter-cell separation 60nm and the switching time $T_{\text{sw}} = 2$ ps. The second were macro-molecular cells (2nm, 6nm, $T_{\text{sw}} = 0.02$ ps). Yano-memory cells were used for the SET calculations.**

2.3 Discussion

The performance of the RTD/HFET circuits is an order of magnitude faster than their CMOS equivalents. This is mainly due to the higher speed of the III-V HFET devices in comparison with CMOS. The RTD/HFET circuits also have reduced numbers of components. However, since RTDs are combined with HFETs, the power dissipation is high. For example the calculated standby power dissipation for a TSRAM cell is 7 nW for $\lambda = 0.25 \mu\text{m}$ and 0.5 nW for $\lambda = 0.05 \mu\text{m}$, whereas for silicon CMOS SRAM it is 0.4 pW and 0.04 pW respectively [3]. The scaling (and therefore the sizes of the functional blocks) is only slightly better than in the CMOS case.

From Fig. 3 it can be seen that the performance of QCAs for semiconductor technology at the proposed typical size, is poor in comparison with CMOS. In particular the operational speed is about two orders of magnitude lower than both CMOS technologies. The estimated device densities are an improvement on current CMOS but still lower than $0.05 \mu\text{m}$ technology. The prospects for molecular QCAs are somewhat better. They show comparable speed performance and the possible implementation density is increased by a factor of a few tens. It has to be noted that these results ignore manufacturing or transient errors, which would require extra redundant components to maintain reliability.

The performance of the SET memory-adder model appears to be much worse than any of the other systems. The access time of the memory cell completely dominates the time required to add two numbers. However, it should be noted that we used the Yano memory as a basis for the calculations. This is a relatively slow memory cell, but it is insensitive to background charge effects, it operates at room temperature and it is one of the very few SET devices which has actually been made. An alternative Coulomb blockade memory cell, using small numbers of electrons, has been designed and fabricated at the University of Cambridge [8]. At present it only operates at low temperatures, but it

has a much faster access time ($\sim 10\text{ns}$) than the Yano cell. By reducing its size it should be possible to make it run at room temperature. It should also be noted that these SET based cells offer the advantages of lower power operation than CMOS.

3 Fault-tolerant techniques for nanocomputers

The invention of nanometre-scale devices should eventually permit extremely large scales of integration, of the order of 10^{12} devices per chip. At the present time, only a handful of truly nanoscale or molecular scale logic or memory devices exist, and the question of how to assemble 10^{12} such devices on a working chip seems academic. Nevertheless, it is almost certain that it will be very difficult to make even small nanoscale circuits - for example having one hundred devices - with any degree of certainty. Furthermore, it is probable that the proposed nanoelectronic devices will be more fragile than conventional devices, and be sensitive to external influences such as radiation related effects (radioactive decay or cosmic rays), high temperature, electromagnetic interference, parameter fluctuations etc. Hence if progress is to be made in nanoelectronics, the question of fault tolerance needs to be considered as early as possible.

Here we first examine two generalizations of the well-known technique of Triple Modular Redundancy (TMR), namely:

- R -Modular Redundancy (RMR, $R = 3, 5, 7, 9, \dots$) and
- Cascaded TMR (of the i -th order) - CTMR.

Then we investigate a massive redundancy idea, the so-called multiplexing method, proposed by von Neuman [10] at the time when early computers were introduced which were notoriously unreliable. Finally we present a simplified theory of reconfigurable architecture, which was originally proposed as a defect-tolerant computer architecture (i.e. for tackling manufacturing defects rather than transient errors). The effectiveness of this idea was successfully demonstrated on a massively parallel computer, ‘Teramac’, built at Hewlett-Packard Laboratories [11].

3.1 R -Modular Redundancy

The concept of TMR is to have three units working in parallel, and to compare their outputs with a majority gate (Fig. 4(a), $R = 3$). Then TMR can provide an assemblage that behaves like one of its constituent components, but with an improved probability of working. The tradeoff is that instead of n devices, at least $3n$ devices plus a majority gate, are needed to make this new ‘unit’.

In our analysis we assume a chip to contain N_{total} devices, with a probability p_f of an individual device failing. The probability P_{fail} of a complete chip failing during its working lifetime is minimized under the condition $N_c p_f \ll 1$, where N_c is the number of devices in the constituent unit (module). Imperfect majority gates have B outputs and mB devices. First we assume that a module of N_c devices works only if every single device in the module works:

$$P_{\text{module, works}} = (1 - p_f)^{N_c} \approx e^{-N_c p_f} \quad (p_f \ll 1). \quad (2)$$

Now the probability that a module fails, if $N_c p_f \ll 1$, will be:

$$P_{\text{module, fails}} = 1 - P_{\text{module, works}} = N_c p_f. \quad (3)$$

A group consisting of R modules and a majority gate works correctly when at least $(R + 1)/2$ modules work correctly (with probability $P_{\text{g,w}}$) and when the majority gate also works (with probability $P_{\text{mg,w}} \approx e^{-mB p_f}$, see eq. (2)). The probability that a group fails (P_f^{group}) is then:

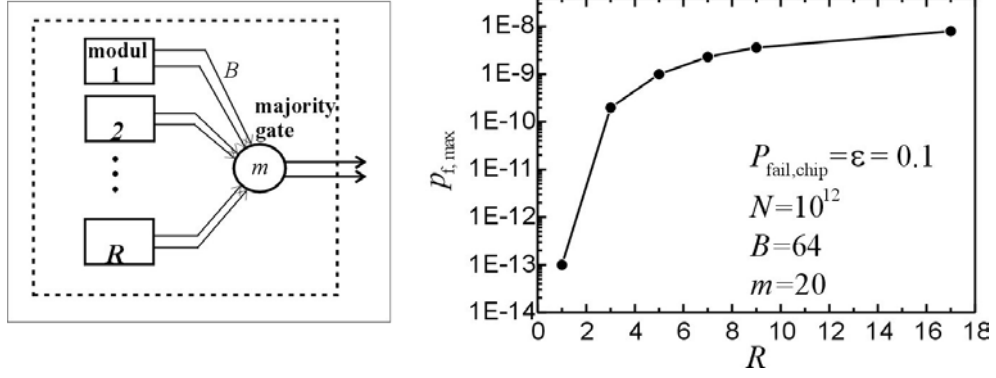


Figure 4: (a) Schematic layout of an R -modular redundancy block, with a majority gate having m devices for each set of inputs. (b) The individual device failure probability p_f required to obtain a chip failure probability of 10%, as a function of the level of redundancy, where the total number of devices on the chip is $N_{\text{total}} = 10^{12}$. The majority gate contains imperfect devices.

$$P_f^{\text{group}} = 1 - P_{g,w}P_{mg,w} \approx 1 - P_{g,w}(1 - mBp_f) = 1 - P_{g,w} + P_{g,w}mBp_f \approx P_{g,f} + mBp_f \quad (4)$$

where the probability that a majority of R modules fails is:

$$P_{g,f} = \binom{R}{(R-1)/2} P^{(R-1)/2} Q^{(R+1)/2} + \dots + \binom{R}{1} P Q^{(R-1)} + \binom{R}{0} Q^R \quad (5)$$

where $P \equiv P_{\text{module, fails}}$ and $Q = 1 - P$. When $Q \ll 1$ equation (5) reduces to the first term, and by using (3), eq. (4) becomes:

$$P_f^{\text{group}} \approx C (N_c p_f)^{(R+1)/2} + mBp_f, \quad \text{where } C = \binom{R}{(R-1)/2} \quad (6)$$

The number of devices in a group is $RN_c + mB$, so the total number of groups is: $N_{\text{groups}} = N_{\text{total}}/(RN_c + mB)$. The probability that the whole chip with N_{total} devices fails is (again when $P_f^{\text{group}} \ll 1$) approximately:

$$P_{fail}^{\text{chip}} \approx N_{\text{groups}} \cdot P_f^{\text{group}} = \frac{N_{\text{total}}}{RN_c + mB} \left[C (N_c p_f)^{(R+1)/2} + mBp_f \right]. \quad (7)$$

The equation $dP_{fail}^{\text{chip}}/dN_c = 0$ gives the optimum module size (N_c) for a given p_f , which substituted in (7), yields the minimum chip failure probability. In Table 1 are given results for some values of R .

Fig. 4(b) demonstrates the effectiveness of RMR. For example with redundancy level $R = 5$ we can achieve the same level of chip reliability, but with four orders of magnitude less reliable devices. The tradeoff for this improvement is that the effective number of devices is reduced to $N_{\text{total}}/5$.

3.2 Cascaded Triple-Modular Redundancy

The TMR process can be repeated by combining three of the TMR ‘units’ with another majority gate to form a ‘second-order’ TMR unit with even higher reliability (a technique called CTMR). A detailed analysis of the CTMR technique was presented in the ANSWERS Second Annual Report. Here we repeat the main results. The probability that a CTMR configuration of i th order works is:

R	$P_{\text{chip}}^{\text{fail}} = \epsilon$	$p_{f,\text{max}}$	optimum modul size N_c
1	$N_{\text{total}}p_f$	$\epsilon/N_{\text{total}}$	N_{total}
3	$N_{\text{total}}p_f \cdot 1.15(mBp_f)^{1/2}$	$\frac{0.9}{(mB)^{1/3}} (\epsilon/N_{\text{total}})^{2/3}$	$\left(\frac{mB}{3}\right)^{1/2} (p_f)^{-1/2}$
5	$N_{\text{total}}p_f \cdot 0.8(mBp_f)^{2/3}$	$\frac{1.1}{(mB)^{2/5}} (\epsilon/N_{\text{total}})^{3/5}$	$\left(\frac{mB}{20}\right)^{1/3} (p_f)^{-2/3}$
7	$N_{\text{total}}p_f \cdot 0.6(mBp_f)^{3/4}$	$\frac{1.3}{(mB)^{3/7}} (\epsilon/N_{\text{total}})^{4/7}$	$\left(\frac{mB}{105}\right)^{1/4} (p_f)^{-3/4}$
9	$N_{\text{total}}p_f \cdot 0.5(mBp_f)^{4/5}$	$\frac{1.5}{(mB)^{4/9}} (\epsilon/N_{\text{total}})^{5/9}$	$\left(\frac{mB}{500}\right)^{1/5} (p_f)^{-4/5}$
\vdots	\vdots	\vdots	\vdots
limit	$\sim N_{\text{total}}p_f \cdot (mBp_f)$	$\sim \frac{1}{(mB)^{1/2}} (\epsilon/N_{\text{total}})^{1/2}$	$\sim 1/p_f$

Table 1: Efficiency of R -modular redundancy at reducing chip failure rates ($N_cp_f \ll 1$).

$$P_w^{(i)} = (1 - p_{\text{fail}})^{mB} \left[\left(P_w^{(i-1)} \right)^3 + 3 \left(P_w^{(i-1)} \right)^2 \left(1 - P_w^{(i-1)} \right) \right] \quad (8)$$

where mB is the number of devices in a majority gate with imperfect devices. In Fig. 5 is shown the effectiveness of the CTMR technique. It shows that there is no advantage in using CTMR for units containing a small number of devices, when the majority gates are made from the same devices as the units that they are monitoring. However, at least in principle, improvement is possible for units with large values of N_c .

There are three regions in each set of curves:

- a) $N_cp_f > \ln 2$, where redundancy affords no advantage,
- b) $10^{-3} \lesssim N_cp_f < \ln 2$, where redundancy is most effective, and
- c) $N_cp_f < 10^{-3}$, where only first order redundancy offers an advantage.

In case (b), the effectiveness of redundancy scales as a power law with the order of CTMR. The failure probability is:

$$P_{\text{fail}}^{(i)} \propto (N_cp_f)^{2i}.$$

For the case (c) the effectiveness of redundancy depends on the ratio mB/N_c . Starting from equation (8), it can be shown that in region (c) the failure probabilities are:

$$\begin{cases} P_f^{(0)} \equiv P_{\text{fail}} \approx N_cp_f \\ P_f^{(i)} \approx \frac{mB}{N_{\text{total}}} N_cp_f = \frac{mB}{N_c} P_f^{(0)}, \quad i = 1, 2, \dots \end{cases} \quad (9)$$

3.3 NAND Multiplexing

In this section we describe an unusual technique for providing fault-tolerance. First described by von Neumann in 1956 [10], but little used since, it appears to offer the highest possible degree of protection against transient faults. It also provides protection against manufacturing faults, but in this case requires a large number of redundant components.

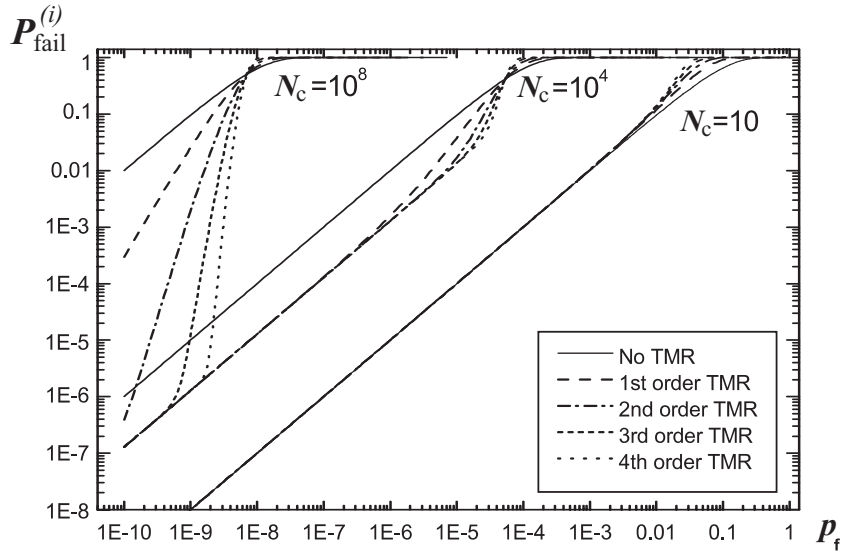


Figure 5: The probability of obtaining a defective (TMR/CTMR) unit ($P_{\text{fail}}^{(i)}$) (with $3^i N_c$ devices) with B -bit outputs, as a function of the individual device failure probability p_f , using imperfect majority gates. The groups of curves are for: $N_c = 10$, $B = 1$, $m = 10$ (right), $N_c = 10^4$, $B = 64$, $m = 20$ (centre), and $N_c = 10^8$, $B = 64$, $m = 20$ (left).

3.3.1 Multiplexing

Consider an information-processing unit with an arbitrary number of inputs and outputs. Let the processing unit be constructed of elementary devices, each with a probability of failure p_f . Let the resulting probability of failure of the entire unit be ε . It is clear that at best the entire unit, or larger units constructed from this basic processing unit, cannot have a failure rate lower than that of a constituent device. This is because each output line must originate from at least one elementary device and thus the output cannot have an error rate smaller than that of the device. *Multiplexing* is a fault-tolerant architecture that is designed to remove this restriction on the reliability of information processing units, constructed of unreliable devices.

The basic technique of multiplexing is to replace within a network, processing units of any size by multiplexed units containing N_{bundle} number of lines for every single input and output. Fig. 6(a) depicts an imaginary network comprised of identical processing subunits. Fig. 6(b) then shows the multiplexed equivalent of the network for $N_{\text{bundle}} = 4$ (*N.B.* the units to be multiplexed do not have to be identical to one other as depicted here). The multiplex units have many redundant devices inside them, and they process the input lines (which ideally carry identical bits) in parallel to give a set of output lines for each output. If the inputs and processing units are perfectly reliable then the lines comprising each output should be identically stimulated (1) or unstimulated (0). However, due to errors in the input data as well as errors occurring in the processing of the inputs from faulty devices, not all of the output lines in each output will be identically stimulated. Thus for multiplexed networks, the final outputs are considered to be 1 if $\geq (1 - \Delta) \cdot N_{\text{bundle}}$ lines are stimulated and 0 if $\leq \Delta \cdot N_{\text{bundle}}$ lines are stimulated, where Δ is a critical level that is pre-defined ($0 < \Delta < 0.5$). Stimulation levels in between are considered to be undecided (consequently resulting in malfunction).

The aim of multiplexing is to ensure that the output of a network of processing units is not affected by the failure of a small number of basic units making up the network. We now describe the basic design of the multiplex units. Essentially, a multiplex unit consists of two stages. One, the *executive stage*, performs the basic function of the processing unit in parallel. The second, the *restorative stage*,

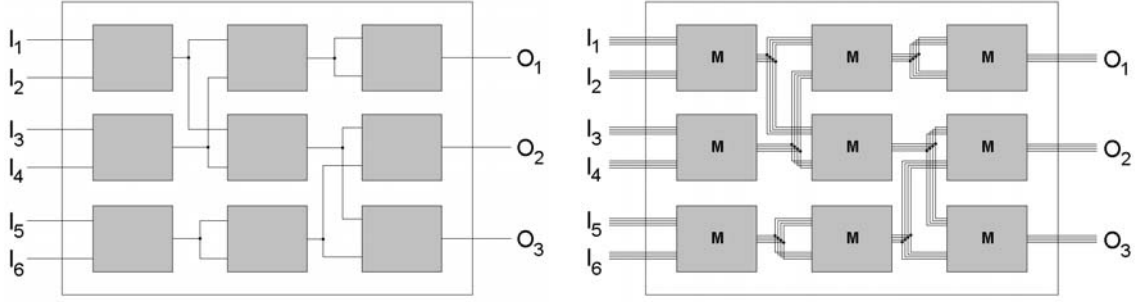


Figure 6: (a) Network of basic processing units making up a larger processing unit. (b) Network of multiplexed basic processing units (M) making up a larger multiplexed unit ($N_{bundle} = 4$).

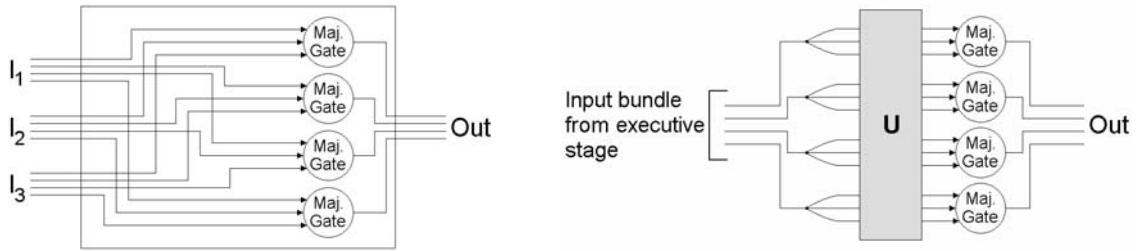


Figure 7: (a) The executive stage of a majority gate multiplex. (b) The restorative stage for a single output bundle of a $N_{bundle} = 4$ executive stage. Together, the executive and restorative stages in series with each output bundle comprise the multiplex unit.

reduces the degradation caused by the executive stage and thus acts as a non-linear amplifier of the output. To illustrate the need for the restorative stage, we will consider the multiplex equivalent of a majority gate. If we desire N_{bundle} to be 4, then the executive stage for the majority gate is that shown in Fig. 7(a). The majority gate gives an output that is the majority of the inputs. If most of the input lines for the three inputs are stimulated/unstimulated then most of the output lines will be stimulated/unstimulated. Consider now the case for an executive stage with $N_{bundle} = 5$ where the critical level is set at $\Delta = 1/5$. If now four of the input lines are stimulated in two of the input bundles and none of the lines are stimulated in the last input bundle then it is possible for the output to have only three lines stimulated. This would then lie outside the critical stimulation level of $\Delta \cdot N_{bundle} = 4$ and thus be interpreted as a malfunction. In comparison, for a single line majority gate, the corresponding input would give an unequivocal value of 1 for the output. It is thus clear that some means of restoring the output stimulation level is required. Thus as can be seen, in order to construct a multiplex unit, a restorative stage must be attached to the output of the executive stage.

A restorative stage for a multiplex unit is shown in Fig. 7(b). Each line of an output bundle of the executive stage is split into a triad and passed through a randomiser U that randomly connects the triads to the inputs of N_{bundle} number of majority gates. The majority gate is in itself a kind of amplifier of its input signals and thus a parallel array of them can amplify the signals from the executive stage, provided the signals are statistically uncorrelated (hence the need for the randomiser). It is sufficient to assume the randomiser to be a complicated permutation of the lines. Thus if the input level of stimulated lines is close to all 0 or all 1, then the restorative stage acts to bring the corresponding output even closer to its ideal of all stimulated or all unstimulated.

Note that each output bundle from the executive stage requires a separate restorative stage. Also note that the restorative stage depicted is general and can be used to multiplex any processing unit, not

just a majority gate.

Thus as we have seen, the multiplex analogue of a majority gate in our example performs exactly the same function as a single majority gate, but with an improved probability of operation provided that ε , the probability of failure of a majority gate, is small. The same technique can be used to multiplex a processing unit of arbitrary size and with arbitrary number of inputs and outputs, by building an executive stage and attaching the restorative stage shown in Fig. 7(b) to each output bundle. For processing units consisting of a large number of elementary devices, the resulting redundancy of a multiplex unit is $\sim N_{bundle}$.

3.3.2 NAND Multiplexing

The question arises as to the optimum circuit level for applying the technique of multiplexing. We want multiplexing to maximise the device failure rate p_f , that can be tolerated in a high-density chip that has a maximum acceptable bound for failure. As we have seen, multiplexing is effective only for small ε . As ε for a processing unit containing N_u devices is given by:

$$\varepsilon = 1 - (1 - p_f)^{N_u} \quad (10)$$

then ε can be minimised for a large p_f by minimising the number of devices N_u in a processing cluster. Thus multiplexing works best by applying it to the most basic processing unit possible in a circuit. In microprocessors this is the logic gate. In conventional architectures, the most basic logic gates are NAND, NOR and NOT. If multiplexing is applied to such conventional architectures, matters become complicated, as then three different multiplexing structures would need to be implemented. For the sake of simplifying calculations, we will assume the most basic logic gate in a chip to be the NAND gate. This is a universal logic gate and can be used to build the basic logic gates NOT and NOR (containing one and four NAND gates respectively). Thus a NAND network equivalent can replace any conventional architecture. The use of only NAND gates in extremely large-scale integration (XLSI) architectures may also be justified in that then construction is simplified through the use of identical repeating subunits.

In conventional CMOS circuits, four devices each are required to construct NAND and NOR gates whilst two are required to construct a NOT gate. In a NAND network, the number of devices required would then be 4, 4 and 16 respectively for NAND, NOT and NOR. Thus approximately double the number of devices is required for a NAND network than for conventional architectures. However, this figure does not take into account the fact that simpler networks of NAND gates can sometimes be found for conventional networks, where NOR and NOT are simply replaced by their NAND equivalents.

We now need to consider the construction of the multiplex equivalent of NAND (the *NAND multiplex*). Using the method outlined before, this would imply the construction of an executive stage to perform the function of NAND, the output being passed through the restorative stage shown in Fig. 7(b) for signal amplification. However, the construction of a majority gate requires four NAND gates, and thus, in the interests of minimising redundancy, the question of whether the restorative stage can be constructed from fewer NAND gates must be asked. Indeed, it turns out that this is possible by simply replacing the majority gates in Fig. 7(b) with NAND gates. The output lines from the executive stage then split into doubles instead of triads and pass through a randomiser as before, subsequently entering N_{bundle} parallel NAND gates (see Fig. 8).

Although such a restorative stage amplifies the input signal as required, it also inverts it. It is thus necessary to place two of the described restoring circuits in series in order to revert the signal back to its correct stimulation (nearly all 0 or nearly all 1). These together thus comprise the final restorative stage for a NAND multiplex. This is essentially the structure that was described by von Neumann. Fig. 8 shows the complete NAND multiplex for $N_{bundle} = 4$. Note that NAND multiplexing introduces a redundancy of $3N_{bundle}$.

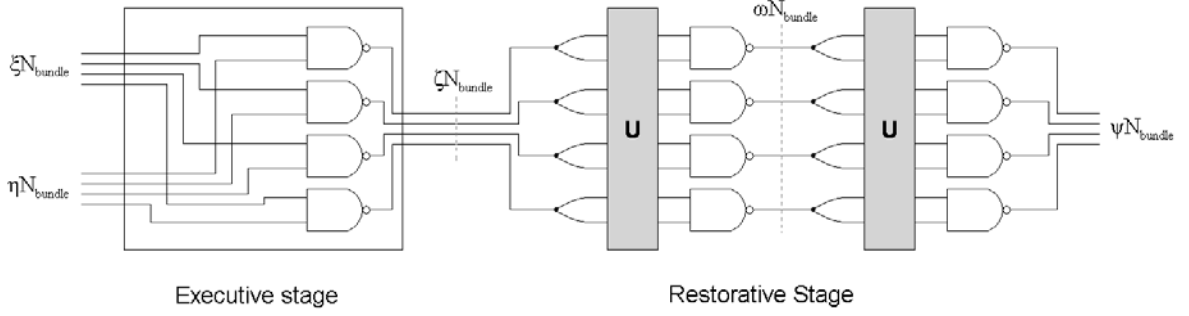


Figure 8: **The NAND multiplex.** The executive stage is constructed as in Fig. 7(a). The restorative stage is different to Fig. 7(b), as this method uses fewer NAND gates in its construction. Note also that the restorative stage is iterated due to its inverting effect. ξ and η represent the input stimulation fractions, whilst ζ and ω represent the stimulation fractions at the intermediate stages. ψ represents the stimulation fraction of the final output.

3.3.3 Theory

It is possible to develop a quantitative theory for NAND multiplexing whereby the probability distribution of the output stimulation fraction ψ , of a NAND multiplex can be calculated as a function of the input stimulation fractions, ξ and η , taking into account a probability of failure ε in each NAND gate. If ζ and ω are the stimulation fractions of the executive stage and intermediate restorative stage outputs (see Fig. 8), then probability distributions Φ_ζ , Φ_ω , and Φ_ψ can be derived for ζ , ω and ψ respectively. For large N_{bundle} , these distributions are approximately Gaussian and indeed the theory is only valid for large N_{bundle} due to this assumption. Φ_ζ is a function of ξ and η , whilst Φ_ω is a function of the mean value of ζ . Similarly, Φ_ψ is a function of the mean value of ω . Thus the final distribution for ψ , Φ_ψ , is obtained by iteration as described.

For details of the theory developed by von Neumann and the formulae for Φ_ζ , Φ_ω , and Φ_ψ see [10].

3.3.4 Reliability of Multiplexed Systems

We now present the scheme for calculating the reliability of an $N_{total} = 10^{12}$ device nanochip implemented using NAND multiplexing. It is of course difficult to speculate on the architecture that would be implemented in future nanochips, but one plausible idea is that instead of a highly complex system of logic and memory spanning 10^{12} devices, a nanochip would simply consist of hundreds to thousands of present day high-end microprocessors on one chip, processing different parts of a task independently, before integrating their outputs through a final processor. Alternatively, the chip might contain large numbers of smaller processing elements, containing perhaps $10^3 - 10^5$ devices, connected in a rectangular grid. Such a structure might be used as a neural network or for image processing. In our model for calculating the reliability of a NAND multiplexed nanochip, we will thus assume that a nanochip with N_{total} devices is divided into processing clusters of *effective* device count N_c , each calculating a task independently. As explained earlier, NAND multiplexing introduces a redundancy factor of $3N_{bundle}$. Thus the total number of devices on a nanochip is related to N_c by:

$$N_{total} = n \cdot N_c \cdot 3N_{bundle} \quad (11)$$

where n is the number of processing clusters, each of effective device count N_c , that can be packed onto the nanochip. Note that the *actual* number of devices in each NAND multiplexed processing cluster is $3N_c \cdot N_{bundle}$.

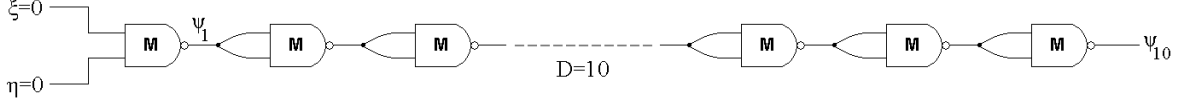


Figure 9: A simple NAND multiplex circuit of logical depth 10. The input bundles are both assumed to be all unstimulated.

The maximum logical depth D for logic circuits of reasonable size is typically of order 10. A circuit as small as a full-bit adder has a maximum logical depth of 6, whilst conventional microprocessors are typically designed to have a maximum logical depth of about 10. We will thus assume the processing clusters of effective size N_c to have a fixed logical depth $D = 10$. This assumption of course fails for very small device counts, but it is reasonable even for circuits with as few as around ten logic gates.

To calculate the error in processing perfect inputs into a cluster, we consider a simplified model for a logic circuit of depth 10, as shown in Fig. 9. We assume the inputs to be perfectly reliable and set the lines in both input bundles to all 0. If the NAND gates work perfectly, the output stimulation fraction should be $\psi_{10} = 1$ (the corresponding probability distribution being a delta function), since a conventional chain of NAND gates arranged as shown would give an output of 1 with both inputs 0. If the NAND gates have a finite probability of failure ε per operation, then the fraction ψ_{10} will take up a Gaussian probability distribution as described earlier. In our calculations we use equation (10) to calculate ε . Although the nature of devices in future nanochips is unknown, a reasonable estimate for the number of devices in a NAND gate is $N_u = 4$ (as with CMOS devices), and thus equation (10) becomes:

$$\varepsilon = 1 - (1 - p_f)^4 \quad (12)$$

As outlined before, numerically iterating Φ_ζ , Φ_ω , and Φ_ψ given ξ and η , and ε from (12) gives us the probability distribution for ψ_1 . To calculate the distribution for ψ_2 , the mean of the result for ψ_1 is fed into the iteration again such that $\langle \psi_1 \rangle = \xi = \eta$. Repeating this procedure ten times thus gives us $\Phi_{\psi_{10}}$, the probability distribution of ψ_{10} .

Fig. 10 shows the distribution curves for Φ_{ψ_1} and $\Phi_{\psi_{10}}$, where the values for N_{bundle} and ε have been chosen for clarity. One can see that even after a logical depth of 10, the distribution curve spreads only slightly.

The output ψ_{10} represents that of one output bit from the processing cluster. We now wish to calculate the reliability of the data from this output bit. A reasonable method is to calculate the area under the curve of $\Phi_{\psi_{10}}$ inside the stimulation fraction $(1 - \Delta)$ and divide it by the total area under the curve. Thus:

$$P_{w-bit} = \frac{\text{area under } \Phi_{\psi_{10}} \text{ inside } (1 - \Delta)}{\text{area under } \Phi_{\psi_{10}}} \quad (13)$$

If the processing cluster has an m -bit output, the reliability of the of the processing cluster is then given by:

$$P_{w-cluster} = P_{w-bit}^m \quad (14)$$

Since the number of processing clusters, n , on a chip is given by equation (11), the reliability of the whole nanochip is then given by:

$$P_{w-chip} = P_{w-cluster}^n \quad (15)$$

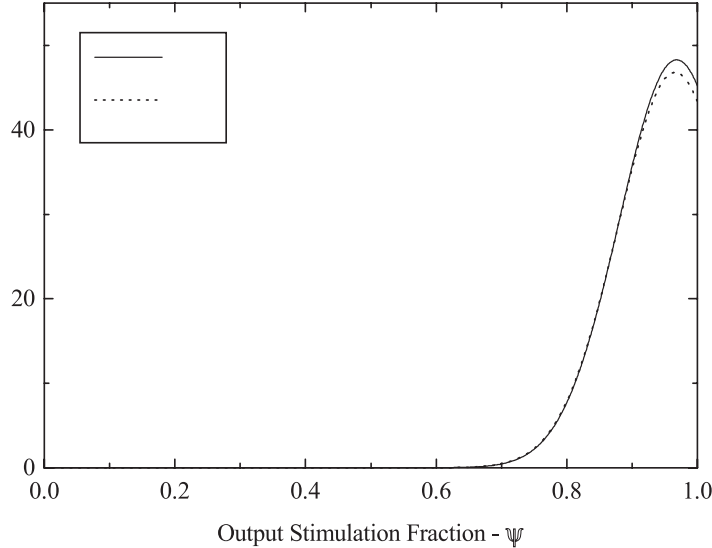


Figure 10: Plot of Φ_{ψ_1} and $\Phi_{\psi_{10}}$ for $N_{bundle} = 1000$ and $\varepsilon = 0.015$. As N_{bundle} approaches ∞ and ε approaches 0, the Gaussian distributions approach delta functions.

In investigating the suitability of multiplexing as a fault-tolerant strategy for nanochips, we are interested in looking at how the maximum tolerable failure rate per device, p_f , varies with effective redundancy R , where the failure rate of the nanochip, P_{f-chip} , is fixed at a reasonable level:

$$P_{f-chip} = 1 - P_{w-chip} = 0.1 \quad (16)$$

One can then produce a plot of p_f vs. R with the reliability constraint in equation (16), using the scheme for calculating chip reliabilities outlined earlier. This plot for various N_c , is included in Figs. 12 & 13.

It must be noted that the assumption of a Gaussian distribution is incorrect when the centre of the distribution approaches 0 or 1 closely, due to the cut-off in the function (see Fig. 10). It has not yet been possible to develop a rigorous analysis to take this phenomenon into account, in addition to extension of the validity of the results to small N_{bundle} .

3.4 Reconfigurable Computers

This section is concerned mainly with examining the concept of reconfigurability, which is one way to maximise the probability of obtaining a working chip, even though the chip, as manufactured, may have a very large number of dead transistors or other devices. The assumption is made that it is possible to examine the chip and to locate the dead devices by some means. Once the dead devices have been located, then the further assumption is made that it is possible to avoid these dead devices and to build a working system from the remaining ‘live’ devices.

The concept of reconfigurability has been used by computer manufacturers for many years, with great success. For example, when memory chips are built, a few spare columns are included. If a defective column is found during testing then one of the spare columns is switched in to replace it.

This technique is less successful for use with dedicated logic circuits, which are usually much less regular in their layout than memory circuits. For example, it would be unrealistic to suppose that a spare multiplier circuit could be included on a processor chip, if the probability of failure of the ‘primary’ multiplier is small. If the probability of a chip failing is only a few percent, or a few tens

of percent, then it is sometimes cheaper to test each chip for functionality, and to throw away the ones that do not work.

These two strategies work well, as long as the quantity $N_{total} \cdot p_f$ is $\lesssim 1$, where N_{total} is the total number of transistors on a chip and p_f is the probability of an individual transistor not being fabricated properly.¹

At present, the number of transistors on the main chip for a workstation or PC is 20 to 50 million, and about 250 million on a large DRAM chip. By taking extreme care in manufacturing, it is possible to obtain failure rates per transistor in the range $p_f \sim 10^{-9}$ to 10^{-7} . Thus the test-and-reconfigure or test-and-discard strategies described above are both feasible.

As conventional CMOS devices get smaller and smaller, it is becoming harder and harder to maintain these relatively good (i.e. small) values of p_f , and many clever techniques have been developed to improve manufacturing reliability in particular and computer performance in general. Amongst these is the concept of the reconfigurable computer.

The reconfigurable computer concept is based on the use of field programmable gate arrays (FPGAs). The latest FPGA chips are extremely sophisticated and complicated structures (cf. e.g. [13]), but the basic principles are relatively straightforward. Fundamentally, a FPGA contains a large, regular grid of logic units, which are often called configurable logic blocks (CLBs). Each CLB is the same size, and contains between 100 and 1000 transistors. The CLBs are organised so that they can be individually reprogrammed. One CLB might be programmed to behave like a four-input NAND gate, another might be programmed to behave like a 128-bit random access memory, and so on. Each CLB can exchange data with its north, south, east and west neighbours, and the CLBs are further grouped in blocks, then blocks of blocks - for example 4 by 4, then 16 by 16 or larger, so that data can be sent rapidly over large distances between blocks.

FPGAs have many advantages. They can be programmed (and re-programmed) relatively rapidly, and a wide variety of more-or-less arbitrary logic or memory structures can be mapped onto the array of CLBs. It is also relatively easy to check whether part or all of a CLB is working or not. Thus defective transistors can be located, at least to within the boundaries of one CLB or another.

This flexibility comes with some disadvantages. One limitation is that the density of the devices in a FPGA is only 10% to 15% of what is available with dedicated circuits. There is therefore a 'spatial' redundancy factor of 7-10. A second limitation, which varies in amount with from circuit to circuit, is that it is usually not possible to use all of the CLBs and to connect them so that none is wasted. In one design this 'device packing' redundancy multiplier was reported to be 1.5 [12]. In what follows we assume for simplicity that the inherent redundancy due to the combination of these two effects has a value of 10, irrespective of the particular circuit that is being considered.

It is obvious that there are benefits in using FPGAs to provide protection against manufacturing defects, and numerous papers have been published about this. In the context of nanoelectronics, the most famous of these is a paper on the Teramac computer [11]. This paper described the design, manufacture, testing and programming of a machine that used 864 FPGA chips, that were deliberately built to a relatively low standard, with a reported total of 22000 manufacturing faults. Special fault-detection and re-routing software enabled the system to be reconfigured and to be programmed to solve a variety of test algorithms. The implications for the manufacture of future fault-tolerant nanocomputers were emphasised.

Although the Teramac paper has achieved some mild notoriety, it does not provide enough information to permit a theoretical analysis of the relation between the number of devices on a chip, the component layout, the manufacturing defect probability per transistor (p_f) and the probability of obtaining a working chip or computer. To do this we turn instead to a more informative paper by Lach et al. [12]. In the next section we extend the theory given in [12], in order to provide first-order estimates

¹It should be noted that defects in wiring, capacitors or other components are not discussed in this section, although the theory could be extended to cover these other items.

of the ultimate limits to the performance of nanocomputers that use reconfigurability to overcome manufacturing defects.

3.4.1 Theory of reconfigurable fault-tolerant chips

In what follows we confine our discussion to a hypothetical single-chip nanocomputer, and consider only uncorrelated manufacturing defects in individual devices ('transistors'). We do not discuss the extension of the theory to multi-chip systems, correlated defects, wiring defects, or systems with mixtures of dedicated and reconfigurable components. The terminology is that of [12].

Fig. 11 illustrates the chip model. Devices ('transistors'), each having a probability p_f of being defective during manufacture, are assembled in groups of N_{trans} to form a configurable logic block (CLB). A number of these CLBs are grouped together, N_c at a time, to form an *atomic fault-tolerant block* (AFTB). It is assumed that the AFTB can be configured to perform some basic set of operations, even though any one of its constituent CLBs may be faulty. In general, different types of AFTBs can be designed to carry out different functions, and each type may incorporate different numbers of CLBs. However, it is assumed here that all AFTBs contain the same number of CLBs.

The AFTBs are then grouped together, N_A at a time, to form a cluster which then performs some desired function. In the present context we suppose that a quite high-level function is implemented. For example, the cluster might operate as if it contained a large block of memory, or a 64-bit full adder, or a processing element for an artificial retina, or even the equivalent of a present-day workstation CPU chip. On the other hand, the cluster function could be much simpler.

One further stage is needed. It is assumed that the chip is completely filled with identical, independent copies of the cluster. Suppose that we would like to be able to manufacture chips, so that after fault-detection and reconfiguration, they have a certain probability - for example 90% - of working. What we are interested in, is how large a value of p_f , the defect probability per device, can be tolerated.

The analysis given in [12] assumes that each CLB has a certain failure probability, and calculates the failure probability of a cluster: the discussion does not extend down to the device level or up to the chip level (except by implication). In what follows we extend their analysis down to the individual device level, and to a higher level of redundancy.

Consider a CLB, containing N_t transistors. The probability P_{clb-w} that the CLB works is:

$$P_{clb-w} = (1 - p_f)^{N_t} \quad (17)$$

An AFTB, containing an arbitrary number N_c of CLBS ($N_c \geq 2$) is considered suitable for reconfiguration if all of the CLBS work, or if only one of the CLBs is defective (it is assumed here that the AFTB can then be reconfigured). Thus the effective probability of the AFTB to work, P_{aftb-w} , is given by:

$$P_{aftb-w} = (P_{clb-w})^{N_c} + N_c(P_{clb-w})^{N_c-1}(1 - P_{clb-w}) \quad (18)$$

and the 'failure' probability of the AFTB, P_{aftb-f} , is given by:

$$P_{aftb-f} = 1 - P_{aftb-w} \quad (19)$$

If N_A AFTBs are grouped together to form a cluster, which we want to work like an adder, multiplier, memory block, etc., then the probability that the cluster fails, $P_{cluster-f}$, is given by:

$$P_{cluster-f} = 1 - (P_{aftb-w})^{N_A} \quad (20)$$

We now apply a higher level of redundancy, by grouping the clusters together, R at a time, to form a supercluster. For simplicity in this first-order analysis, we assume that a *deus ex machina* can detect

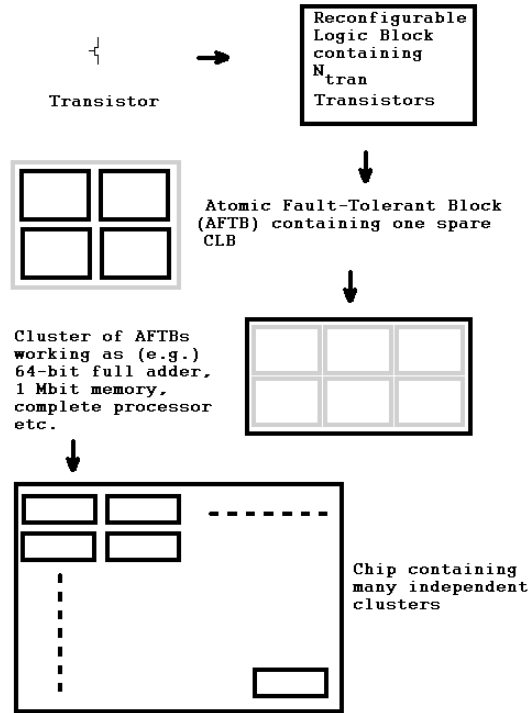


Figure 11: The lowest-level logical unit in a *field programmable gate array* (FPGA) is the *configurable logic block* (CLB), which typically contains $\sim 100 - 600$ transistors. The CLB can be reconfigured to implement a wide range of basic Boolean logic functions, and to send/receive data from neighboring CLBs or from more distant units. The fault-tolerant PGLA concept, as described by Lach *et al* [12], assumes that a more complicated logic function is implemented using a small number ($\sim 4 - 12$) of CLBs, which are contained in an *atomic fault-tolerant block* (AFTB). Each AFTB contains one spare CLB, to allow for possible defective CLBs. Higher level functions are implemented using clusters of CLBs. The reconfigurable can be considered to be a generalisation of the fault-tolerant PGLA concept.

which, if any, of the R clusters work, and then reconfigure the supercluster so that it gives a valid output. The supercluster is then considered to be acceptable if at least one of the R clusters works. The probability that at least one cluster works out of the R in the supercluster, P_{sc-w} , is given by:

$$P_{sc-w} = 1 - (P_{cluster-f})^R \quad (21)$$

Now the number of superclusters on the chip, N_{sc} , is given by:

$$N_{sc} = \frac{N_{total}}{R \cdot N_t \cdot N_c \cdot N_A} \quad (22)$$

Hence the probability of the chip working, P_{chip-w} , is:

$$P_{chip-w} = (P_{sc-w})^{N_{sc}} \quad (23)$$

Note that it is assumed that each independent cluster can be tested and any defective clusters can be disabled (but not replaced in any way). The chip will therefore have an increased probability of working, but at the expense of a lower level of computing power.

Is there an upper bound to the maximum value of p_f that can be tolerated, for given values of N_{total} and P_{chip-w} ? The answer is yes - the upper bound is $p_f = 1.0$, and this can be approached (from below) as closely as one desires. However, this limit is useless in reality: for example, it corresponds to

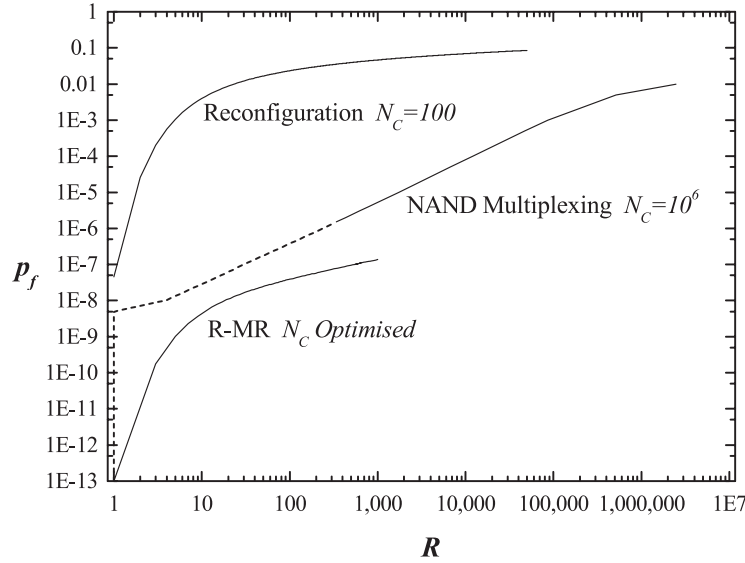


Figure 12: These graphs show the allowable failure rate per device, p_f , as a function of the amount of redundancy, R , for three different strategies, assuming that there are 10^{12} devices on a chip and that the chip must work with 90% probability. R-MR is the conventional technique of multiple redundancy, and for the curve shown, $B = 64$ and $m20$; NAND multiplexing and reconfiguration are described in the text. These curves are the approximate upper limits for each technique.

a chip with 10^{12} devices working reliably, but only having perhaps 100 working devices. We therefore consider three ‘realistic’ examples. The first provides an extremely optimistic, but semi-realistic, upper bound to the maximum acceptable value of p_f . The second and third examples provide more realistic estimates of what might be achieved in practice.

To find the ‘optimistic’ upper bound to p_f , we choose the following values:

$$N_{total} = 10^{12} \quad N_{trans} = 100 \quad N_c = 2 \quad N_A = 1$$

$$N_{clusters} = \frac{10^{12}}{(100)(2)(1)} = 5 \times 10^9$$

This describes CLBs having the smallest allowable amount of functionality. An atomic fault tolerant block has two CLBs. If one or other of these CLBs works, then the AFTB works like a single CLB. It is further assumed that there each cluster contains only one CLB, and that a higher level of fault detection is available. This ‘higher level fault detection’ model is extremely crude: it simply makes the assumption that some external counter is able to record where the dead clusters are, and only records which clusters will give a valid output. In the extreme limit, only one cluster might work out of 5×10^9 . The chip would still be considered to work, with massively reduced functionality. The allowable failure rate per transistor for this upper bound, and its dependence on the degree of redundancy, are shown in Fig. 12.

We now consider a slightly more realistic example. We choose the following values:

$$N_{total} = 10^{12} \quad N_{trans} = 500 \quad N_c = 5 \quad N_A = 50$$

$$N_{clusters} = \frac{10^{12}}{(500)(5)(50)} = 10^7$$

We now have significant numbers of useable transistors - 100 000 - in each cluster, enough to implement quite sophisticated processing elements in some hypothetical image processing architecture, neural network or pipelined digital signal processing system. Fig. 13 shows the allowable maximum value of p_f per device, as a function of the effective redundancy R , for this case.

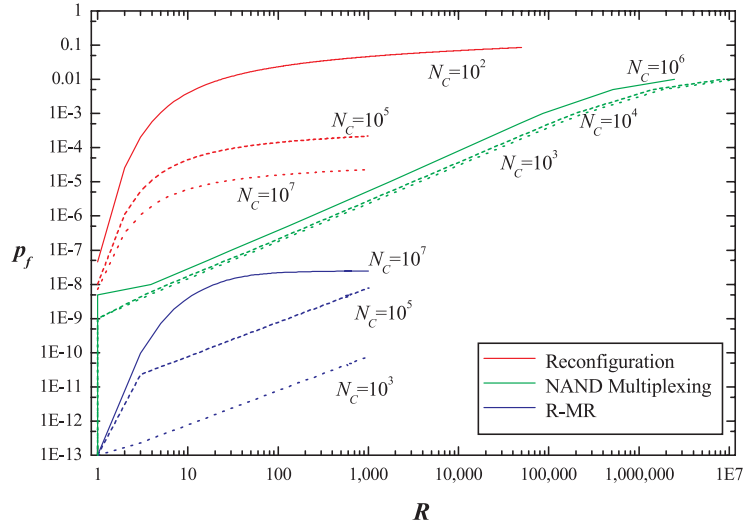


Figure 13: These graphs show the allowable failure rate per device, p_f , as a function of the amount of redundancy, R , for three different strategies, assuming that there are 10^{12} devices on a chip and that the chip must work with 90% probability. The numbers alongside each curve denote the number of effective devices in a working unit. Thus 10^2 would correspond to a small digital logic unit, whilst 10^7 is representative of a workstation chip. For R -MR, the parameters are as in Fig. 12.

Finally, we examine the p_f vs. R curve for the values:

$$N_{total} = 10^{12} \quad N_{trans} = 1000 \quad N_c = 5 \quad N_A = 10000$$

$$N_{clusters} = \frac{10^{12}}{(1000)(5)(10000)} = 2 \times 10^4$$

These numbers are intended to be representative of the numbers of devices (5×10^7) that would be needed to build a cluster with the same computing power as a present-day workstation chip. If all of the devices were manufactured perfectly then, *ceteris paribus*, the chip would contain the equivalent of 20000 workstations. Fig. 13 shows what value of p_f is acceptable for a given level of redundancy. For example, if the devices can be made with a defect rate of 0.000007% (i.e. one dead device in 140000) then the ‘molecular chip’ would have the equivalent of 33 present-day workstations on it. However, it is very unlikely that the clock speed of the ‘molecular workstations’ would be as high as that of present-day workstations.

3.5 Discussion

The theories for NAND multiplexing and reconfiguration are at present only approximate, and need refinement. However, the results show that, for overcoming manufacturing defects, the technique of reconfiguration is much the best, as expected. However, the results also show it will be necessary to produce nanodevices with defect rates in the range 10^{-5} to 10^{-4} if useful systems are to be manufactured. It should also be noted that the reconfiguration technique has an extra ‘hidden’ redundancy factor of about 10, over and above the R values shown in the graphs.

There are many other ways in which errors can occur. For example, consider transient errors, which might be due to fluctuations in electron numbers in any one clock cycle. Since there may be billions of clock cycles during the lifetime of a computer, the allowable error per clock cycle has to be extremely small. Reconfiguration is not a practical option for protecting against errors that occur unpredictably in time, and it is necessary to use multiple redundancy (R -MR or CTMR) or multiplexing.

Most existing nanodevices have only been made experimentally in one's or two's, or a few hundreds at most. It may seem inappropriate to consider faults in systems with 10^{12} devices on a chip, when even CMOS chips contain less than 10^9 devices. However, the theories that have been described in this section have to be developed now, so that the most suitable manufacturing and operational methods can be developed in good time.

4 Current status of nanoscale devices for information processing and computing

In the last ten years, several concepts for a suitable replacement for CMOS technology have appeared in the literature. The present concern with CMOS is the apparent scaling limitation of the technology. The reduction in device size also affects other important issues, such as power dissipation, operational speed, cost, reliability, etc. The revolution in computing and information processing generally has been driven by technological advancement in integrating huge numbers of electronic devices. There is thus significant interest in determining if there is any technology on the horizon beyond conventional CMOS.

The *Status Table* for emerging nanoelectronic devices (see Fig. 14) represents the situation at present, according to the available literature. We have tried to include all major devices proposed for nanoelectronics. The table should have educational as well as reference value, and might be helpful in defining new project proposals, where the authors of proposals could indicate how their research would lead to advancement on the Status Table, or the addition of extra rows in the form of new ideas and devices.

4.1 About the Status Table

Device scaling, combined with the scaling of interconnects, has allowed more components to be placed on a single chip and also led to faster operational speeds. These trends have so far produced an exponential increase in the performance and functionality of computers. However, as CMOS devices approach their scaling limits, so the search for alternative nanoscale devices is becoming more important. In addition, several new concepts and devices offer possible advantages in reduced circuit complexity and power dissipation. However, this focus of many research activities is slowly shifting into the performance of both simple and larger-scale circuits incorporating these new devices. Hence, we believe it is of considerable use to have a status table which would register the milestones in the progress of practical functional implementation of these devices. We have collected as much information as possible from the open literature and put together the status table presented in Fig. 14.

Short descriptions and references for the emerging devices in the status table are provided in the next subsection. The status table attempts to identify the development stage of certain technologies. The development status is separated into the following five groups:

- Pre-fabrication phase:

1. theory – basic theoretical concepts of devices have been put forward and appropriate analytical and numerical models of devices and simple circuits has been performed,
2. simulation – CAD software packages are available for circuit simulation and circuit layout (note: this point is in reality connected to successful device fabrication, since new or more advanced CAD packages are usually created when those circuits can be readily fabricated)

- Fabrication phase:

3. agony/struggle – experimental attempts with limited success,

	Device Name	single device	simple circuits	logic gate mem.cell	sub-system	small chip	big chip
1	CMOS						
2	Magnetic random access memory (MRAM)						
3	Organic transistors						
4	Resonant-tunnelling diode-HFET (III-V)						
5	Rapid single flux quanta (RSFQ)						
6	Single electron transistor (SET) RAM						
7	Bulk molecular logic/memory						
8	Nanotube/nanowire transistors						
9	Quantum cellular automata/magnetic (MQCA)						
10	Resonant tunnelling diodes (RTDs) (Si-Ge)						
11	Magnetic spin-valve transistors						
12	Quantum cellular automata/electronic (EQCAs)						
13	Josephson junction persistent current qubit/cubit						
14	Single electron transistor (SET) logic						
15	Molecular (hybrid electromechanical)						
16	Quantum interference/ballistic electron devices						
17	Mott transition field effect transistors (MottFET)						
18	Mono-molecular transistors and wires						







Pre-fabrication phase	 no information	 theory	 simulation
Fabrication phase	 agony/struggle	 working demonstration	 commercial or available

Figure 14: Status table for emerging nanaoelectronic devices.

4. working demonstration – experiments and/or patent descriptions which demonstrate and indeed confirm the possibility of fabricating the proposed system,
5. commercial or available – when a system can be ordered from a laboratory or company, or found on the market.

The complexity levels are classified into the following groups:

- (a) single device
- (b) simple circuit – any combination of devices,
- (c) memory cell or logic circuits (e.g. NAND, threshold gate, etc.) – relatively simple circuits but which perform desired function,
- (d) subsystem, for memory or logic function (e.g. half adder, ALU, neuron, etc.),
- (e) ‘small’ chip (memory or logic) – currently this would contain perhaps 10^6 devices,
- (f) ‘big’ chip (memory or processor) – currently this would contain $10^7 - 10^8$ devices.

The purpose of the status table is not to judge or rank the proposed devices, but to monitor their progress on the road to possible practical use. In this way one is able to see clearly what problems, at which stage, are impeding further advancement of each new device. Furthermore, every new project proposal might try to identify what improvements on the status table the proposed research activities would achieve. Alternatively, one could propose a new device, i.e. generate a new row in the status table as a project outcome. In this way the table can serve as a tool for expressing new ideas whilst clearly defining their major potential. Obviously, a single table of summary information content is not sufficient to contain all possible directions for future research. It is possible to refine the table in several ways, for example to include the scale of devices which have achieved certain milestones such as operating temperature, etc. In the case of CMOS all milestones have been achieved, and further research is concentrating on scaling down the devices and interconnects, introducing new materials, reducing power dissipation, increasing speed, enhancing throughput and generally improving the architecture for microprocessors of ever increasing complexity and increasing the number of devices for DRAM chips. One could say that these types of achievements could not be assessed with the table, but for the vast majority of proposed nanodevices the experimental demonstration of device or circuit performance at each level is still awaited, and thus the table is helpful.

At this stage, it is possible the table might include a few more devices (e.g. quantum computing devices are not included), but as an initial step we include only the major devices covered by the EC-NID projects. At a later stage it might become possible to split some of the rows (e.g. molecular devices into DNA, carbon or some other nanotubes, C_{60} , etc.) depending on the future success of some of the specific realisations of the proposed device.

4.2 Short descriptions and references for the emerging devices status table

1. *Complementary Metal Oxide Semiconductor (CMOS)*

See current Road Map.

2. *Magnetic Random Access Memory (MRAM)*

The possible bistable orientation of the magnetic state of ferromagnetic materials is a basis for non-volatile information storage (when the magnetic field is switched off, the magnetisation

orientation may be retained for a long period). Magnetic memory devices are now used for high-density, low power, high-speed, reliable (radiation hardened), non-volatile RAM. In addition MRAM offers non-destructive readout (unlike similar ferroelectric memories) and very high retention times (which eliminates the need for data refreshing) and durability. There are two main approaches in magnetoelectronic memory device, based on:

- giant magneto-resistance (GMR) by using pseudo spin-valves: Honeywell Corp. announced a 1Mbit GMRAM chip for 2001, and
- magnetic-tunnel junctions: Motorola demonstrated and successfully processed a 256Kbit chip in 2001 (access time 50ns, power 24mW at 3V); IBM is another significant player.

Both systems use MOSFET in the memory cell. Currently there is a manufacturing problem because fabrication processes for multilayer magnetic elements and conventional CMOS are incompatible.

2.1 M. Johnson: "Magnetoelectronic memories last and last", *IEEE Spectrum* pp. 33-40, Feb. 2000.

2.2 B. Heinrich: "Magnetic nanostructures. From physical principles to spintronics", *Can. J. Phys.* **78**, pp. 161-199 (2000).

2.3 <http://e-www.motorola.com>

3. *Organic transistors*

Integrated circuits based on organic transistors, with gate lengths of ~ 10 micrometres, are at present nowhere near the nanoscale, but there is no fundamental reason why they cannot be made at or near the nanoscale. The largest reported circuit has 864 devices, and smaller circuits are commercially available. Clock frequencies are at present in the range 1-20kHz.

3.1 B. Crone, A. Dodabalapur, Y.-Y.Lin, R.W. Filas, Z. Bao, A. LaDuca, R. Sarpeshkar, H.E. Katz and W. Li: "Large-scale complementary integrated circuits based on organic transistors", *Nature* **403**, pp. 521-523 (2000).

4. *Resonant Tunnelling Diode-Heterostructure Field Effect Transistor RTD-HFET (III-V)*

Resonant Tunnelling Diodes (RTDs) are the most developed 'quantum-mechanical' devices (two very thin potential barriers separated by a very small distance, $\sim 3 - 10$ nm, which enables big energy level separations $\Delta E \gg kT$ even for room temperatures). The very small size of this structure along the electron transport direction makes it very fast (switching time constants of 1.5ps) and in addition, they can operate at relatively low voltages (0.5-1V). RTDs show negative differential resistance in the $I - V$ characteristics, and this feature has been successfully exploited for many applications, for example in very high frequency oscillators and amplifiers as well as in digital systems with Boolean or threshold logic, low-power memory cells, multi-valued and self-latching logics, and even in neural networks. III-V RTDs are used in conjunction with a high-mobility field effect transistor, which provides the necessary gain. Small (4×4 -bit) tunnelling-based SRAM memory arrays have been fabricated and an 1kbit chip has been designed (Raytheon Systems) [4.1,4.2], and parallel adder units, containing 20 RTD-HFET transistors [4.3], have been built, tested, and reported in the open literature. RTD memories are potentially interesting as high-speed, dense, low-power, on-chip memories, for fast compound semiconductor microprocessors. The performance of large-scale mixed logic/memory circuitry using RTD-HFET devices has also been investigated using HSPICE simulation [4.4].

4.1 J.P.A. Van der Wagt, A.C. Seabaugh and E.A. Beam: "RTD/HFET Low Standby Power SRAM Gain Cell", *IEEE Electron Device Lett.* EDL-19, pp. 7-9 (1998).

- 4.2 J.P.A. van der Wagt: “Tunneling-based SRAM”, *Nanotechnology* **10**, pp. 174-186 (1999).
- 4.3 C. Pacha, U. Auer, C. Burwick, P. Glösekötter, K. Goser, W. Prost, A. Brennemann and F.-J. Tegude: “Threshold Logic Circuit Design of Parallel Adders Using Resonant Tunneling Devices”, *IEEE Tran. on VLSI Systems* **8**, pp. 558-572, (2000).
- 4.4 K. Nikolic and M. Forshaw: “The operational speed and power dissipation of scaled one-transistor RTD/HFET memories”, *Int. J. Electronics* **88**, pp. 453-462 (2001).

5. *Rapid Single Flux Quantum (RSFQ)*

RSFQs rely on the use of Josephson junctions in superconducting circuits and a magnetic flux quantum for encoding information bits. Although these are not nanoscale devices, it is predicted that they can be made down to $\sim 300\text{nm}$ in size [5.1]. Small circuits running at $\sim 20\text{GHz}$ are already commercially available and bit rates of 750Gbit/s with flip-flops have been achieved [5.2]. Circuits with $\sim 10,000$ devices have been demonstrated. Random access memories, adders and multipliers have been demonstrated [5.3]. The current generation of devices is mainly based on the low-temperature (4-10K) superconductors, but high-temperature ($\sim 50\text{K}$) superconductor technology is also developing. The main advantages of RSFQ technology are: very high speed, reduced thermal noise (in comparison with room-temperature circuits) and low power (some RSFQ circuits dissipate over 10^5 times less power than the semiconductor version). This low dissipation is quite important, since the very dense packing of circuits required for, e.g. a petaflop computer, makes it very difficult to cool high dissipation circuits. However a cooling system is needed and another problem with these devices is their lack of gain. Currently there are also technological problems in achieving device parameter consistency for operating standards when large number of devices are integrated.

- 5.1 D.K. Brock; E.K. Track; J.M. Rowell, “Superconductor ICs: the 100-GHz second generation”, *IEEE-Spectrum* **37(12)**, pp. 40-46 (2000).
- 5.2 W. Chen, A.V. Rylyakov, V. Patel, J. E. Lukens and K.K. Likharev: “Rapid Single Flux Quantum T-flip flop Operating up to 770 GHz”, *IEEE Trans. Appl. Supercond.* **9 (2)**, pp. 3212 -3215 (1999).
- 5.3 P. Bunyuk, K. Likharev, D. Zinoviev and D. Brock, “RSFQ Technology: Physics, Devices, Circuits and Systems”, *Int. J. High Speed Electronics and Systems*, Jan. (2001)

6. *Single Electron Transistor Random Access Memory (SET RAM)*

Single Electron Tunnelling (SET) devices consist of one or more small islands, separated from each other and from external electrodes by tunnel junctions. If the island capacitance is very small ($\sim 1 - 100\text{aF}$), the electrostatic potential of the island may change considerably when an electron tunnels onto/off the island, affecting further tunnelling to/from the island. Possible applications are in metrology (current standard), instrumentation (e.g. high-sensitivity electrometers for the detection of fractions of e) and for computing. A 128Mbit memory has been designed and fabricated (Hitachi, 1998), based on Yano memory cells (ultrathin-film transistors with poly-Si channel) in combination with CMOS peripheral circuits [6.1]. The cell size was $0.15\mu\text{m}^2$, read time $1.2\mu\text{s}$, room-temperature operation, but less than a half of the cells were operational. Another memory type was proposed by a Cambridge group [6.2], which has fabricated a $3 \times 3\text{bit}$ memory array of integrated SET/MOSFET cells. The memory node is the gate of a split-gate MOSFET which is connected to the word line by a very narrow silicon wire, which behaves as a multiple-tunnel junction SET. A much-discussed SET memory was the NOVORAM (non-volatile RAM [6.3]), which relied on the storage of electrons in a floating gate (‘nano-flash’ device). So far this exists only as a design study. These memory devices, although they are called ‘single electron’ devices, and are based on the Coulomb blockade effect, in fact rely on sensing more than one electron for reliable operation.

- 6.1 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”, *Proc. IEEE* **87**, pp. 633-650, (1999).
- 6.2 Z.A.K. Durrani, A.C. Irvine and H. Ahmed: “Coulomb Blockade Memory Using Integrated Single-Electron Transistor/Metal-Oxide-Semiconductor Transistor Gain Cell” *IEEE Trans. Electron Devices* **47**, pp. 2334-39 (2000).
- 6.3 K.K. Likharev: “Single-Electron Devices and Their Application”, *Proc. of the IEEE* **87**, pp. 606-632 (1999).

7. *Molecular bulk devices*

Molecular layers, sandwiched between metallic or polysilicon electrodes, have been demonstrated to act as switches or memory elements. By configuring several devices together, AND and OR logic gates have been demonstrated. The devices use tunnelling and do not exhibit gain.

- 7.1 C.P. Collier, E.W. Wong, M. Belohradsky, F.M. Raymo, J.F. Stoddart, P.J. Kuekes, R.S. Williams and J.R. Heath: “Electronically configurable molecular-based logic gates”, *Science* **285**, pp. 391-394 (1999).
- 7.2 C.P. Collier, G. Mattersteig, E.W. Wong, Yi-Luo, K. Berverly, J. Sampaio, F.M. Raymo and J.F. Stoddart: “A [2]catenane-based solid state electronically reconfigurable switch”, *Science* **289**, pp. 1172-1175 (2000).

8. *Nanotubes/nanowires*

Carbon nanotube transistors have been demonstrated, but so far they exhibit very low gain [8.1,8.2]. The first array of field-effect transistors made of single-walled nanotubes has been recently constructed [8.3]. Using the same procedure it was demonstrated that multiwalled nanotubes can be peeled controllably shell-by-shell. The band gap of carbon nanotubes depends on the tube diameter, therefore it seems feasible to prepare nanotubes with custom band gaps, which should provide opportunities for wider electronic applications. It has been proposed recently that the hierarchical assembly of one-dimensional nanostructures into well-defined functional networks may lead to novel design technique for nanoscale electronic and photonic structures [8.4]. The semiconductor nanowires used in these studies include GaP, InP and Si nanowires, which could function as building blocks for nanoscale electronics [8.5]. The simple circuit assembly avoids the need for complex and costly fabrication facilities, and combines fluidic alignment and surface-patterning techniques. With p and n doped silicon nanowires used as building blocks, and by making structures consisting of crossed p- and n-type nanowires, three types of semiconductor nanodevices were assembled: diodes, bipolar transistors and complementary inverter-like structures.

- 8.1 M. Kruger, M. R. Buitelaar, T. Nussbaumer and C. Schonenberger: “Electrochemical carbon nanotube field-effect transistor”, *Appl. Phys. Lett.* **78**, pp. 1291-93 (2001).
- 8.2 M. Ahlskog, R. Tarkiainen, L. Roschier and P. Hakonen: “Single-electron transistor made of two crossing multiwalled carbon nanotubes and its noise properties”, *Appl. Phys. Lett.* **77**, pp. 4037 (2000).
- 8.3 Philip G. Collins, Michael S. Arnold, and Phaedon Avouris: “Engineering Carbon Nanotubes and Nanotube Circuits Using Electrical Breakdown”, *Science* **292**, pp.706-709 (2001).
- 8.4 Y. Huang, X. Duan, Q. Wei and C.M. Lieber: “Direct assembly of One-Dimensional Nanostructures into Functional Networks”, *Science* **291**: (5504) pp. 633 (2001).
- 8.5 Y. Cui and C.M. Lieber: “Functional nanoscale electronic devices assembled using silicon nanowire building blocks”, *Science* **291**: (5505), pp. 851-853 (2001).

9. *Magnetic Quantum Cellular Automata (MQCAs)*

Quantum Cellular Automata devices offer an alternative computing architecture to CMOS technology. QCA circuits consist of arrays of cells. Each cell affects its neighbouring cells through a (electric or magnetic) field, and it does not normally have any other connections. Magnetic QCAs rely on a property of very small ferromagnetic structures (nanomagnets), namely that the electronic spins act coherently as a 'giant spin'. The direction of cell magnetization can represent the logical state (it is possible to achieve that only two directions are energetically favourable). The cells interact through magnetostatic interaction. A 'wire' of 68 magnetic QCA 'dots', each $\sim 100\text{nm}$ in diameter, has been shown to propagate a 1/0 signal at room temperature.

9.1 R.P. Cowburn and M.E. Welland, "Room temperature magnetic quantum cellular automata", *Science* **287**, pp. 1466-1468 (2000).

10. *Resonant Tunnelling Diodes (RTD) (Si)*

So far RTDs have been realized in many III-V compound systems, but silicon based RTDs are even more desirable, so that integration with silicon based devices is possible. Intensive research is currently being undertaken and significant progress has been reported on Si/SiGe heterostructure RTDs [10.1], but there are still problems with stability and/or low current peak-to-valley ratios. Silicon structures with Si-oxide barriers should be a natural choice for silicon RTDs, but methods for the growth of crystalline Si overlayers on top of oxide barriers are still being developed [10.2].

10.1 D.J. Paul, P. See, I.V. Zozoulenko, K.-F. Berggren, B. Kabius, B. Hollander and S. Mantl: "Si/SiGe electron resonant tunneling diodes", *Appl. Phys. Lett.* **77**, pp. 1653-55 (2000).

10.2 Y. Wei R.M. Wallace RM, A.C. Seabaugh: "Controlled growth of SiO₂ tunnel barrier and crystalline Si quantum wells for Si resonant tunneling diodes", *J. Appl. Phys.* **81**, pp. 6415-24 (1997).

11. *Magnetic spin-valve transistors*

Spin-valve transistors are magnetoelectronic devices which can operate at room temperatures. The current flow through two (thin) layers of magnetic material, separated by a layer of a conductor (Cu, Au), may be strongly dependent on the magnetic moments of the magnetic layers and therefore the current can be modulated by applying an external magnetic field. Although the use of magnetoelectronic spin valve technology in memory devices is well-established, the development of spin-valve transistors, first described in 1995, is less advanced. However, on-off ratios of 4:1 have recently been described in room temperature spin-valve transistors, and it is possible that these devices may be useful in logic circuits. Spin-valve devices find application as magnetic sensors or in read heads of disk drives, i.e. as giant magnetoresistance field sensors.

11.1 P.S. Anil Kumar, R. Jansen, O.M.J. Van't-Erve, R. Vlutters, S.D. Kim and J.C. Lodder: "300% magnetocurrent in a room temperature operating spin-valve transistor", *Physica-C* **350 (3-4)**, pp. 166-170, 2001.

12. *Electronic Quantum Cellular Automata (EQCAs)*

One of the most-analysed EQCA cell designs consists of four quantum-dots [12.1]. Cells are electro-neutral (but have two free electrons), and can be in a polarized or non-polarized state, depending on the tunnelling potential between quantum wells. There are two polarized states (i.e. cell charge configurations) which can be used for encoding binary information. The mutual cell interactions can be used to propagate and compute binary information. Every QCA system is supposed to evolve into a ground state determined by the polarisation of edge cells (and possible external fields). There have been only a few experimental realizations of EQCA cells so far

[12.2], but theoretical work has produced a number of functional QCA circuit designs, for example: an adiabatically clocked ripple carry adder [12.2], 1-bit addressable SRAM cell [12.3], etc. A low-level QCA simulation program (with a GUI) named AQUINAS [13.4] was developed, where smaller assemblies of QCAs were simulated at quantum mechanical level. A high level simulation tool, SQUARES [13.4], allows simulations of large assemblies of QCAs but at the logical level. One major problem with EQCA devices is how to control the unpredictable effects of charge hopping in the substrate ('buried charge').

- 12.1 C.S. Lent and P.D. Tougaw: "Dynamics of quantum cellular automata", *J. Appl. Phys.* **80**, pp. 4722-36 (1996).
- 12.2 W. Porod, C.S. Lent, G.H. Bernstein, A.O. Orlov, I.H. Amlani, G.L. Snider and J.L. Merz: "Quantum-dot cellular automata: computing with coupled quantum dots", *Intl. Journal of Electronics* **86**, pp. 549 (1999).
- 12.3 K. Nikolic, D. Berzon and M. Forshaw: "Relative performance of three nanoscale devices - CMOS, RTDs and QCAs - against a standard computing task", *Nanotechnology* **12**, pp. 38-43 (2001).
- 12.4 AQUINAS - A QUantum Interconnected Network Array Simulator 1996 - 1997, (University of Notre Dame), <http://www.nd.edu/~qcahome>. SQUARES - Standard QUantum cellular automata Array Elements (University College London, 1998), <http://ipga.phys.ucl.ac.uk/reports/rep98-1.pdf>.

13. *Josephson Junction Persistent Current Bits - Qubits and classical bits ('cubits') JJPCB*

A superconducting loop with three Josephson junctions around its perimeter can support two opposite-circulating currents, whose relative magnitude can be controlled by external magnetic fields. A working device has been built. It was originally proposed (and demonstrated) as being capable of storing a quantum bit (qubit), but it has also been proposed for use as a 'classical' binary logic device ('classical qubit', or 'cubit'). Nanoscale superconducting quantum bits based on Josephson junctions are the possible building blocks for future quantum computers, because they combine the coherence of the superconducting state with the control possibilities of single-charge circuits.

- 13.1 P. Jonker & J. Han: "On quantum and classical computing with arrays of superconducting persistent current qubits", Proc. CAMP2000, Fifth IEEE international Workshop on Computer Architectures for Machine Perception (Pavona, Italy, Sep.11-13, 2000) *IEEE Computer Society*, Los Alamitos, California, UASDA, pp. 69-78 (2000).
- 13.2 T.Orlando, J.E. Mooij, L.Tian, C.H. van der Wal, L. Levitov, S. Lloyd, J.J. Mazo: "A superconducting persistent-current qubit", *Phys. Rev. B* **60**, 22 (1999)
- 13.3 J.E.Mooij, T.P. Orlando, L.Levitov, L.Tian, C.H. van der Wal, S.Lloyd: "Josephson persistent-current qubit", *Science* **285**, pp. 1036, (1999).
- 13.4 C.H. van der Wal, A.C.J. ter Haar, F.K. Wilhelm, R.N. Schouten, C.J.P.M. Harmans, T.P. Orlando, S. Lloyd, J.E. Mooij: "Quantum superposition of persistent-current states", *Science* **290**, pp. 773-777 (2000).

14. *Single Electron Transistor (SET) logic*

One of the basic blocks of conventional logic circuits is the inverter. An SET inverter circuit has been demonstrated [14.1], having GaAs SET transistor and a variable load resistance (operation temperature 1.9K). An SET inverter with a voltage gain has been experimentally demonstrated [14.2], based on metallic, Al/Al₂O₃/Al, tunnel junctions, but only at temperatures below 0.14K. By making slight alternations to the circuit, NAND and NOR logic gates might be produced. An attempt to experimentally realize a NAND logic gate, by using SET transistors based on the silicon nanowires (at temperature 1.6K), has been reported [14.3], but the results point to

severe problems (low ON/OFF voltage ratio, low output, need for individual adjustment of each transistor, etc). One major problem with SET logic devices is how to control the unpredictable effects of charge hopping in the substrate ('buried charge'). Theoretical investigations of SET circuits have been very extensive and circuit level simulation packages have been developed (SIMON, MOSES, SENECA [14.4]). More efficient simulations than Monte-Carlo calculations would be possible if a SPICE-type circuit model of SET devices could be introduced. However, macromodelling of SETs is in general problematic, because the tunnelling events are random and the evolution of the system depends on the free energy of the whole system. A number of SET digital circuit designs have been suggested (e.g. ref. [14.5]) based on conventional circuit design methods.

- 14.1 F. Nakajima, K. Kumakura, J. Motohisa and T. Fukui: "GaAs single electron transistors fabricated by selective area metalorganic vapor phase epitaxy and their application to single electron logic circuits", *Jpn. J. Appl. Phys.* **38**, pp. 415-417 (1999).
- 14.2 C.P. Heij, P. Hadley and J.E. Mooij: "Single-electron inverter", *Appl. Phys. Lett.* **78**, pp. 1140-42 (2001).
- 14.3 N.J. Stone and H. Ahmed: "Logic circuit elements using single-electron tunneling transistors", *Electronics Letters* **35**, pp. 1883-84 (1999).
- 14.4 SIMON – SIMulations of Nanostructures (TU Wien, 1997), a Monte-Carlo type SET device and circuit simulator with a graphical circuit editor embedded in a graphical user interface. C. Wasshuber, H. Kosina, and S. Selberherr: "SIMON - A simulator for single-electron tunnel devices and circuits", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **16** (9), pp. 937-944 (1997).
- 14.5 M.G. Ancona: "Design of computationally useful single-electron digital circuits", *J. Appl. Phys.* **79**, pp. 526-539 (1996).

15. *Molecular - electromechanical devices*

This type of device uses the effect of mechanical deformation on the electrical properties of molecules. Transistor-like structures have been fabricated, that use a molecule (C_{60} [15.1], carbon nanotube [15.2]) which is compressed by an STM tip, and gain has been demonstrated. Hybrid electromechanical transistors have been designed, and the properties of small-scale and large-scale memory-logic circuits using these devices have been analysed in theoretical simulations [15.3]. In principle this type of device can operate at room temperature. The main challenges with this technology are the accurate positioning of molecules in nano-junctions and the accurate control of the position and dimensions of the deformation tool.

- 15.1 C. Joachim, J.K. Gimzewski and A. Aviram, "Electronics using Hybrid-Molecular and Monomolecular Devices", *Nature* **208**, pp. 541-545, 2000.
- 15.2 T.W. Tombler, C. Zhou, L. Alexseyev, J. Kong, H. Dai, L. Liu, C.S. Jayanthi, M. Tang & S.-Y. Wu: "Reversible electromechanical characteristics of carbon nanotubes under local-probe manipulation", *Nature* **405**, pp. 769-772 (2000).
- 15.3 S. Ami and C. Joachim, "Logic gates and memory cells based on single C_{60} electromechanical transistors", *Nanotechnology* **12**, in press, 2001.

16. *Quantum interference/ballistic electron devices*

When inelastic electron scattering is sufficiently reduced, coherent transport and interference effects begin to appear. Switches or transistors have been designed and built, that rely on quantum interference phenomena. Theoretical predictions of transistor-type action in a T-shaped electron waveguide in the ballistic regime [16.1], have been experimentally confirmed [16.2]. Theoretical descriptions of simple circuits of quantum interference transistors based on stubbed waveguides

also exist [16.3]. The main problem with the interference-type devices is that the interference effects are very sensitive to fluctuations of the system geometry, therefore fabrication tolerances are very limited. Furthermore, there are certain device size and temperature limits for the required coherent transport. There is also the contact resistance problem, since the resistance at the junctions of a narrow, mono-mode electron waveguide is $h/(2e^2) \approx 13k\Omega$, which limits the high-frequency response. A 3-terminal (Y-branch) switching device has been proposed and experimentally demonstrated [16.4] (strictly speaking this is not an interference type device, but it has a similar structure). Gates are located on the both sides near the branching point, and by applying voltage on the gates the incoming electron is probabilistically deflected between two output branches. At present these devices do not exhibit gain.

- 16.1** F. Sols, M. Macucci, U. Ravaioli and K. Hess: "On the possibility of transistor action based on quantum interference phenomena", *Appl. Phys. Lett.* **54(4)**, pp. 350-352 (1989).
- 16.2** P. Debray, O.E. Raichev, P. Vasilopoulos, M. Rahman, R. Perrin, W.C. Mitchell: "Ballistic electron transport in stubbed quantum waveguides: Experiment and theory", *Phys. Rev.* **61 (16)**, pp. 10950-958 (2000).
- 16.3** K. Nikolic, P. Nikolic and R. Sordan, "Conductance of quantum interference transistors in parallel and in series", *Superlattices and Microstructures* **26**, pp. 47-55 (1999).
- 16.4** J-O. Wesstrom, "Self-gating effect in the electron Y-branch switch", *Phys. Rev. Lett.* **82(12)**, pp. 2564-2567 (1999).

17. *Mott-transition Field Effect Transistors (Mott-FET)*

Transistors using the Mott-Hubbard metal-insulator transition for switching between 'ON' (low impedance) and 'OFF' (high impedance) state, have been proposed by IBM researchers [17.1] and prototype devices have been built [17.2,17.3]. One potential advantage over Si-based FET devices is that the gate length of these devices could perhaps be much smaller than 30-35nm. Room temperature operation has been achieved. The channel material must be capable of undergoing a Mott metal-insulator transition induced by the gate field. The cuprate family of perovskite structure materials, which are related to high-temperature superconductors, has this desired ability.

- 17.1** C. Zhou, D.M. Newns, J.A. Misewich, P.C. Pattnaik: "A field effect transistor based on the Mott transition in a molecular layer", *Appl. Phys. Lett.* **70(5)**, pp. 598-600 (1997).
- 17.2** D.M. Newns, J.A. Misewich, C.C.Tsui, A. Gupta, B.A. Scott and A. Schrott: "Mott transition field effect transistor", *Appl. Phys. Lett.* **73(6)**, pp. 780-782 (1998).
- 17.3** J.A. Misewich and A.G. Schrott: "Room-temperature oxide field-effect transistor with buried channel", *Appl. Phys. Lett.* **76(24)**, pp. 3632-3634 (2000).

18. *Molecular nanoelectronic devices*

The basic idea is to have a molecule or a group of connected molecules which can perform as nanoelectronic elements. For example, the conductance of a molecule could be controlled by an external electric field and thus a molecular switch (molecular transistor) is created, or molecules could be rectifiers, resonant tunnelling diodes, logic gates, etc [18.1]. Molecules, in general, have the advantage of being very small, identical, and could be produced in very large quantities. If molecular-scale transistors could be fabricated then the possibility of having chips with $\sim 10^{12}$ transistors may become a reality. There is still a long way to go before such circuits can become a reality. There are difficult technological problems in manipulating single molecules, in providing reliable contacts to molecules and in assembling individual molecular devices into extended structures and circuits. However, experimental work so far has demonstrated that a

single molecule can conduct current and therefore molecular wires are possible (see e.g. [18.2]). A large amount of theoretical work has been devoted to identifying molecules which might be used as diodes, RTDs, transistors, logic gates, etc. [18.3]. Due to current difficulties in performing experiments on the proposed molecular electronic devices, molecular modelling is very important (e.g. [18.4]).

- 18.1** C. Joachim, J.K. Gimzewski and A. Aviram: "Electronics using hybrid-molecular and mono-molecular devices", *Nature* **408**, pp. 541-548 (2000).
- 18.2** L.A. Bumm, J.J. Arnold, M.T. Cygan, T.D. Dunbar, T.P. Burgin, L. Jones II, D.L. Allara, J.M. Tour, and P.S. Weiss: "Are Single Molecular Wires Conducting?", *Science* **271**, pp. 1705-1707 (1996).
- 18.3** J.C. Ellenbogen and J.C. Love: "Architectures for molecular electronic computers. 1. Logic structures and an adder designed from molecular electronic diodes", *Proc. of the IEEE* **88(3)**, pp. 386-426 (2000).
- 18.4** E. Emberley and G. Kirczenow: "Principles for the design and operation of a molecular wire transistor", *J. Appl. Phys.* **88**, pp. 5280-5282 (2000).

References

- [1] Semiconductor Industry Association: "The National Technology Roadmap for Semiconductors", 1999.
- [2] R. Campaño, L. Molenkamp, D.J. Paul: "Technology Roadmap for European Nanoelectronics", 2000.
- [3] K. Nikolic and M. Forshaw: "The operational speed and power dissipation of scaled one-transistor RTD/HFET memories", *Int. J. Electronics* **88**, pp. 453-462 (2001).
- [4] C. Pacha, U. Auer, C. Burwick, P. Glöseköter, K. Goser, W. Prost, A. Brennemann and F-J. Tegude: "Threshold Logic Circuit Design of Parallel Adders Using Resonant Tunneling Devices", *IEEE Tran. on VLSI Systems* **8**, pp. 558-572 (2000).
- [5] K. Yano, T. Ishii, T. Sano, F. Murai and K. Seki: "Single-Electron-Memory Integrated Circuit for Giga-to-Tera Bit Storage", *Proc. of IEEE Intl. Solid-State Circuits Conf.* , pp. 266-267 (1996).
- [6] C.S. Lent and P.D. Tougaw: "A device architecture for computing with quantum dots", *Proc. IEEE* **85**, pp. 541-557 (1997).
- [7] C.S. Lent and P.D. Tougaw: "Dynamics of quantum cellular automata", *J. Appl. Phys.* , **80**, pp. 4722-36 (1996).
- [8] Z.A.K. Durrani, A.C. Irvine and H. Ahmed: "Coulomb Blockade Memory Using Integrated Single-Electron Transistor/Metal-Oxide-Semiconductor Transistor Gain Cell", *IEEE Trans. Electron Devices* **47**, pp.2334-39 (2000).
- [9] Star-HSPICE CMOS circ. simulator, ©1998 Avant! Corp, <http://www.avanticorp.com/>
- [10] J. von Neumann: "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components", *Automata Studies*, C.E. Shannon and J. McCarthy, eds., Princeton University Press, Princeton N.J. 1955, pp. 43-98.
- [11] J.R. Heath, P.J. Kuekes, G.S. Snider and R.S. Williams: "A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology", *Science* **280**, pp. 1716-1721 (1998).
- [12] J. Lach J, W.H. Mangione-Smith and M. Potkonjak: "Low Overhead Fault-Tolerant FPGA Systems", *IEEE Trans. VLSI* **6**, pp. 212-221 (1998).
- [13] Xilinx Advanced Product Specification for Virtex-II FPGAs: April 2001 www.xilinx.com