Testing and Defect Tolerance: A Rent's Rule Based Analysis and Implications on Nanoelectronics

Arvind Kumar and Sandip Tiwari
School of Electrical and Computer Engineering,
Cornell University, Ithaca, NY 14853
{ak226, st222}@cornell.edu

Abstract

Defect tolerant architectures will be essential for building ***economical*** *gigascale nanoelectronic computing systems to permit functionality in the presence of significant number of defects. The central idea underlying a defect tolerant configurable system is to build the system out of partially perfect components, detect the defects and configure the available good resources using software. In this paper we discuss implications of defect tolerance on power, area, delay and other relevant parameters for computing architectures. We present a Rent's rule based abstraction of testing for VLSI systems and evaluate the redundancy requirements for observability. It is shown that for a very high interconnect defect density, a prohibitively large number of redundant components are necessary for observability and this has adverse affect on the system performance. Through a unified framework based on a priori wire length estimation and Rent's rule we illustrate the hidden cost of supporting such an architecture.*

1: Introduction

Traditionally, MOS transistor scaling simply meant decreasing linear dimensions augmented by further modifications needed for better electrostatic control (supply voltage, junction depths, channel doping, gate dielectric thickness etc). As the dimensions shrunk, physical problems such as lithography, power supply, threshold voltage, short channel effect, gate dielectric leakage, high-field effects, and random dopant fluctuations have become more important. To make things worse, devices are now more sensitive to external environments such as radiation effects, temperature variations, and electromagnetic interference. The devices exhibit increased parameter fluctuations. Scaling of CMOS is further complicated by power density, reliability, electromigration, high interconnect resistance and crosstalk issues. The increasing miniaturization is causing the chip failure rate to increase as both the number of devices and the individual devices failure rate increases. Hence, defect/fault tolerant architectures are increasingly of interest to produce reliable system that are immune to manufacturing defects and to transient runtime errors.

Testing and locating defects are inherent part of defect tolerance. Testing complexity increases as the device density increases due to access restrictions. Also, in the worst case, computational time for test pattern generation increases exponentially with number of logic blocks (LBs). Therefore, it is important to evaluate the cost of testing in presence of significant defect rates and impact on system performance.

2: Reconfigurability and TERAMAC

Traditionally reconfigurable architectures have been studied extensively for extracting multiple functionalities out of the same silicon core. This flexibility also provides a mean for post-fabrication fault

Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)
1550-5774/04 $ 20.00 IEEE

IEEE COMPUTER SOCIETY

tolerance by detecting, locating and avoiding the defects. The reconfigurable architecture concept is greatly assisted by the use of FPGAs (Field Programmable Gate Arrays).

One example of defect tolerant configurable system is the Teramac (Tera Multiple Computer Architecture) [7]. Teramac consists of 65,536 LUTs (Look Up Tables) distributed among 864 FPGAs and connected via crossbars in a fat-tree network. This extremely exible architecture had few critical paths (power and clock wires) and highly redundant network connectivity through the fat-tree; this allowed the compiler to map out the defects in the system, and simply route around bad LUTs and interconnects when compiling.

In particular, the conscious design decision to out-wire Rent's Rule (Teramac used Rent's Rule exponents of 2/3 to 1, as opposed to 1/2 to 2/3) allowed the system to function normally despite defects in 10% of logic cells and 10% of interchip interconnects. Despite running at a clock frequency of only 1 MHz, the machine outperformed many powerful workstations in certain applications (Graph Partitioning, Long Integer Multiply, DNA String Matchers etc).

In this paper we further explore feasibility of reconfigurable architecture. A reconfigurable implementation requires (a) array of simple and identical computational cores capable of heterogeneous computation, (b) localized communication and interconnects and (c) **dynamic reconfigurability with low overhead with scalable defect/fault detection scheme**. Testing is essential to locate the defective components before reconfiguring circuits around the defects. But testing process itself is affected (observability and controllability) due to presence of defects. Hence there is an implementation cost due to redundancy in terms of area, delay etc. In section 3 we will briefly discuss Rent's rule and wire length estimation based on it. Section 4 then introduces our Rent's rule based abstraction of testing and its consequence on observability. We then further explore testing in presence of defects, the redundancy requirements and impact on system parameters.

3: Rent's Rule

Rent's rule is a power law first used by Landman and Russo [6] for estimating the average number of terminals T required to connect a subregion of a circuit layout with the remainder of the circuit as a function of the number of gates N in that region. The empirical formula for this estimation,

$$T = tN^p \tag{1}$$

is determined by the Rent parameters: the Rent exponent p $(0 < p < 1)$ and the Rent coefficient t. Rent's rule holds for all hierarchal levels when a circuit is partitioned. The number of interconnections among a group of sub-components at any level is proportional to the total terminal count of all the subcomponents. Smaller values of the Rent exponent, p, represent placement optimization within a statistically homogeneous circuit which favors short over long range communication. $p = 1$ implies there is no placement optimization, and the circuit is interpreted as a random gate arrangement. Rent exponent is characteristic of a given architecture with microprocessors, gates arrays, and high-speed computers characterized by Rent exponents of 0.45, 0.5, and 0.63, respectively [1].

Wire length estimation: Applications of Rent's rule include: layout parameter estimations in electronic design automation, studies of new computer architectures, and the generation of synthetic circuit benchmarks. Apart from the last application which uses Rent's rule directly, all applications use Rent's rule to obtain wire length estimates.

Donath's method is the basis for nearly every a priori wire length estimation technique. The Manhattan grid serves as a model for the physical architecture in which the circuit placement process minimizes the total wire length (i.e., the sum of all distances between connected gates). Since it is assumed that wires are always routed along the shortest path, the wire length follows from the placement information

IEEE COMPUTER SOCIETY

alone. Thus, the method estimates the number of terminals from Rent's rule, predict the number of nets from the number of terminals, and estimates the average length of each of those nets.

We choose wire length as the prime reference parameter for performance evaluation since it has a direct impact on other design parameters such as delay, area, power dissipation, routability and congestion. An analysis of dependence of these parameters on wire length is given in [9] and is summarized in appendix for completeness.

4: Testing

Testing complexity increases as the device density increases. Any test procedure now needs to access a larger number of devices and interconnects through a proportionality small number of output terminals, and it becomes more complex with higher level of integration. The modules away from the periphery are difficult to access directly from outside and may require several cycles to be activated and read. We are thus required to observe these modules through other accessible modules. This makes testing of internal nodes more difficult as they can no longer be easily controlled by signals form outside (controllability) and not easily observed from outside (observability).

This abstraction of testing procedure can be understood using Rent's rule. If there is no placement optimization, ($p = 1$), then output terminals of all the LBs will be accessible from outside for testing. But in real cases, only a fraction of LBs are directly accessible ($N^I_{accessible}$) through the available external terminals. We can estimate this using Rent's rule and assuming that it requires t terminals per LB to fully diagnose it.

$$tN^I_{accessible} \sim T(= tN^p) \quad or \quad N^I_{accessible} \sim N^p \tag{2}$$

In the next phase, the external terminals and tested LBs (along with associated interconnects) are used to access remaining untested LBs. This can be estimated again using Rent's rule. Number of LBs remaining to be tested after the first phase are $N - N^I_{accessible} = N - N^p$. And the number of LBs accessible ($N^{II}_{accessible}$) for testing in the second phase again depends on the available terminals from the $N - N^p$ LBs.

$$tN^{II}_{accessible} \sim T(= t(N - N^p)^p) \quad or \quad N^{II}_{accessible} \sim (N - N^p)^p \tag{3}$$

The method is iterated till we can access and test the desired fraction of LBs. This procedure is illustrated in Figure 1 for a hypothetical mesh architecture where LBs are connected to their nearest neighbors. In the first phase, only the peripheral LBs are accessible and are tested (shown in white). In the second phase, these tested LBs are used to access further connected LBs. Each iteration represents complexity in terms of time and test analysis.

This abstraction assumes that any tested LB is electrically and logically transparent for subsequent testing phases. In designs comprising of heterogenous elements, this is an over simplification. It neglects the complexity to generate signals and test vectors to access internal LBs indirectly. Only in regular array architectures where elements are interchangeable, e.g. FPGAs, the non-defective LBs can be configured appropriately to pass on signals.

Figure 2 shows the percentage of components left for testing after each phase for different Rent exponent and number of LBs. Plots are generated using the algorithm describe above. It is imperative to observe the exponential dependence of these parameters on testing complexity.

5: Testing in presence of defects

In the proposed abstraction the effect of defects is taken into account by their impact on accessible LBs during a testing phase. Intuitively, more defects will make it harder to test all the modules due to

IEEE COMPUTER SOCIETY

loss of signals in the defective components. Thus, there is a redundancy requirement for observability which can be much higher (at higher defect densities) than the functional redundancy required to compensate for the defective elements. For ease of analysis, we assume uniform and independent defect density among the components. In this work we have assumed defect densities as high as 1% for the computational or communication resources which may not be unlikely for systems based on molecular electronics made by self-assembly, even though it is a pessimistic estimate for state-of-the-art CMOS silicon based technology.

We will first consider the case where only LBs are defective and the interconnect network is still perfect to access all the modules. Invoking Rent's rule estimates reduction in the number of output terminals available for testing. Assuming a LB defect density of d_{LOGIC}, we have

$$tN_{accessible} \sim T(= t((1 - d_{LOGIC})N)^p) \quad or \quad N_{accessible} \sim ((1 - d_{LOGIC})N)^p \tag{4}$$

In practical situations the term on the right hand is very small and hence a reduction by a factor $(1 - d_{LOGIC})^p$ doesn't change the number of accessible modules significantly (in worst case, for $p = 0.7$ and $d_{LOGIC} = 0.01$, $(1 - d_{LOGIC})^p = 0.9939$).

On the other hand, defects in interconnects and output terminals have more severe consequences. Assuming an interconnect defect density of d_{INT}, the number of accessible LBs available for testing in first phase wouldn't be N^p but $(1 - d_{INT})N^p$. **An important consequence of defective terminals is that not all logic modules can be tested irrespective of whether they are functional or not.** This is explained further in Figures 3 and 4 which shows fraction of LBs tested as a function of interconnect defect density for different Rent exponent. As expected, for smaller Rent exponent we cannot access a significant fraction of LBs (e.g. more than 90% for $p = 0.5$ and interconnect defect density of 0.01). Note the discussion assumes that any surviving and connected device assists in further testing. In realistic designs comprising of heterogenous elements, failure of certain critical elements may cause many more components to be nonfunctional. Thus, the work is an optimistic estimate about the testing complexity. The heterogenous case can be modelled by using local Rent parameters and assigning statistical weights for the circuits.

To evaluate the impact of testing on system performance under a common framework, we use various parameters based on wire length estimation as described in the appendix. Thus, these parameters implicitly depend on the number of logic blocks (N) and Rent exponent (p).

We calculate the redundancy required for a given defect rate from Figure 5 which shows number of LBs available after testing for different Rent exponents and a given defect density. For example, if we require $100K$ logic blocks with a Rent exponent of 0.5 to implement a system, then with a interconnect defect density of 0.01 we will need $1,423K$ logic blocks. Using wire length estimation with the required redundant elements, we can now calculate the overhead in terms of power, delay etc. Note that for a given defect density, a richer interconnect (higher p) requires lesser number of extra LBs. We have chosen system with $p = 0.5$ as the baseline and calculated the required redundancy and relative increase in different parameters for various interconnect defect densities (Figure 6). Note that performance parameters show a non-monotonic behavior due to changes in both the number of LBs and Rent exponent for different defect densities.

Wire length is more sensitive to number of LBs for higher Rent exponent. Putting a richer interconnect for ease of testing can lead to an overhead of $30 - 50$ times than the baseline. This penalty is not mitigated by the lower number of redundant LBs required at higher p. Area is influenced both by Rent exponent and number of LBs. For high defect density the penalty is not trivial. Congestion and routability is an important issue for lower p and higher defect rates. The relative increase can be as high as 100 times for $p = 0.5$. This is due to large amount of redundancy needed for testing the required number of LBs to implement the design.

IEEE COMPUTER SOCIETY

- A higher Rent exponent doesn't come without a price. Figure 6 shows the exponential growth in wire length and other parameters for different Rent exponent. This leads to a trade off between testing complexity and performance.

6: Conclusions and Caveats

The simple model for testing using Rent's power law allows us to draw the following salient and fundamental conclusions:

- Testing is mandatory for reconfigurability to locate the defects. But testing for defect tolerance also puts an overhead on interconnect resources and logic blocks for observability and controllability.
- **First order estimation suggests that the penalty on speed, area, power and routing is intolerable for high defect densities.**
- Defect tolerance through reconfigurability provides a cheap alternative for enhancing yield in the nanoscale era provided a scalable reconfiguration and defect detection scheme can be implemented and the defect rates are very low.

7: Appendix

Wire Length: Recently there had been considerable work [2] [10] in a priori wire length estimation based on Rent's rule where analysis of multi-terminal nets and occupation probability resulted in improvement over Donath's model [5] According to [10], average wire length for the Steiner trees l_{st}^{2D} is given by

$$l_{st}^{2D} = R(p)\frac{1-\gamma}{\gamma}\frac{H(K,p,1)}{H(K,p,2)}, \tag{5}$$

where

$$H(K,p,x) = \frac{2^{K(2p-x)}-1}{2^{2p-x}-1} \quad \& \quad R(p) = 4\frac{2p-3}{2p+1}\frac{4^{2p-1}-(p+2)2^{2p-1}+(p+1)}{4^{2p-1}-(2p+3)2^{2p-1}+(4p+2)}. \tag{6}$$

$K = \log_4 N$ is the total number of levels of hierarchy, p is the Rent exponent and γ is a parameter less than 0.5 denoting fraction of new output terminals versus the total number of new terminals due to the cutting of nets in hierarchical partitioning.

Delay: Delay is determined by the longest length of source-sink pair at the highest hierarchical level excluding the external nets [10]. The expected value of the average length for this level is given by

$$l_K^{2D} = 2^K R(p), \tag{7}$$

where $R(p)$ is given by Equation 6. Since the global and local distributed capacitance are similar, the RC delay of the longest interconnect is proportional to the square of its length. Note that modern VLSI designs extensively use repeaters which makes delay linear with wire length and also cut communication over long wires into multiple cycles.

Power Dissipation: The dominant source of load capacitance in CMOS circuits for sub-100 nanometer era is the wiring capacitance [1]. Assuming a constant activity factor for each capacitive node, the average dynamic power dissipation of the signal interconnects is proportional to the total capacitive load and the clock frequency. Estimation of capacitive load requires detailed knowledge of geometry (pitch, width etc) and interconnect density function. For identical distributed capacitance for

IEEE COMPUTER SOCIETY

each hierarchical level the load can be approximated to be proportional to total wire length [3]. Clock frequency, on the other hand, is inversely proportional to the maximum delay and hence the square of longest wire in the design (Equation 7).

The total net length L can be calculated by summing the product of number of net segments and average length over each hierarchical level.

$$L_{TOTAL} = R(p)(1-\gamma)t4^K(1-4^{p-1})\frac{2^{(2p-1)K}-1}{2^{2p-1}-1}, \tag{8}$$

where $R(P)$, γ, p and K are as defined in Equation 6. And therefore,

$$P_{DYN} \sim \frac{L_{TOTAL}}{(l_K^{2D})^2} = (1-\gamma)t(1-4^{p-1})\frac{1}{R(p)}\frac{2^{(2p-1)K}-1}{2^{2p-1}-1}. \tag{9}$$

Layout Area: Wiring space requirement has been modeled [4] [6] [8] by many using minimum number of wire segments. Simple analytical approaches model wiring area as product of wire length times the gate pitch [3]. In real implementations, gate pitch is different for local and global interconnects. For our analysis we assume same pitch for all levels and therefore the required area will be proportional to the total wire length given by Equation 8.

$$Area \sim L_{TOTAL} \tag{10}$$

Congestion and Routability: As emphasized before interconnect management is a critical parameter for VLSI designs and one of the important issue is ability to efficiently route a given design on the physical architecture. The analysis gives an estimate of the routing resource requirement and utilization. For our analysis we use the congestion model based on cut ratio in recursive bipartitioning [11] as a metric for routability. The method gives an upper bound on maximum routing demand as

$$C_{max} < \frac{C_1(1-\alpha^{2\log_4 N})}{(1-\alpha)}, \tag{11}$$

where $C_1 = \frac{tN}{2}$ is the net cut of first bipartitioning and $\alpha = \frac{C_{i+1}}{c_i} = 2^{-p}$ is the ratio between net cuts of two consecutive partitionings.

References

[1] H. B. Bakoglu. *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley, 1990.

[2] J. Davis, V. De, and J. Meindl. A stochastic wire-length distribution for gigascale intergration - part 1: Derivation and validation. *IEEE Trans. on Electron Devices*, 45(3):580–589, 1998.

[3] J. Davis, V. De, and J. Meindl. A stochastic wire-length distribution for gigascale intergration - part 2: Applications to clock frequency, power dissipation, and chip size estimation. *IEEE Trans. on Electron Devices*, 45(3):590–597, 1998.

[4] W. E. Donath. On the equivalence of memory to random logic. *IBM J. of Res. and Develop.*, 18:401–407, 1974.

[5] W. E. Donath. Placement and average interconnection lengths of computer logic. *IEEE Trans. on Circuit and Systems*, CAS-26:272–277, 1979.

[6] M. Feuer. Connectivity of random logic. *IEEE Trans. on Computers*, C-31:29–33, 1982.

[7] J. R. Heath, P. J. Kuekes, G. S. Snider, and S. Williams. A defect-tolerant computer architecture: Opportunities for nanotechnology. *Science*, 280:1716–1721, 1998.

[8] W. Heller, C. Hsi, and W. Mikhail. Wirability - designing wiring space for chips and chip packages. *IEEE Design and Test Mag.*, pages 43–51, 1984.

[9] A. Kumar and S. Tiwari. Defect tolerance for nanocomputer architecture. In *Proc. of Intl Workshop on System Level Interconnect Prediction (SLIP)*, 2004.

[10] D. Stroobandt. *A Priori Wire Length Estimates for Digital Design*. Kluwer Publications, 2001.

[11] X. Yang, R. Kastnr, and M. Sarrafzadeh. Congestion during top-down placement. *IEEE Trans. on CAD of ICs and Systems*, 21(1):72–80, 2002.

IEEE COMPUTER SOCIETY

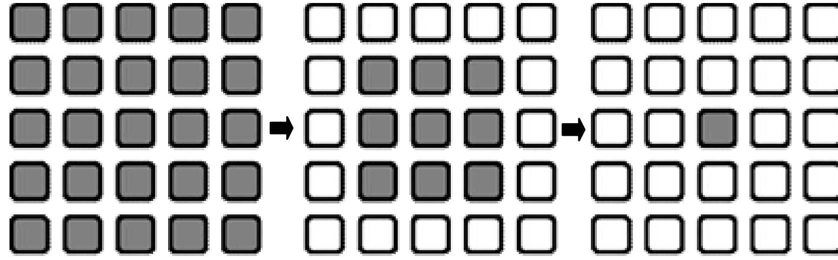

Figure 1: Schematic of testing process for mesh architecture. White squares represent tested blocks.

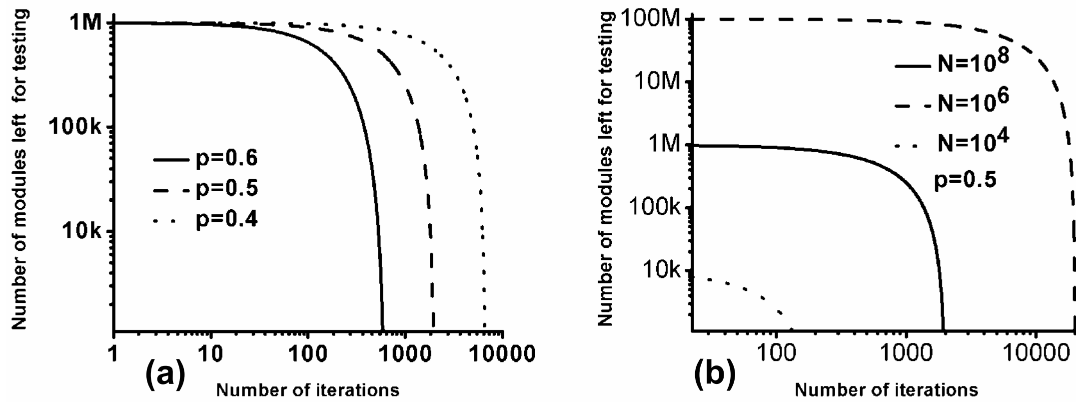

Figure 2: Number of modules left for testing for (a) different Rent exponent (N=1M) (b) different number of logic blocks (p=0.5)

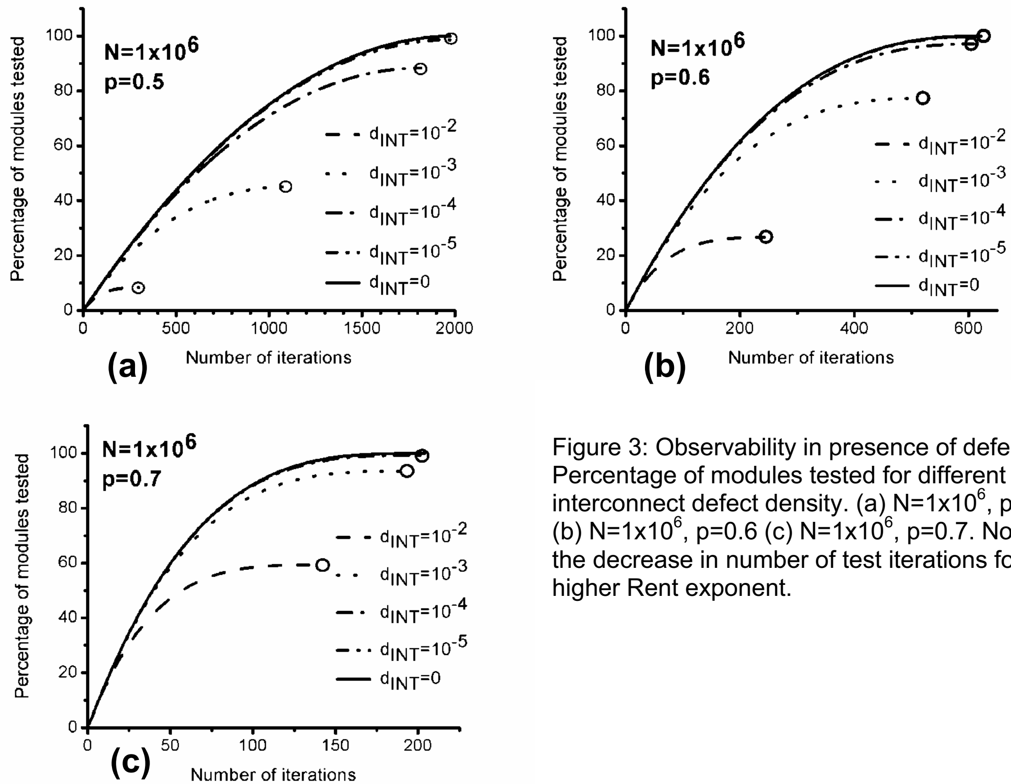

Figure 3: Observability in presence of defects. Percentage of modules tested for different interconnect defect density. (a) $N=1x10^6$, p=0.5 (b) $N=1x10^6$, p=0.6 (c) $N=1x10^6$, p=0.7. Note the decrease in number of test iterations for higher Rent exponent.

IEEE COMPUTER SOCIETY

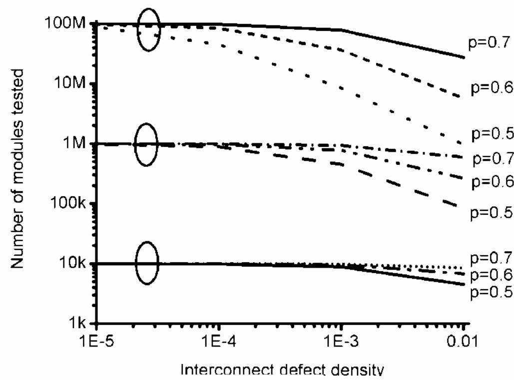

Figure 4: Number of modules tested as a function of interconnect defect density for $N=1x10^4$, $N=1x10^6$ and $N=1x10^8$ (based on Figure 3).

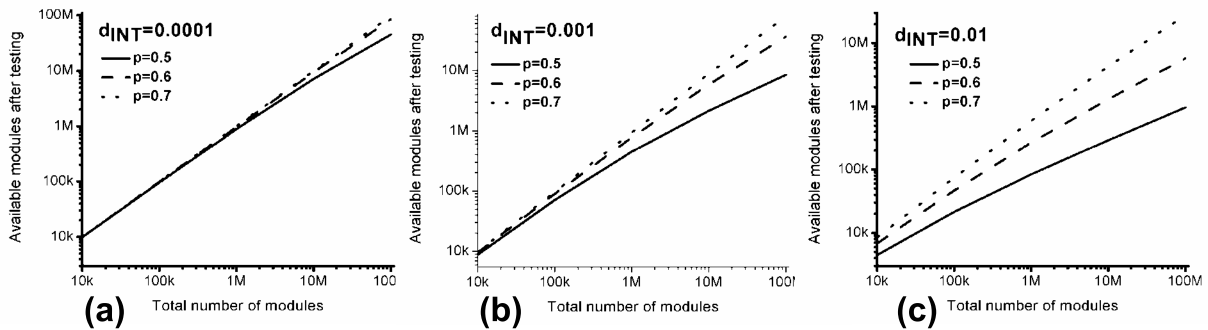

Figure 5: Redundancy requirement for different defect density and Rent exponent as a function of number of logic blocks. (a) d_{INT}=0.0001 (b) d_{INT}=0.001 and (c) d_{INT}=0.01

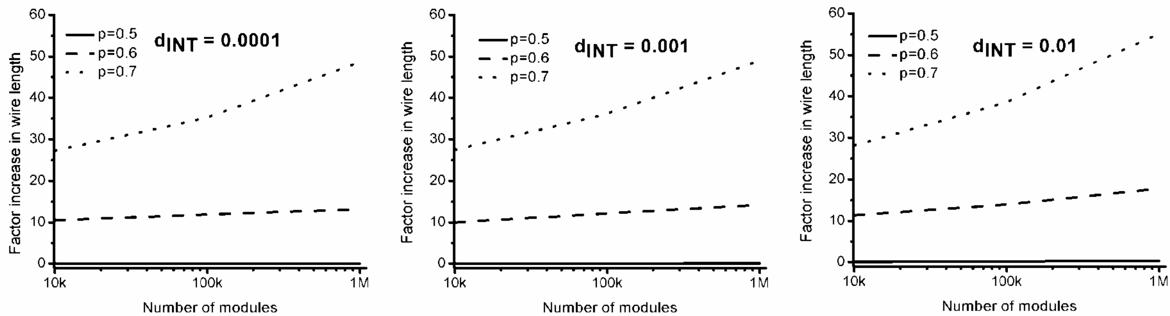

Figure 6: (a) Factor increase in wire length

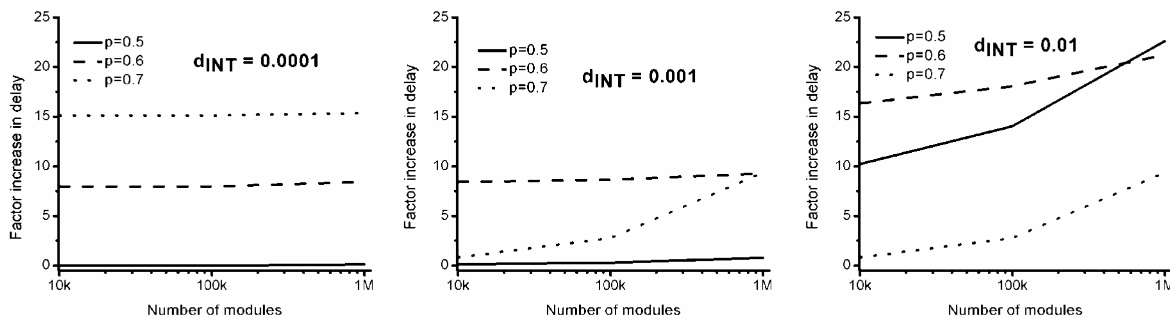

Figure 6: (b) Factor increase in delay

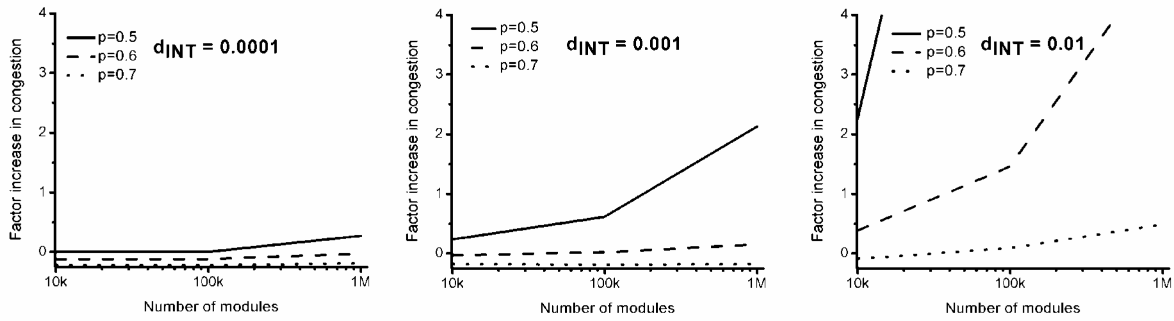

Figure 6: (c) Factor increase in area

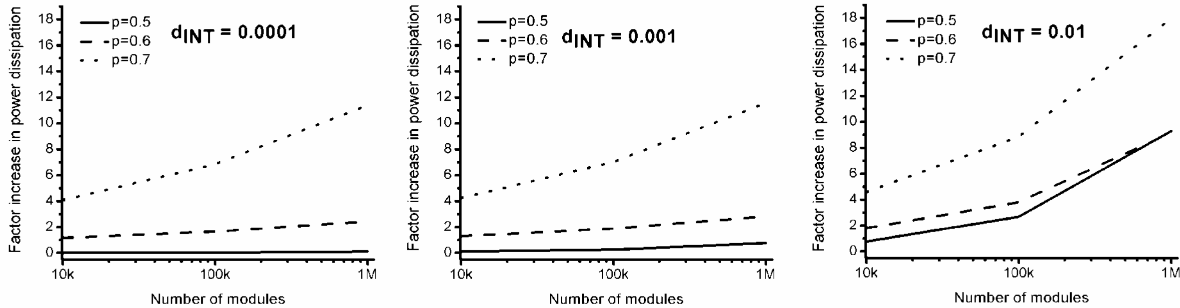

Figure 6: (d) Factor increase in power dissipation

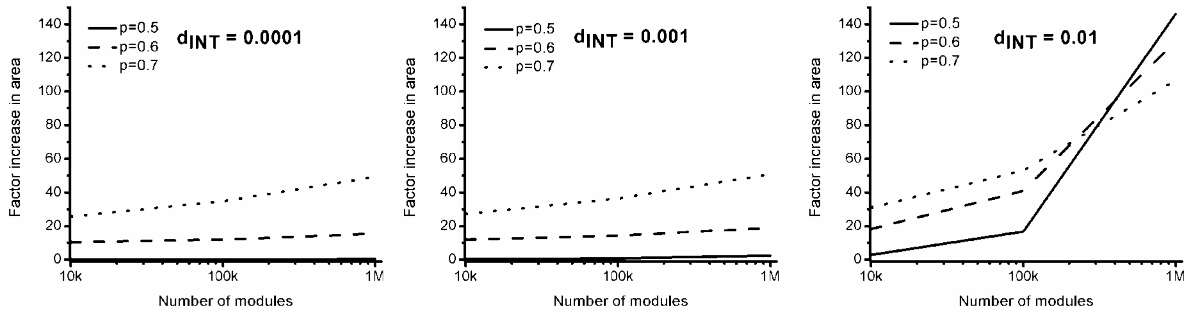

Figure 6: (e) Factor increase in congestion and routability

	d_{INT}=0.0001			d_{INT} =0.001			d_{INT} =0.01		
Rent exponent	0.5	0.6	0.7	0.5	0.6	0.7	0.5	0.6	0.7
	Wire length								
10K	~0	10.55	27.25	0.02	10.02	27.49	0.13	11.39	28.13
100K	~0	11.91	35.24	0.04	12.20	36.14	0.24	13.97	38.59
1M	0.02	13.21	48.59	0.08	14.2	48.95	0.37	17.81	55.12
	Delay								
10K	~0	7.96	15.11	0.11	8.43	0.80	10.24	16.33	0.80
100K	~0	7.96	15.11	0.27	8.65	2.77	14.03	18.06	2.77
1M	0.12	8.43	15.36	0.77	9.29	9.36	22.63	21.25	9.36
	Area								
10K	~0	10.38	25.73	0.27	11.81	27.27	2.69	18.30	31.03
100K	~0	11.87	34.83	0.69	14.24	36.42	16.75	40.94	53.04
1M	0.29	15.38	49.27	2.44	18.97	50.85	146.25	129.68	106.54
	Power dissipation								
10K	~0	1.16	4.09	0.11	1.29	4.25	0.76	1.80	4.61
100K	~0	1.65	6.84	0.26	1.89	7.03	2.69	3.82	8.84
1M	0.12	2.45	11.38	0.76	2.83	11.55	9.28	9.15	17.95
	Congestion								
10K	~0	-0.12	-0.22	0.24	-0.03	-0.18	2.26	0.39	-0.09
100K	~0	-0.12	-0.22	0.62	0.02	-0.19	13.25	1.46	0.09
1M	0.27	-0.03	-0.19	2.13	0.15	-0.18	106.51	5.08	0.49

Figure 6: Relative increase in various system parameters for different defect density. System with p=0.5 and no defect is used as baseline. Redundancy requirements are extracted from Figure 5.

IEEE COMPUTER SOCIETY