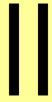
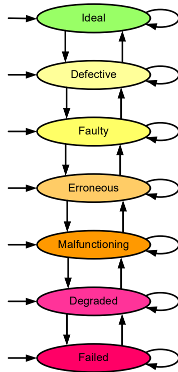


Defects: Physical Imperfections



"[An engineer] recognizes, unfortunate though they may be, that defects are unplanned experiments that can teach one how to make the next design better."

Henry Petroski, To Engineer Is Human, 1985

"The search for perfection begins with detecting imperfection."

Anonymous

Chapters in This Part
5. Defect Avoidance
6. Defect Circumvention
7. Shielding and Hardening
8. Yield Enhancement

Any manufactured component, particularly if mass produced, is bound to have killer and/or latent defects. A killer defect renders the part completely useless, whereas latent defects potentially disrupt its operation or performance at a later stage. Understanding the nature, and learning how to manage the consequences, of such defects is the starting point of our journey through the six undesirable states in the multilevel model of dependable computing. Advances in integrated-circuit and other technologies used to build computers and computer-based systems have unfortunately not helped in this arena: if anything, the exponentially increasing complexities and much greater densities have made defects more prevalent and harder to detect. In this part, after discussing general concepts relating to defect avoidance and circumvention, we study a class of avoidance methods based on shielding and hardening of components and then consider how the harsh impact of defects on integrated circuit yield can be mitigated through redundancy and switch-based reconfiguration methods.

5 Defect Avoidance

"Better a diamond with a flaw than a pebble without one."

Chinese proverb

"The omission of the silicon that had been put in nickel [core of the cathode] to make processing easier . . . raised the effective life of vacuum tubes from 500 hours to 500,000 hours. The marginal checking gave another factor of ten on that."

Jay W. Forrester, reflecting on the Whirlwind I computer, 1983

Topics in This Chapter

5.1. Types and Causes of Defects

5.2. Yield and Its Associated Costs

5.3. Defect Modeling

5.4. The Bathtub Curve

5.5. Burn-in and Stress Testing

5.6. Active Defect Prevention

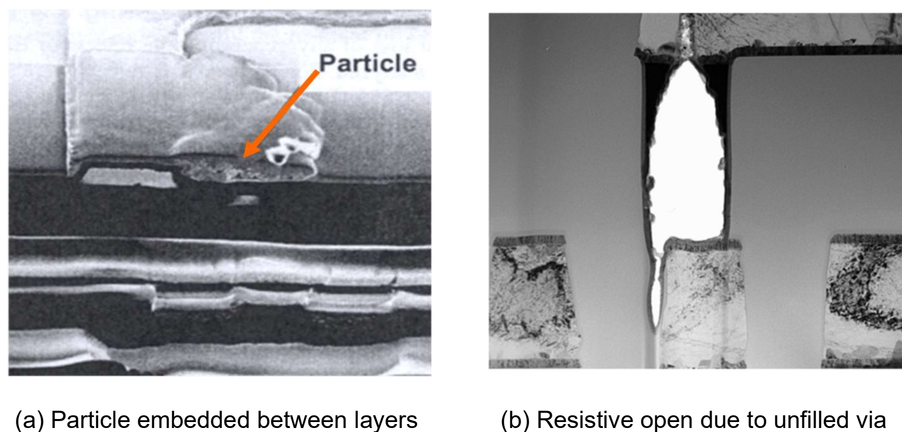
Complete defect avoidance, if possible, would be the preferred choice for dealing with dependability concerns at the device level. Unfortunately, however, defect-free devices and components may be very expensive (due to stringent manufacturing and/or careful screening requirements) and perhaps even impossible to build. Thus, we do what we can, within technical and budgetary constraints, to reduce the occurrence of defects. We then handle what remains through accurate modeling and appropriate circumvention techniques. In this chapter, we deal with the understanding, detecting, and modeling of defects, leaving the discussion of defect circumvention techniques to Chapter 6.

5.1 Types and Causes of Defects

Defects can be viewed as imperfections or weaknesses that may lurk around without causing any harm, but that can potentially give rise to faults and other undesirable behavior. Any manufactured part is prone to defects, particularly if it is mass produced. In this section, we review the types of defects that one finds in integrated circuits and in certain mass storage devices as prominent examples of what can go wrong during manufacturing and how the presence of defects can be detected.

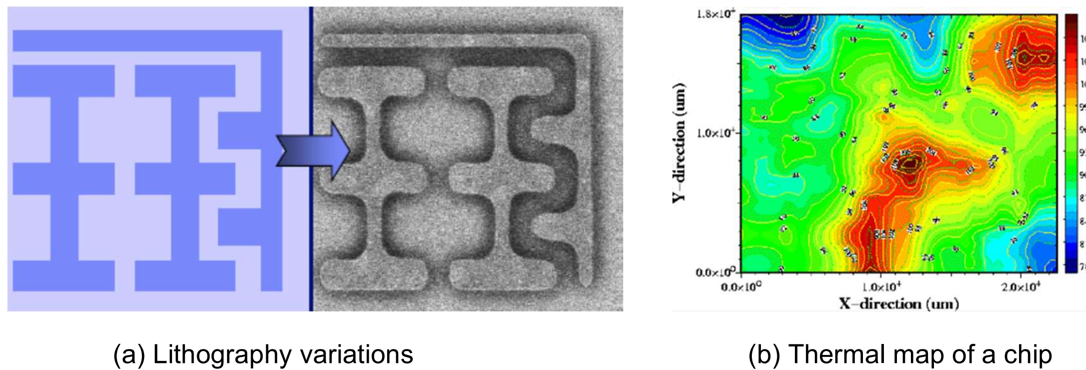
Defects in integrated circuits

Modern integrated circuits are produced via a sequence of complex processing steps involving the deposition of material and structures in layers, beginning with a substrate. As the structures shrink in size, things can and do go wrong. Small impurities in the material involved, tiny particles or air bubbles, or even the natural variations associated with automatic production can lead to problems. Figure 5.1 shows two examples of defects in modern ICs that affect the circuit elements deposited on a surface and vertical interconnections between different layers. Figure 5.2a stresses the fact that ideal designs often gain nonuniformity through the mass production process, making circuit parameters and other aspects of the system different from one point on the chip to another. The same absolute difference in physical dimensions and shapes becomes much more serious in relative terms as the technology is scaled down. Figure 5.2b shows the temperature distribution across a chip. Because speed and other circuit parameters are affected by temperature, the effect of temperature variations is similar to those of nonuniformity resulting from the finite precision of the manufacturing process.



(a) Particle embedded between layers (b) Resistive open due to unfilled via

Fig. 5.1 Typical defects in high-density integrated circuits.



(a) Lithography variations (b) Thermal map of a chip

Fig. 5.2 Process and run-time variations can lead to subtle defects, and associated performance problems, arising from changes in resistance, capacitance, and other parameters.

Detection of integrated-circuit defects is a challenging proposition. Some obvious defects may be detectable through visual inspection. Most defects, however, require much more elaborate methods. Inspection via high-resolution imaging systems and testing of circuit parameters (as opposed to its functionality, which may be unaffected by the presence of defects) are some possibilities. One approach of the latter variety for CMOS circuits is I_{DDQ} testing, where the method's name arises from the measurement of the supply current I_{DD} in the quiescent state. When a defect-free CMOS digital circuit is in quiescent state, static current between the power supply and ground should correspond to a small amount of leakage. Many common manufacturing defects lead to a significant rise in this static current, thus facilitating their detection. Experience has shown that I_{DDQ} testing reveals not only manufacturing defects but also certain logic-level faults that are otherwise difficult to detect by tests based on the stuck-at fault model (see Section 9.1).

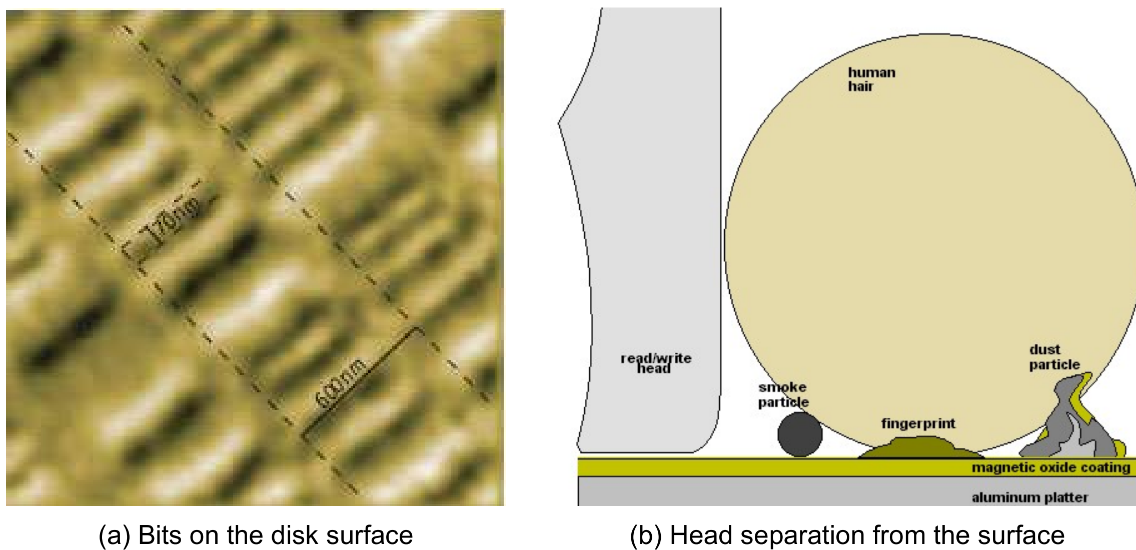
Because defects may not be directly noticeable, one approach to their detection is to intentionally push the system from defective to faulty or erroneous state in the multilevel model of Fig. 1.6, so as to make the system state more observable. Such burn-in or stress tests will be discussed in Section 5.5.

Besides on-chip defects discussed thus far, defects can occur in elements found at higher levels of the digital system packaging hierarchy, including in connectors, printed-circuit boards, cabling, and enclosures. However, these types of defects are deemed less serious, because they lend themselves to easier detection though visual inspection and assembly- or system-level testing.

Defects in disk storage devices

The currently dominant technologies for mass storage devices consist of writing data on smooth surfaces, using magnetic or optical recording schemes. Like integrated circuits, these recording schemes have also experienced exponential density improvements, recording more and more bits per unit area. When the rectangular area devoted to the storage of a single bit has sides that are measured in nanometers (see Fig. 5.3a), slight impurities, surface variations, dust particles, and minute scratches can potentially wipe out thousands of bits worth of stored data. It would be utterly impractical to discard every manufactured magnetic or optical disk if it contained the slightest surface defect.

To see the magnitude of the problem resulting from the high recording density on a hard magnetic disk, for example, consider the comparative dimensions depicted in Fig. 5.3b. With such minute dimensions, a small scratch on the disk surface can affect multiple tracks and thousands of bits. To make matters worse, the read/write head must be placed very close to the disk surface to allow accurate reading and writing of data with the densities shown in Fig. 5.3a. The read/write head actually flies on a cushion on air, nearly touching the surface of the disk. Slight variations in the surface or the presence of an external particles can cause a head crash, leading to substantial damage to the disk surface and the stored data. This is why modern disk drives are built with tightly sealed enclosures.



(a) Bits on the disk surface (b) Head separation from the surface

Fig. 5.3 The high density and small head separation (less than 1 μm) in magnetic recording storage technology.

Surface defects, and their impact on the stored data, are not unique to magnetic mass storage devices. Similar considerations apply to other storage media, such as CDs and DVDs.

Challenges from disk defects are similar to those faced by IC designers and manufacturers: namely, the detection of these defects and appropriate schemes to circumvent them. Data on disk memory is often encoded using CRC or a similarly strong error-correcting code. When a sector exhibits repeated violations of the code, it may be remapped to a different physical disk location and its original location flagged as unusable. Computer operating systems routinely monitor disk operation, using externally observable characteristics and certain sensor-provided information to avoid serious disk failures and the ensuing data loss. Here is a partial list of monitored parameters in modern disk drives: head flying height (a downward trend often portends a head crash); number of remapped sectors; frequency of error correction via the built-in code. Additionally, the following are signs of mechanical or electrical problems that may lead to future data loss: changes in spin-up time, rising temperatures in the unit, reduction in data throughput.

Modern disk memories typically have strong protection built in against defect-caused data corruption. As shown in Fig. 5.3disk, the protection may span multiple levels, from individual sectors (red protective coding), blocks of sectors (blue), and blocks of blocks of sectors (green).

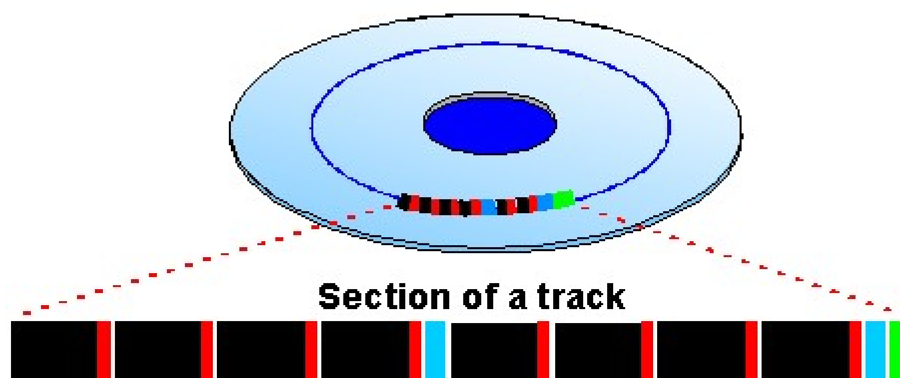


Fig. 5.3disk Multiple levels of protective error-coding in a Hitachi disk. Black areas represent raw (non-redundant) data sectors.

5.2 Yield and Its Associated Costs

The multistep chemical and physical processes that lead from a silicon crystal ingot to a finished IC chip is depicted in Fig. 5.4. Defects on the sliced wafer lead to a certain number of defective dies after the wafer has been patterned (converted into a collection of dies, via a number of processing steps). In the example of Fig. 5.4, 11 of the 120 dies are shown as defective. So, assuming that no other defects arise in the processes of dicing, testing, and mounting the dies, a yield of 109/120 ≈ 91% is achieved.

Example 5.1: Financial gain from yield improvement Consider a company that manufactures 5000 wafers per week. A wafer holds 200 dies, with each good die generating a revenue of $15. Estimate the annual revenue gain from each percentage point in improved yield.

Solution: Each percentage point improvement in yield results in 2 additional good dies per wafer, corresponding to a revenue gain of $30. So, the expected annual revenue gain for a 1% yield improvement is $30 gain/wafer × 5000 wafers/week × 52 weeks/year = $7.8M.

Experimental evidence suggests that the die yield is related to *defect density* (number of defects per unit area) and die area, as shown in equation (5.2.yield). The parameter *a* is a technology-dependent constant which is in the range 3-4 for modern CMOS processes.

$$\text{Die yield} = \frac{\text{Number of good dies}}{\text{Total number of dies}} = \left[1 + \frac{\text{Defect density} \times \text{Die area}}{a}\right]^{-a} \qquad (5.2.\text{yield})$$

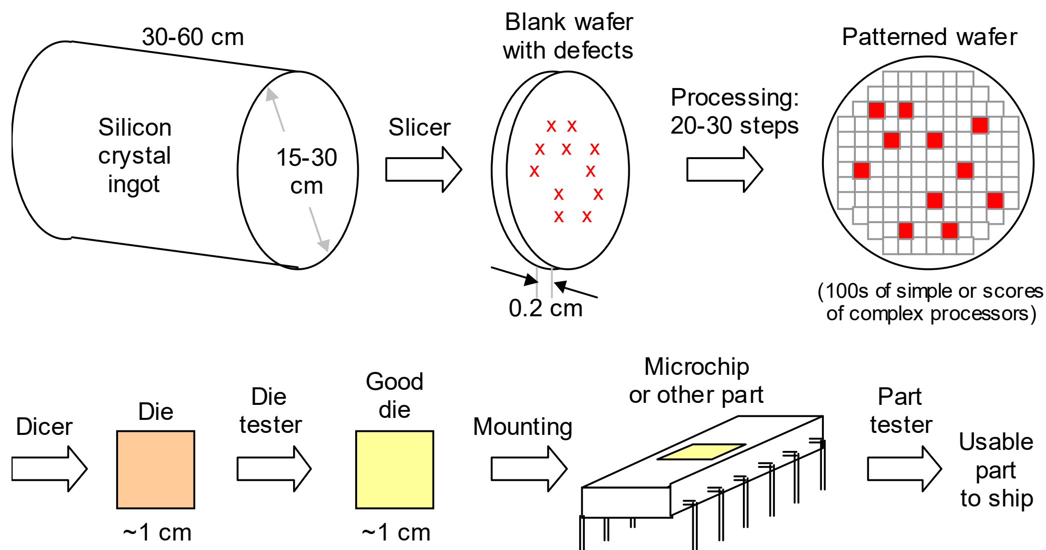


Fig. 5.4 **The manufacturing process for an IC part.**

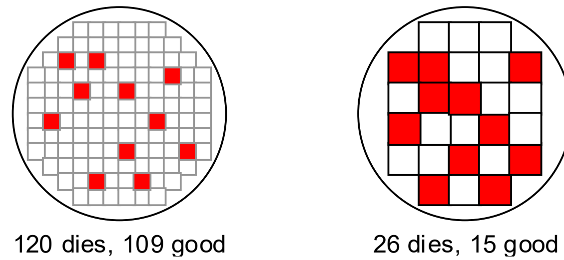


Fig. 5.5 **Why larger dies lead to a dramatic reduction in yield.**

It is this nonlinear relationship that causes die cost to grow superlinearly with the die size (chip complexity). Note that assuming a fixed cost for a wafer and a good yield at the wafer level (i.e., only a small number of wafers have to be discarded), the cost of a die is directly proportional to its area and inversely proportional to the die yield.

$$\text{Die cost} = \frac{\text{Wafer cost}}{\text{Usable wafer area}} \times \frac{\text{Die area}}{\text{Die yield}} \qquad (5.2.\text{cost})$$

Because, according to equation (5.2.yield), die yield is a decreasing function of the die area, the cost of a die will grow superlinearly with its area. This effect is evident from Fig. 5.5, where the same defect pattern has rendered 11 of the dies useless, leading to a much smaller yield in the case of the larger dies of Fig. 5.5b than the smaller dies in Fig. 5.5a. In the extreme of using an entire wafer to implement a single integrated circuit, that is, having one die per wafer, yield becomes a very serious problem. This is why many of the defect circumvention methods, discussed in Chapter 6, were first suggested in connection with wafer-scale integration.

Example 5.2: Effects of dies size on yield and cost Assume that the dies in Fig. 5.5 are 1 × 1 and 2 × 2 cm^2 in size and ignore the defect pattern shown. Assuming a defect density of $0.8/cm^2$, how much more expensive will the 2 × 2 die be than the 1 × 1 die?

Solution: Let the wafer yield be w. From the die yield formula, we obtain a yield of $0.492w$ and $0.113w$ for the 1 × 1 and 2 × 2 dies, respectively, assuming $a = 3$. Plugging these values into the formula for die cost, we find that the 2 × 2 die costs (120/26) × (0.492/0.113) = 20.1 times as much as the 1 × 1 die; this represents a factor of 120/26 = 4.62 greater cost attributable to the smaller number of dies on a wafer and a factor of 0.492/0.113 = 4.35 due to the effect of yield. With $a = 4$, the ratio assumes the somewhat larger value (120/26) × (0.482/0.095) = 23.4.

The aforementioned effect of die size on yield is widely known and duly emphasized in VLSI design courses. Another cost factor associated with yield, however, is often ignored: low yield leads to much higher testing cost, if an overall part quality is to be achieved. This is illustrated in the following example.

Example 5.3: Effects of yield on testing and part reliability Assuming a part yield of 50%, discuss how achieving an acceptable defective part rate of 100 defects per million (DPM) affects the part cost. Include all factors contributing to cost.

Solution: Consider manufacturing 2M parts of which 1M are expected to be defective, given the 50% yield. To achieve the goal of 100 DPM in parts shipped, we must catch 999,900 of the 1M defective parts. Any testing process is imperfect in that the test will miss some of the defects (imperfect test coverage) and will also generate a number of false positives. Thus, we require a test coverage of 99.99%. Going from a coverage of 99.9% (a fraction 10^{-3} or 0.1% of the defects missed) to 99.99% (10^{-4} or 0.01% missed), for example, entails a significant investment in test development and application times. False positives do not constitute a major cost in this particular context, because discarding another 1-2% of the parts due to false positives in testing does not change the scale of the financial loss.

To make the discussion in the solution to Example 5.3 more quantitative, we need to model the testing cost as a function of test coverage (Fig. 5.6). This modeling cannot be done in general, as testing cost depends on the tested circuit's functionality and implementation technology. There is a significant body of research, however, to assist us with this task in specific cases of interest [Agra01].

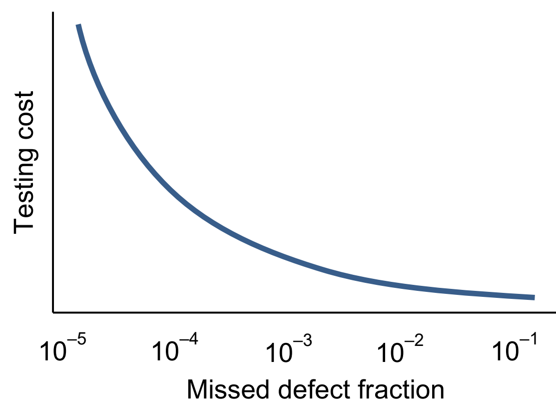


Fig. 5.6 **Testing cost rises sharply with a reduction in the desired fraction of missed defects.**

5.3 Defect Modeling

Defects are of two main types. Global or gross-area defects result from scratches (e.g., from wafer mishandling), mask misalignment, or over/under-etching. Such defects can be eliminated or minimized by appropriate provisions and process adjustments. Local or spot defects result from process imperfections (e.g., extra or missing material), process variations, or effects of airborne particles. Even though spot defects are harder to deal with, not all such defects lead to structural or parametric damage. The actual damage suffered depends on the location and extent of the defect relative to feature size.

Two examples of defect modeling are depicted in Fig. 5.7. Excess material deposited on the die surface can cause physically proximate conductors to become connected. If we model extra-material defects as circles, then the lightly shaded rectangular regions in Fig. 5.7a indicate possible locations for the center of the defect circle of a certain size that would lead to improper connectivity. Pinhole defects result from tiny areas where material may be missing (due to burst bubbles, for example). This may cause problems because missing dielectric material between two vertically adjacent conductors may lead to their becoming connected. Critical regions for pinhole defects, shown as small lightly shaded squares in Fig. 5.7b, correspond to overlapping conductors that are separated by a thin dielectric layer.

Under such assumptions, the modeling process consists of determining the likelihood of having defects that fall in the corresponding critical regions, based on some knowledge about defect kind and size distributions.

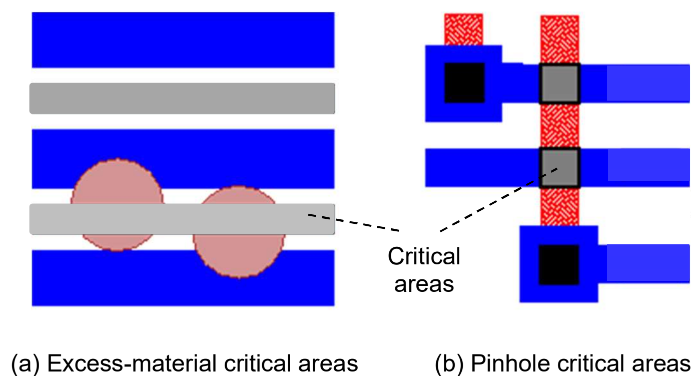


Fig. 5.7 Excess-material defects (modeled as circular areas) and pinhole defects.

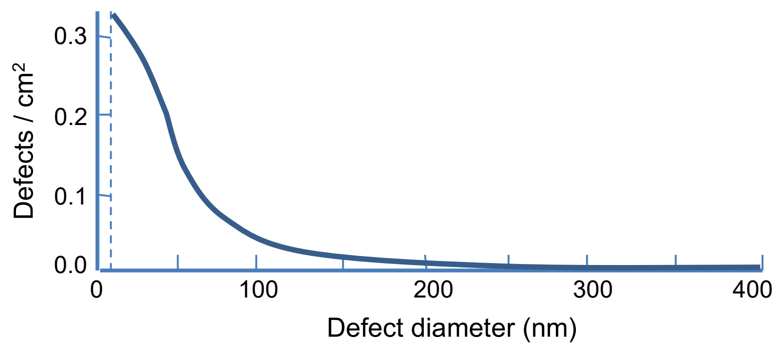


Fig. 5.8 A sample defect size distribution for an overall defect rate of 0.3/cm^2.

Here is an empirical defect-size distribution model. Defects typically range from a minimum size x_{min} to a maximum size x_{max}, with defects outside the range so rare as to have a negligible effect. The defect density $f(x)$ as a function of defect diameter x follows a power law:

$$f(x) = kx^{-p} \text{ for } x_{min} < x < x_{max}\text{; 0 otherwise} \qquad (5.\text{defect1})$$

The exponent parameter p is typically in the range [2.0, 3.5] and k is a normalizing constant. Figure 5.8 depicts a sample defect size distribution, assuming an overall defect density of 0.3/cm^2.

5.4 The Bathtub Curve

Most components or systems do not have a constant failure rate over their lifetimes. A simplified model that accounts for variations in the failure rate over time, known as the *bathtub curve*, is based on the hypothesis that three factors contribute to failure. The first factor, infant mortality, is due to imperfections in system design and construction that may lead to early failures. Taking the analogy of a new car, it happens that mere factory inspections and testing are inadequate for removing all defects, leading to quite a few defective or low-quality cars (the so-called “lemons”) to be marketed and sold. If the particular car you buy survives this early phase, then it will likely function without much trouble for many years.

The second factor, random failures, can arise at any time during a component’s or system’s life due to environmental influences, normal stresses, or load conditions. The constant failure rate λ is often used to model this phase of useful life.

The third factor is the wearing out of devices or circuits, leading to higher likelihood of failures as a component or system ages. As depicted in Fig. 5.btc, the wearout effect is more pronounced for mechanical devices than for electronics. In fact, many computer and communication equipment become obsolete and discarded so quickly that wearout isn’t a significant concern. On the other hand, fatigue or wearout is a major concern for aircraft parts, including those forming the fuselage, and is dealt with by preventive maintenance and periodic replacement. Interestingly, aging or deterioration is not limited to hardware but has also been observed in software, owing to the accumulation of state information (from setting of user preferences, updates, and extensions).

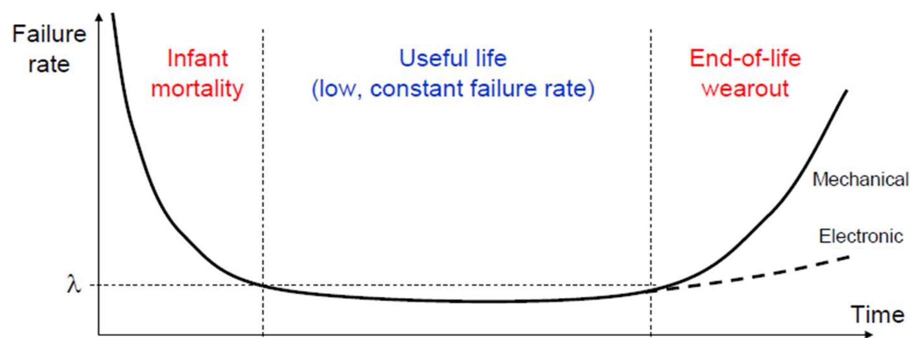


Fig. 5.btc The bathtub curve, showing the three phases of a component’s life.

Using colorful expressions such as “the bathtub curve does not hold water” [Wong88], reliability researchers have been pointing out the weaknesses of the bathtub curve model for quite a long time. Our discussion of the bathtub curve is motivated by the fact that it provides a useful pedagogical tool for drawing attention to infant mortality (and hence the importance of rigorous post-manufacturing tests and burn-in tests) and wearout (often avoided by preventive maintenance and early retirement of devices or systems that are prone to deterioration with age). It also tells us why the constant failure rate assumption might be appropriate during most systems’ post-burn-in, pre-wearout, useful lives.

Referring to Fig. 5.burn1, we note the effect of infant mortality on the reliability function, driving home the point that unless we deal with the infant mortality problem, achieving high reliability would be impossible.

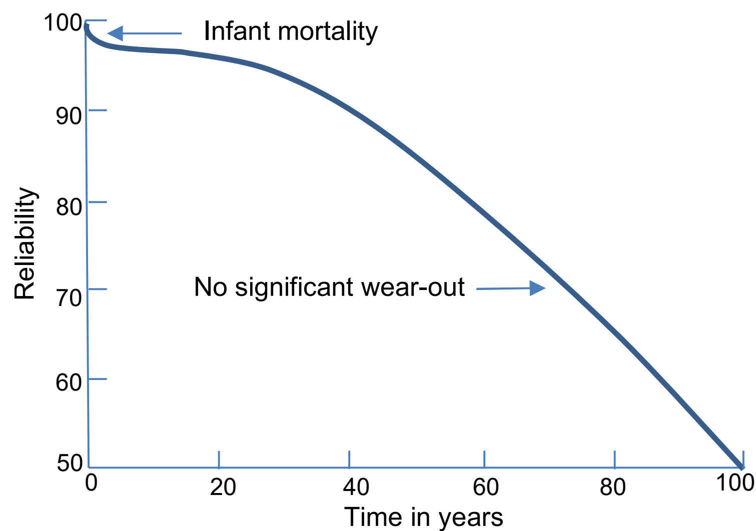


Fig. 5.burn1 Survival probability of electronic components.

5.5 Burn-in and Stress Testing

In order to expose existing and latent defects that lead to infant mortality, one needs to test a component or system extensively under various environmental and usage scenarios. An alternative to such extended testing, which may take an unreasonable amount of time, is to expose the product to abnormally harsh conditions in an effort to accelerate the exposure of defects. The name “burn-in” comes from the fact that for electronic circuits, testing them under high temperatures (literally in ovens) is commonly used, given that intense heat can accelerate the failure processes. In the extended sense, “burn-in” refers to any harsher-than-normal treatment, including using greater loads, higher clock frequencies, excessive shock and vibration, and so on.

The ovens used for high-temperature burn-in testing of electronic devices and systems are quite elaborate and expensive, as they require fine controls to avoid damaging sensitive parts in the circuits under test.

As depicted in Fig. 5.burn2, components that survive burn-in testing, will be left with very few residual defects that could potentially lead to early failures.

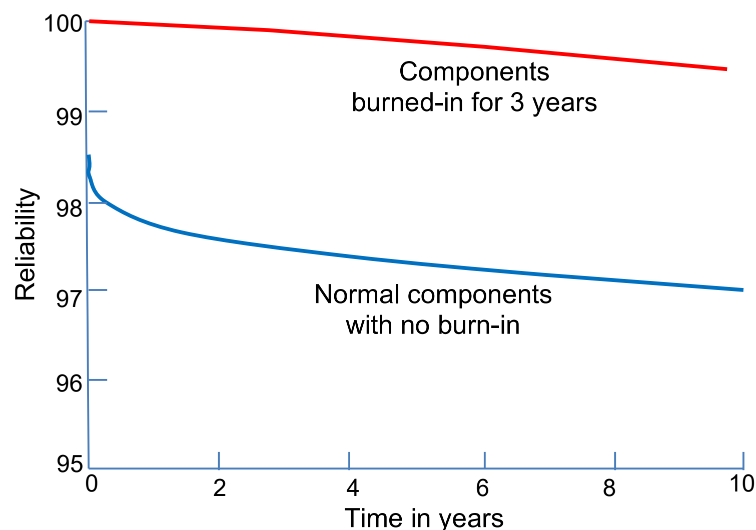


Fig. 5.burn2 Survival probability of electronic components.

5.6 Active Defect Prevention

Besides initial or manufacturing imperfections, wear and tear during the course of a device's lifetime can lead to the emergence of defects. A harsh operating environment or excessive load may speed up the development of defects. Such conditions can sometimes be counteracted by operational measures such as temperature control, load redistribution, or clock scaling. Radiation-induced defects can be minimized by proper shielding or hardening (see Chapter 7) and those resulting from mishandling, shock, or vibration can be mitigated by encasing, padding, or mechanical insulation.

One of the most commonly used strategies for active detect prevention is periodic or preventive maintenance. Preventive maintenance forestalls latent defects from developing into full-blown defects that produce faults, errors, and so on. To grasp the role of preventive maintenance for computer parts, consider that passenger aircraft parts are routinely replaced according to a fixed maintenance schedule so as to avoid fatigue-induced failures. So, an aircraft engine may be replaced at the end of its nominal service period, even though it exhibits no signs of impending failure. Referring to the bathtub curve of Fig. 5.btc, this is akin to resetting the clock and avoiding the wearout phase of the curve for the replaced part. For this strategy to be effective, however, we must also make sure to avoid the infant mortality phase of the new engine by subjecting it to rigorous burn-in and stress testing.

Given that preventive maintenance has as associated cost in terms of personnel and lost system functionality, many studies have been performed to optimize the maintenance schedule under various cost models and system characteristics, including whether the preventive maintenance is perfect (rendering the system "like new") or imperfect (e.g., reducing the *effective age* of the system that dictates its hazard rate, but not fully resetting it to zero [Bart06]). Often, the resulting models for maintenance optimization are too complex for analytical solution, necessitating the use of numerical solutions.

Problems

5.1 Defects and yield

Every three years, from 1980 to 1992, DRAM chips increased in capacity by a factor of 4 and in die area by a factor of about 1.5 (beginning with 64 Kb and 0.15 cm^2 in 1980), while yields remained virtually constant at around 45%. Assuming that these trends have continued up to the present day, what can you say about trends in defect density and memory cost? State all your assumptions.

5.2 Defect modeling

In Section 5.3, it was noted that the density function for the defect size (diameter) x is usually taken to be $f(x) = kx^{-p}$, for $x_{min} < x < x_{max}$, where k, p, x_{min}, and x_{max} are constants.

a. Derive the value of the constant k in terms of the other three constants and the overall defect density d.

b. Estimate the numerical value of k for the sample defect distribution shown in Fig. 5.8.

5.3 Yield variation with die size

Figure 5.5 and Example 5.2 show the effect of increasing the die size from 1×1 cm^2 to 2×2 cm^2.

a. With the same assumptions as in Example 5.2, calculate the yield and relative die cost for 3×3 square dies.

b. Repeat part a for 2×4 rectangular dies.

5.4 Effects of yield on die cost

A wafer containing 100 copies of a complex processor die costs $900 to manufacture. The area occupied by each processor is 2 cm^2 and the defect density is 2/cm^2. What is the manufacturing cost per die?

5.5 Number of dies on a wafer

Consider a circular wafer of diameter d. The number of square dies of side u on the wafer is upper-bounded by $\pi d^2/(4u^2)$. The actual number will be smaller because there are incomplete dies at the edge.

a. Argue that $\pi d^2/(4u^2) - \pi d/(1.414u)$ is a fairly accurate estimate for the number of dies.

b. Apply the formula of part a to the wafers shown in Figure 3.9 to obtain an estimate for the number of dies and determine the error in each case. The dies are 1×1 and 2×2 and $d = 14$.

c. Suggest and justify a formula that would work for nonsquare $u \times v$ dies (e.g., 1×2 cm^2).

d. Is it possible to make a general statement about whether square dies or rectangular dies of the same area waste less space on a wafer?

5.6 Yield modeling

Parts a-c of this problem provide die sizes and associated yields for 3 different DRAM chips over time. Estimate the corresponding defect density in each case.

a. 1 Gb DRAM chip: TBD

b. 4 Gb DRAM chip: TBD

c. 16 Gb DRAM chip: TBD

d. Derive a formula for the required defect density if the yield for an x Gb DRAM were to be 90% and provide numerical values for the examples in parts a-c.

5.7 An alternative yield model

Consider another proposed yield model, in which die area is measured in cm^2 and defect density is given per cm^2: Die yield = $e^{-\text{Defect density} \times \text{Die area}}$

a. Provide an analysis that shows when this alternate model provides results that are nearly the same as those from equation (5.2.yield).

b. Under what conditions would the two models provide significantly different results?

c. Which yield model do you think is more realistic?

5.8 Defect modeling

Intro

a. x

b. x

c. x

5.9 The bathtub curve

Which failure distribution can be used to model all three parts of a bathtub curve? Explain.

5.10 The bathtub curve

The bathtub curve can be viewed as the summation of three curves: a declining curve that represents failures resulting from problems that exist in a product or system at time zero, a horizontal line corresponding to random failures due to loading and operational stress, and a rising curve corresponding to failures caused by wear and tear.

a. Research this idea for a particular system or class of systems.

b. Model the three parts using available or estimated parameters.

c. Verify that the sum of the three parts does yield a bathtub curve.

5.11 The roller-coaster curve

The roller-coaster curve was mentioned in Section 5.4 as a proposed substitute for the bathtub curve.

a. Present diagrams depicting the typical shapes of the roller-coaster curve.

b. Explain failure mechanisms and other reasons for advocating the roller-coaster curve over the bathtub curve.

c. In terms of probability distributions, how can the roller-coaster curve be modeled?

d. How is the modeling of the roller-coaster curve different from that of the bathtub curve?

e. Try to discover if there are any arguments against using the roller-coaster curve.

5.12 Burn-in testing

Intro

a. x

b. x

c. x

d. x

5.13 Burn-in testing

Intro

a. x

b. x

c. x

5.14 Preventive maintenance

Intro

a. x

b. x

c. x

d. x

5.15 Preventive maintenance

Intro

a. x

b. x

c. x

d. x

5.16 Cost and benefits of yield improvement

In example 5.1, we analyzed the financial gain from yield improvement, pretending that improving the yield costs nothing. In reality, improving the yield does have a cost that rises sharply with the target yield, exhibiting a trend similar to Fig. 5.6 (view the horizontal axis as representing the fraction of bad parts, so that 10^{-1} corresponds to a yield of 90%).

a. Discuss the reasons for the much sharper rise in the cost of yield improvement when yield is already quite high.

b. Try to find the parameters of a cost model for yield improvement and reconsider Example 5.1 in light of this cost.

5.17 Probabilistic design

PCMOS (probabilistically correct CMOS logic) adders have been designed that produce an incorrect value 0.25% of the time but consume 3.5 times less power. Further relaxing the probability of correctness to 0.92 leads to a factor-of-15 reduction in power [Anth13]. Additionally, noting that not all bits are created equal allows us to focus on the more-significant bits at the expense of the less-significant bits, which then become less reliable. See also [Pale09].

a. What is the reason for reduced energy consumption with probabilistic design?

b. Is the method applicable to both fixed-point and floating-point computations?

c. How does this method differ from lower-precision or adjustable-precision computation?

d. Name a few applications that would benefit from probabilistic design.

5.18 Defect classes

A consumer organization estimates that 29% of new cars delivered to dealers have a cosmetic defect, such as a scratch or a dent, while 7% have functional defects related to a part or subsystem that does not work properly. That same consumer organization estimates that 2% of all cars have defects of both types.

a. What is the probability that a new car has some kind of defect?

b. What is the probability that a new car has a cosmetic defect but no functional defect?

c. If your new car has a dent, what is the probability that it also has a functional defect?

d. Do you think that cosmetic and functional defects are independent of each other? Explain.

References and Further Readings

[Agra01] Agrawal, V. D., S. C. Seth, and P. Agrawal, “Fault Coverage Requirements in Production Testing of LSI Circuits,” *IEEE J. Solid-State Circuits*, Vol. 17, No. 1, pp. 57-61, May 2001.

[Anth13] Anthes, G., “Inexact Design—Beyond Fault-Tolerance,” *Communications of the ACM*, Vol. 56, No. 4, pp. 18-20, April 2013.

[Bart06] Bartholomew-Biggs, M., B. Christianson, and M. Zuo, “Optimizing Preventive Maintenance Models,” *Computational Optimization and Applications*, Vol. 35, No. 2, pp. 261-279, 2006.

[Cici95] Ciciani, B., *Manufacturing Yield Evaluation of VLSI/WSI Systems*, IEEE Computer Society Press, 1995.

[Eler07] Elerath, J., “Hard Disk Drives: The Good, the Bad and the Ugly,” *ACM Queue*, Vol. 5, No. 6, pp. 28-37, September-October 2007.

[Ghos10] Ghosh, S. and K. Roy, “Parameter Variation Tolerance and Error Resiliency: New Design Paradigm for the Nanoscale Era,” *Proc. IEEE*, Vol. 98, No. 10, pp. 1718-1751, 2010.

[Hawk94] Hawkins, C. F., J. M. Soden, A. W. Righter, and F. J. Ferguson, “Defect Classes—An Overdue Paradigm for CMOS IC Testing,” *Proc. IEEE Int’l Test Conference*, 1994, pp. 413-425.

[Kore07] Koren, I., and C. M. Krishna, *Fault-Tolerant Systems*, Morgan Kaufmann, 2007.

[Liew00] Liew, T., S. W. Wua, S. K. Chowa, and C. T. Lim, “Surface and Subsurface Damages and Magnetic Recording Pattern Degradation Induced by Indentation and Scratching,” *Tribology International*, Vol. 33, No. 9, pp. 611-621, September 2000.

[Sode95] Soden, J. M. and C. F. Hawkins, “I_{DDQ} Testing and Defect Classes—A Tutorial,” *Proc. Custom Integrated Circuits Conf.*, May 1995, pp. 633-642.

[Khar96] Khare, J. B. and W. Maly, *From Contamination to Defects, Faults and Yield Loss: Simulation and Applications*, Kluwer, 1996.

[Klut03] Klutke, G.-A., P. C. Kiessler, and M. A. Wortman, “A Critical Look at the Bathtub Curve,” *IEEE Trans. Reliability*, Vol. 52, No. 1, pp. 125-129, March 2003.

[Pale09] Palerm, K., et al., “Sustaining Moore’s Law in Embedded Computing through Probabilistic and Approximate Design,” *Proc. Int'l Conf. Compilers, Architectures, and Synthesis for Embedded Systems*, October 2009.

[Sham09] Sham, K.-J., “Crosstalk Mitigation Techniques in High-Speed Serial Links,” M.S. Thesis, Univ. of Minnesota, April 2009.

[Wang10] Wang, L.-T., C. E. Stroud, and N. A. Touba. System-on-Chip Test Architectures: Nanometer Design for Testability, Morgan Kaufmann, 2010. Section 8.3, “Manufacturing Defects, Process Variation, and Reliability.”

[Wong88] Wong, K. L., “The Bathtub Curve Does Not Hold Water Any More,” *Quality and Reliability Engineering Int’l*, Vol. 4, No. 3, pp. 279-282, July/September 1988.

6 Defect Circumvention

"We grow tired of everything but turning others into ridicule, and congratulating ourselves on their defects."

William Hazlitt

"We have learned to live in a world of mistakes and defective products as if they were necessary to life. It is time to adopt a new philosophy in America."

Norman Cousins

"The flaw which is hidden is deemed greater than it is."

Marcus Aurelius

Topics in This Chapter
6.1. Detection of Defects
6.2. Redundancy and Reconfiguration
6.3. Defective Memory Arrays
6.4. Defects in Logic and FPGAs
6.5. Defective 1D and 2D Arrays
6.6. Other Circumvention Methods

As the densities of integrated circuits and memory devices rose through decades of exponential growth, defect circumvention methods took on increasingly important roles. Today, it is nearly impossible to build a defect-free silicon wafer or a perfectly uniform magnetic-disk platter. So methods for detecting defective areas, and avoiding them via initial configuration or subsequent reconfiguration, are in widespread use. Defect circumvention methods covered in this chapter have a great deal in common with switching and reconfiguration schemes employed at the level of modules or nodes in parallel and distributed systems. Our focus here is on methods that are particularly suited to fine-grain circumvention.

6.1 Detection of Defects

Defects are detected as a result of post-manufacturing inspections and testing, as well as during normal system operation.

When a wafer emerges from the manufacturing process, visual inspections are performed to identify obvious defects. During this phase, the inspector (human or machine) focuses on the more problematic areas, such as the edges of a wafer.

Defect avoidance and circumvention methods are complementary. Avoidance schemes include defect awareness in design, particularly in the floor planning and routing phases, extensive quality control during the manufacturing process, and comprehensive screening, including burn-in and stress tests. Defect circumvention methods fall under the two strategies of defect removal and defect masking. To remove defects, we must first identify them and then use built-in resources to bypass or disable the defective parts. This approach is very similar to dynamic hardware redundancy at the module or system level. Masking of defects requires static redundancy on the die or wafer. In this scheme, defective parts continue to operate, but their effect is voided or muted by other healthy parts that operate redundantly. Several examples of defect removal and masking techniques will be discussed in the remaining sections of this chapter.

6.2 Redundancy and Reconfiguration

Providing redundant components or cells, plus a capability to avoid or route around bad elements is one way of avoiding defects. This approach is best-suited to systems that have a regular or repetitive structure on the die. Examples include memories, FPGAs, multicore chips, and chip-multiprocessors. Irregular or random logic implies greater redundancy arising from replication, with the interconnect problem exacerbated by the need for the replicated structures not to be too close to each other (to minimize common-cause defects).

A good example of the redundancy and reconfiguration approach to defect circumvention is the method used to avoid bad sectors on a disk memory. Bad sectors are identified by error detection during read operations. Post-manufacturing tests typically detect a number of bad sectors that are included in the so-called P-list (permanent or primary defect table). Such initially damaged sectors do not form part of disk system's storage capacity and have no impact of its performance, given that performance data and guarantees already include the effect of such sectors. As the disk is used, other defective sectors emerge, whose addresses are included in the so-called G-list (growth or post-use defect table) by the disk controller. Upon a disk write operation, such a bad sector is replaced with a spare sector, with all subsequent accesses to it automatically redirected to the new location. Because of this redirection, the presence of bad sectors in the G-list slows down access to the data and affects the overall performance. Once the disk runs out of spare sectors, its defect circumvention capacity has been exhausted and the entire disk must be replaced.

Example 6.1: Disk sector remapping Assume that bad disk sectors are detected with 100% probability and that there is a hard limit on the number of remapped sectors due to performance concerns. Suggest a reliability model for the disk.

Solution: To be provided.

6.3 Defective Memory Arrays

Early semiconductor memories were less reliable than their immediate predecessors (magnetic core memories). Thus, methods of dealing with defective bit cells in such memories were developed early on. One class of methods involving error-detecting/correcting codes will be discussed in Chapters 13 and 14. Here, we focus on defect circumvention methods that allow us to bypass defective memory cells, assuming that their presence is detected via appropriate tests or via concurrent error detection.

A commonly used scheme is to provide the memory array (as a whole or in subarrays of smaller size) with spare rows and columns of cells. In the example of Fig. 6.masrc, the memory array is shown to consist of two subarrays, each with its dedicated spare rows and columns. When a bad memory cell is detected, the entire row or column containing it is replaced with one of the spares. The choice of using a spare row or a spare column is arbitrary when there is an isolated bad cell, whereas in the case of multiple cell defects in the same row/column, one approach can be more efficient than the other. Switches at the periphery of the array or subarray allow external connections to be routed to the spare row/column in lieu of the one containing the bad memory cell(s). There are also defects in wiring and other row/column access mechanisms that may disable an entire row or column, in which case the choice of replacement is obvious.

Let us focus on an array or subarray with m data rows and s spare rows. Assuming perfect switching and reconfiguration, the redundant arrangement can be modeled as an m-out-of-$(m + s)$ system. The modeling becomes somewhat more complex when we have both spare rows and columns, but the relevant models are still combinational in nature.

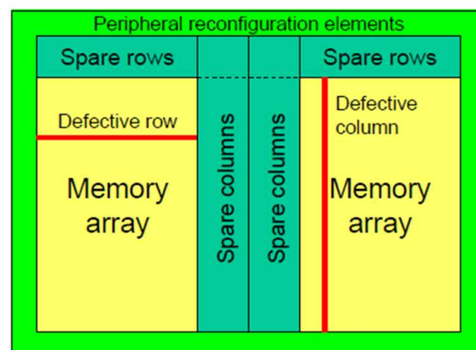


Fig. 6.masrc Memory array with spare rows and columns.

Given a particular pattern of memory cell defects, finding the optimal reconfiguration is nontrivial. We will discuss the pertinent methods in connection with yield enhancement for semiconductor memory chips in Chapter 8.

Example 6.2: Reliability modeling for redundant memory arrays Statement to be provided..

Solution: To be provided.

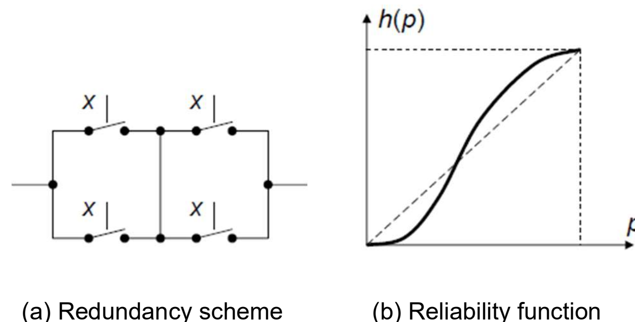
6.4 Defects in Logic and FPGAs

Moore and Shannon, in their pioneering work on the reliability of relay circuits [Moor56] showed how one can build arbitrarily reliable circuits from unreliable, or in their word, “crummy,” relays. Consider relays that are prone to short-circuiting when they are supposed to be open. Let the probability of such an improper short-circuiting event be p. Then, the relay circuit of Fig. 6.M&S will experience a similarly defective behavior (i.e., short-circuiting) with probability

$$h(p) = 4p^2 - 4p^3 + p^4 \qquad (6.4.\text{relay})$$

It is readily verified that $h(p) < p$, provided that $p < 0.382$. In other words, as long as each relay isn’t totally unreliable (a relay with $p \approx 1/3$ is crummy indeed), some improvement in behavior is achieved via the bridge circuit of Fig. 6.M&S with four-fold redundancy. Recursive application of this scheme will lead to arbitrarily reliable relay circuit having the reliability function $h(h(h(\ldots h(p))))$.

The Moore-Shannon method just discussed is an example of defect circumvention via masking. The defective relays remain part of the switching circuit but their effects are counteracted by healthy relays.



(a) Redundancy scheme (b) Reliability function

Fig. 6. M&S Building reliable switching circuits from crummy relays.

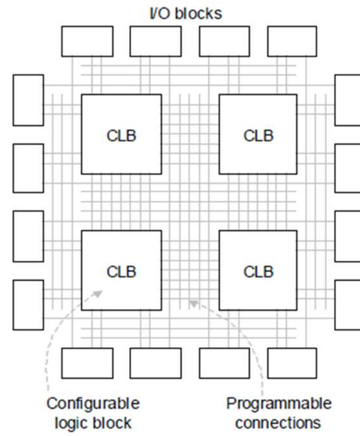
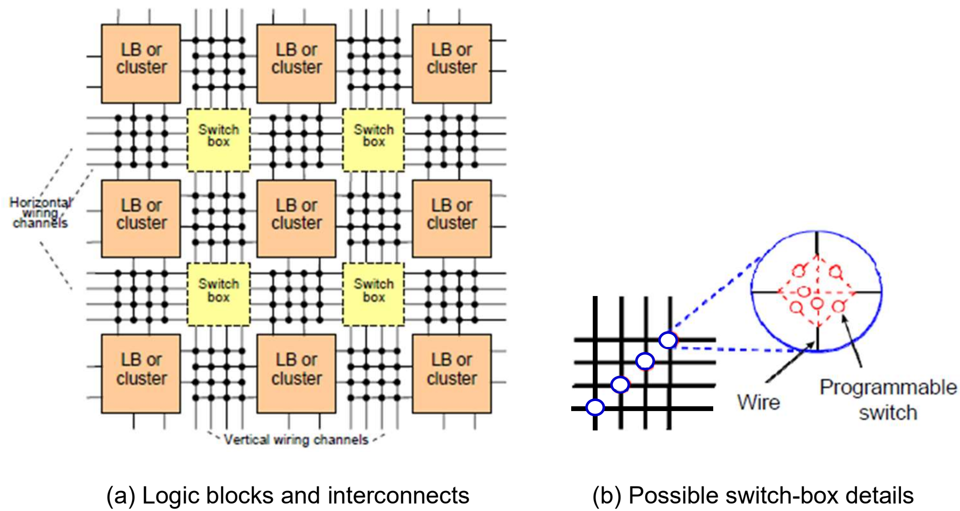


Fig. 6. FPGA1 **Bypassing of defective elements in an FPGA can be done using the same methods that allow us to avoid already used or unavailable elements.**



(a) Logic blocks and interconnects (b) Possible switch-box details

Fig. 6.FPGA2 **Routing resources in an FPGA.**

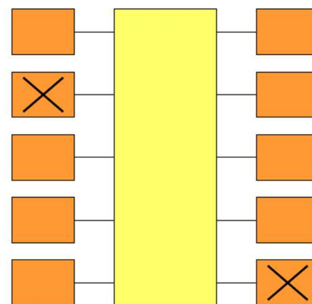


Fig. 6. CMP **Defective processors or memory modules can be disabled or bypassed in a multicore chip or chip-multiprocessor.**

FPGA and FPGA-like devices are particularly suitable for defect circumvention methods via removal (bypassing). As shown in simplified form in Fig. 6.FPGA1, an FPGA consists of an array of configurable logic blocks (CLBs) that have programmable interconnects among themselves and with special I/O blocks at the chip boundaries. The programmable interconnects, or routing resources, can take on different forms in FPGAs, with an example depicted in Fig. 6.FPGA2. Defect circumvention in such devices is quite natural because it relies on the same mechanisms that are used for layout constraints (e.g., use only blocks in the upper-left quadrant) or for blocks and routing resources that are no longer available due to prior assignment.

FPGAs are examples of circuits that are composed of multiple identical parts that are interchangeable. Similar methods are applicable to multicore chips and chip-multiprocessors. In the latter systems, processors and memory modules may be the units that are bypassed or replaced. However, defects may also impact the interconnection network connecting the processors with each other, or linking processors with memory modules. Such networks constitute the main defect circumvention challenge in this case. We will discuss the switching and reconfiguration aspects of such systems when we get to the malfunction level in our multilevel model.

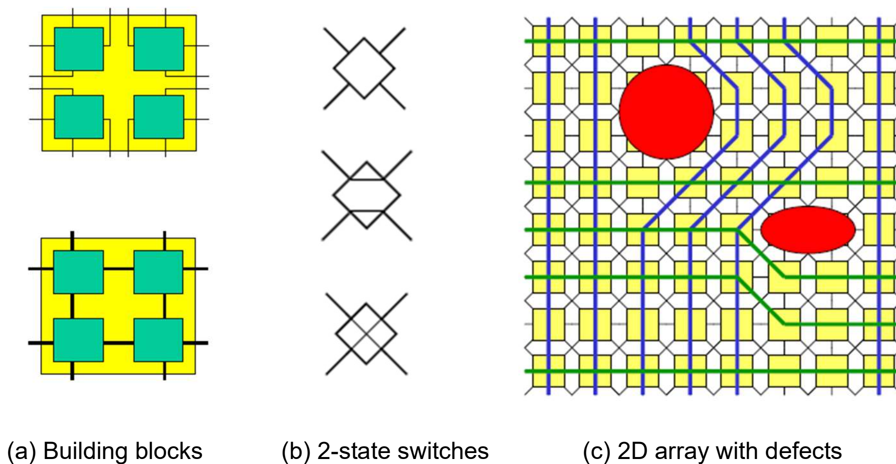
Example 6.3: Defect modeling for FPGAs Statement to be provided..

Solution: To be provided.

6.5 Defective 1D and 2D Arrays

Arrays can be built from identical nodes, of which several can be placed on a single chip. If such nodes are independent of each other and have separate I/O connections, then it would be an easy matter to avoid the use of an defective nodes. For example, to build a massively parallel processor out of 64-processor chips, we might place 72 processors on each chip to allow for up to 8 defective processors. We often prefer, however, to interconnects the nodes on the chip for higher-bandwidth communication, both on-chip and off-chip. As shown in Fig. 6.defarray-a, use of on-chip connections can lead to shorter and more efficient links, while also allowing more pins for each off-chip channel.

It is also possible to embed reconfiguration switches between nodes on a chip so as to allow dynamic bypassing of bad nodes. A simple example of such switches, a 2 × 2 switch having the 'bent' and 'crossed' states, is depicted in Fig. 6.defarray-b. As shown in Fig. 6.defarray-c, defects that make one or more nodes unusable can be circumvented by the proper setting of reconfiguration switches so as to form complete rows and columns. This method of salvaging of a smaller working array from a larger initial array is useful for both VLSI yield enhancement and run-time reconfiguration upon the detection of malfunctioning nodes. The proposed methods differ in the types and placement of switches (e.g., 4-port, single- or double-track), types and placement of spare nodes, algorithms for deriving working configurations, ways of affecting reconfiguration, and methods of assessing resilience.



(a) Building blocks (b) 2-state switches (c) 2D array with defects

Fig. 6.defarray Possible building blocks for arrays, 2-state reconfiguration switches with 'bent' and 'crossed' states, and reconfigured rows and columns in a defective 2D array.

In the following, we assume 4-port, 2-state switches depicted in Fig. 6.defarray-b. For example, a 1D array can be constructed from such switches and a set of functional and spare nodes, as shown in Fig. 6.array1D-a. Alternatively, we can embed mux-switches in each of the blocks so as to select one of two inputs (from the block immediately to the left or from the block two positions to the left) and ignoring the other input, based on diagnostic information. Such embedded switches remove single points of failures that are associated with the nonredundant switches of Fig. 6.array1D-a and also simplify the reliability modeling process.

The same reconfiguration schemes used for 1D arrays can be applied to 2D mesh arrays, as depicted in Fig. 6.arr2D1, with the switches allowing a node to be avoided by moving to a different row or column.

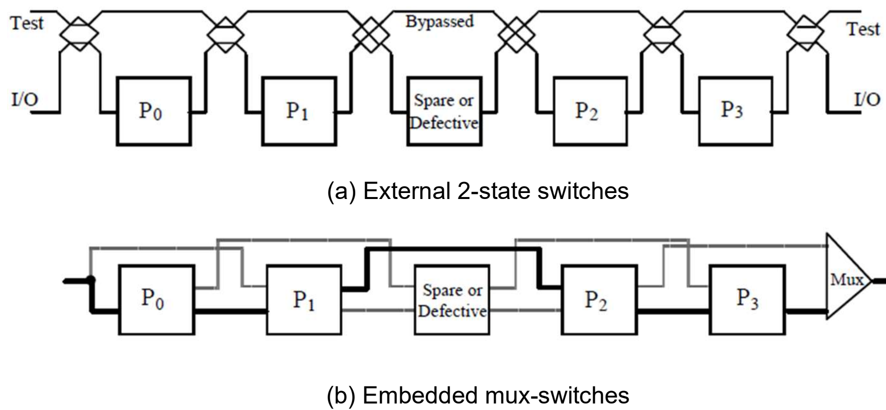


Fig. 6.array1D Reconfigurable arrays with a track of external 2-state switches and with embedded switching.

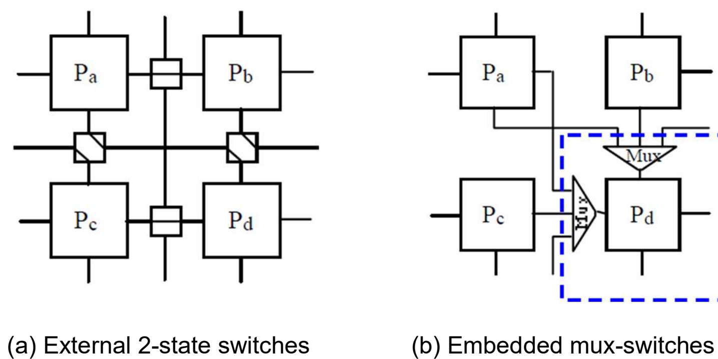
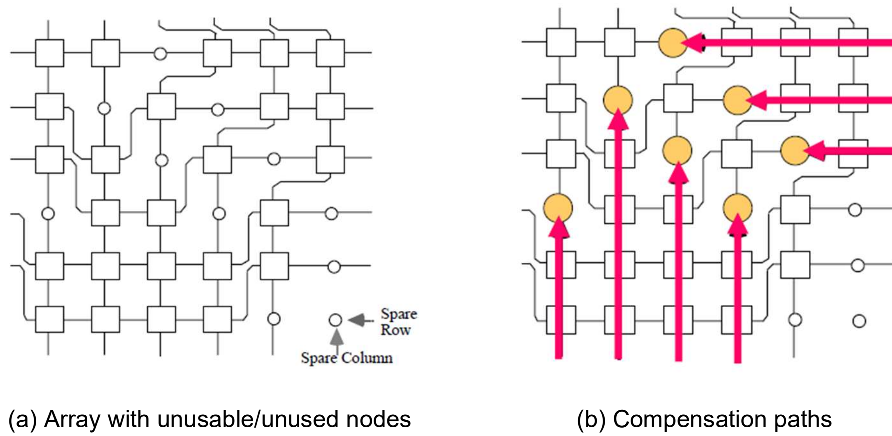


Fig. 6.arr2D1 Two types of reconfiguration switching for 2D arrays.

Assuming that we also have the capability to bypass nodes within their own rows and columns (e.g., via a separate switching scheme not shown in Fig. 6.arr2D1), we can salvage a smaller working array from one with spare rows and/or columns, as depicted in Fig. 6.arr2D2-a. The heavy arrows in Fig. 6.arr2D2-b denote how rows and columns have shifted downward or rightward to avoid the bad nodes. We will discuss both the reconfiguration capacity and the reliability modeling of such schemes in Section 8.5 in connection with yield enhancement methods.



(a) Array with unusable/unused nodes (b) Compensation paths

Fig. 6.arr2D2 Salvaging a 5 × 5 working array from a 6 × 6 one with a spare row/column, and compensation paths associated with the 7 defective nodes shown.

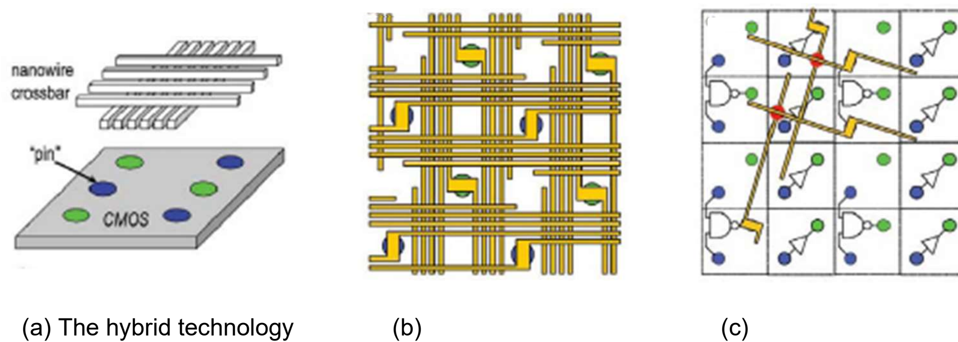
Example 6.4: Reliability modeling for processor arrays To be provided based on [Parh19].

Solution: To be provided.

6.6 Other Circumvention Methods

The notion of "crummy" components, that occupied Moore and Shannon because of unreliable electromechanical relays, is once again front and center as we enter the age of nonoelectronics. The sheer density of nanoelectronic circuits makes precise manufacturing almost impossible and the effects of even minor process variations quite serious. It is, therefore, necessary to incorporate defect circumvention methods into the design process and the structure of such circuits.

For example, hybrid-technology FPGAs, with CMOS logic elements and very compact but unreliable crossbar nanoswitches, need defect circumvention schemes [Robi07] to be deemed practical. Such hybrid schemes, as depicted in Fig. 6.nano, are expected to produce an increase of 8-fold or more in density, while providing reliable operation via defect circumvention. As another example, the use of memory architectures with block-level redundancy has been proposed for hybrid semiconductor/nanodevice implementation [Stru05]. The scheme uses error-correcting codes for defect tolerance, as opposed to using them to overcome damage from operational or "soft" errors. A possible structure is depicted in Fig. 6.memory.



(a) The hybrid technology (b) (c)

Fig. 6.nano **Nanoelectronics with "crummy" components, used in conjunction with CMOS logic, to provide significant improvement in circuit density [Robi07].**

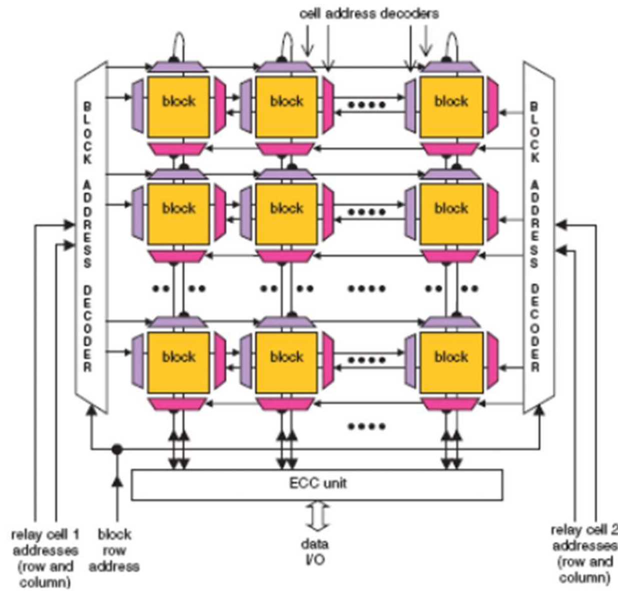


Fig. 6.memory Memory with block-level redundancy for efficient hybrid semiconductor/nanodevice implementation [Stru05].

Problems

6.1 Circuit defect circumvention by masking

Consider that in Fig. 6.M&S, each of the relays is replaced by a resistor of resistance R. The four resistors will then act like a single resistor of equivalent resistance R. Under what conditions would you say the redundant resistor is tolerant of one of the four resistors developing a defect that makes it open (disconnecting its two ends) or causes it to short-circuit?

6.2 Crummy relays

Consider the scheme for building highly reliable relay circuits, as discussed at the beginning of Section 6.4.

a. Fully justify equation (6.4.relay).

b. Determine the number of levels that the method must be applied recursively if we want to get from a relay reliability of $1 - p = 0.8$ to a switching circuit reliability of 0.9999.

c. Given relays of reliability $1 - p = 0.9$, how many relays do we need in a circuit to achieve a reliability goal of $1 - 10^{-9}$?

d. Develop an approximate formula for the number of recursive levels required if our relays have the reliability parameter $1 - p \approx 1 - \varepsilon$ (where ε is quite small) and we have a reliability goal $r > 1 - p$.

e. How good is the approximation of part d when applied to the examples of parts c and b? Discuss.

6.3 Crummy relays

a. Analyze the switching circuit of Fig. 6.M&S-a, after removing the middle vertical connection between the upper and lower relays.

b. Is the resulting circuit better or worse than the one in Fig. 6.M&S-a? Discuss.

6.4 Title

Intro

a. x

b. x

c. x

d. x

6.5 Title

Intro

a. x

b. x

c. x

d. x

6.6 Title

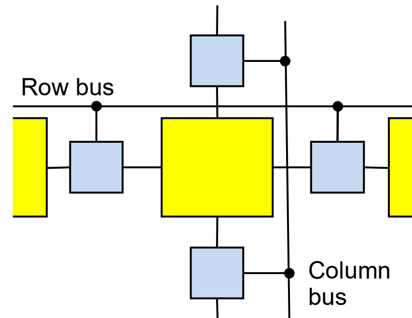
Intro

a. x

b. x

6.7 Reconfigurable 2D arrays

The following diagram represents a reconfigurable 2D array with embedded three-terminal switches, so that each switch (shown as a small box) has two connections to neighboring cells and one connection to a row/column bus that links all horizontally or vertically aligned switches together. Propose a suitable design for the switches so that reconfiguration around defective cells becomes possible in a manner similar to the scheme discussed in Section 6.5. Justify your design choices and show the switch settings for an example 2D array with defects that you specify.



6.8 Reconfigurable 2D arrays

We need 8 × 8 meshes of processors for a particular application. We want the manufactured arrays to be able to circumvent any pattern of 5 defective processors. What would you suggest in terms of providing redundant rows and columns for the array? Please provide complete reasoning for your proposal.

6.9 Reliability inversion

Read the paper [Parh19], where the notion of reliability inversion is defined and an example of where it might occur is provided. Define and analyze a second example that demonstrates reliability inversion.

6.10 Reliability modeling of reconfigurable linear arrays

Using the modeling approach of [Parh19], set up appropriate reliability models for the processor arrays depicted in Figs. 6.array1Da and 6.array1Db and compare the two schemes with reasonable assumptions about the model parameters.

References and Further Readings

[Barl65] Barlow, R. E., F. Proschan, and L. C. Hunter, *Mathematical Theory of Reliability*, 1965, pp. 199-204. (Republished by SIAM, 1996.)

[Breu04] Breuer, M. A., S. K. Gupta, and T. M. Mak, “Defect and Error Tolerance in the Presence of Massive Numbers of Defects,” *IEEE Design & Test of Computers*, Vol. 21, No. 3, pp. 216-227, May-June 2004.

[Durb04] Durbeck, L. J. K. and N. J. Macias, “Obtaining Quadrillion-Transistor Logic Systems Despite Imperfect Manufacture, Hardware Failure, and Incomplete System Specification,” Chapter 4 in *Nano, Quantum and Molecular Computing: Implications to High Level Design and Validation*, S. K. Shukla and R. I. Bahar (eds.), Kluwer, 2004, pp. 109-132.

[Grah04] Graham, P. and M. Kokhale, “Nanocomputing in the Presence of Defects and Faults: A Survey,” Chapter 2 in *Nano, Quantum and Molecular Computing: Implications to High Level Design and Validation*, S. K. Shukla and R. I. Bahar (eds.), Kluwer, 2004, pp. 39-72.

[Koch59] Kochen, M., “Extension of Moore-Shannon Model for Relay Circuits,” *IBM J.*, April 1959, pp. 169-186.

[Kore98] Koren, I. and Z. Koren, “Defect-Tolerant VLSI Circuits: Techniques and Yield Analysis,” *Proc. IEEE*, Vol. 86, No. 9, pp. 1817-1836, September 1998.

[Kore07] Koren, I., and C. M. Krishna, *Fault-Tolerant Systems*, Morgan Kaufmann, 2007.

[Mish04] Mishra, M. and S. C. Goldstein, “Defect Tolerance at the End of the Roadmap,” Chapter 3 in *Nano, Quantum and Molecular Computing: Implications to High Level Design and Validation*, S. K. Shukla and R. I. Bahar (eds.), Kluwer, 2004, pp. 73-108.

[Moor56] Moore, E. F. and C. E. Shannon, “Reliable Circuits Using Less Reliable Relays” (Parts I & II), *J. Franklin Institute*, Vol. 262, No. 3, pp. 191-208, September 1956, and No. 4, pp. 281-297, October 1956.

[Parh19] Parhami, B., “Reliability Inversion: A Cautionary Tale,” submitted for publication.

[Robi07] Robinett, W., G. S. Snider, P. J. Kuekes, and R. S. Williams, “Computing with a Trillion Crummy Components,” *Communications of the ACM*, Vo. 50, No. 9, pp. 35-39, September 2007.

[Stru05] Strukov, D. B. and K. K. Likharev, “Prospects for Terabit-Scale Nanoelectronic Memories,” *Nanotechnology*, Vol. 16, pp. 137-148, January 2005.

[Tref15] Trefzer, M. A., J. A. Walker, S. J. Bale, and A. M. Tyrrell, “Fighting Stochastic Variability in D-Type Flip-Flop with Transistor-Level Reconfiguration,” *IET Computers & Digital Techniques,* Vol. 9, No. 4, pp. 190-196, July 2015.

7 Shielding and Hardening

"Should you shield the canyons from the windstorms you would never see the true beauty of their carvings."

Elisabeth Kubler-Ross

"Go on. Nothing that you can say can distress me now. I am hardened."

E. M. Forester, "The Machine Stops", in The Eternal Moment (Collection of Short Stories), Harcourt Brace, 1928

Topics in This Chapter

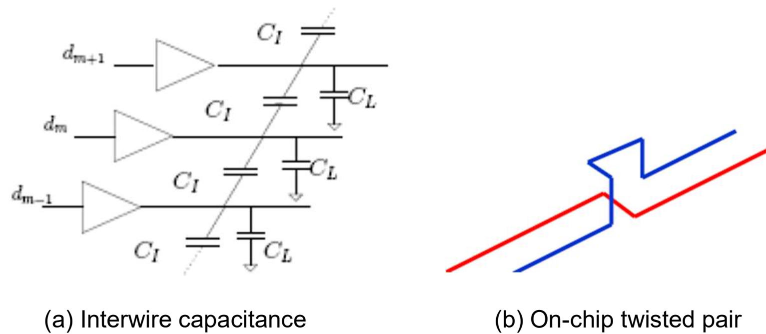
7.1. Interference and Cross-Talk
7.2. Shielding via Enclosures
7.3. The Radiation Problem
7.4. Radiation Hardening
7.5. Vibrations, Shocks, and Spills
7.6. Current Practice and Trends

Shielding is the act of isolating a part or subsystem from the external world, or from other parts or subsystems, with the goal of preventing defects that are caused or aggravated by external influences. This approach, which has been used for decades to protect systems that operate in harsh environments, is now necessary for run-of-the-mill digital systems, given the continually rising operating frequencies and susceptibility of nanoscale components to electromagnetic interference and particle impacts. As effective as shielding can be, it is often not enough. Hardening is the complementary technique of increasing the resilience of components with regard to the undesirable effects named above.

7.1 Interference and Cross-Talk

Electromagnetic or radio-frequency interference (EMI, RFI) is a disturbance that affects an electrical circuit owing to either electromagnetic conduction or electromagnetic radiation emitted from an external source. The disturbance may interrupt, obstruct, or otherwise degrade or limit the effective performance of the circuit. Interference can occur through the air or via shared power supply. Crosstalk (XT) refers to any phenomenon by which a signal transmitted on one circuit or channel of a transmission system creates an undesired effect in another circuit or channel. Crosstalk is usually caused by undesired capacitive, inductive, or conductive coupling from one circuit, part of a circuit, or channel, to another.

Shrinking feature sizes have made on-chip crosstalk a major problem. Increased clock frequency is also an important contributing factor. At very high frequencies, the small, distributed capacitance that exists between mutually insulated circuit nodes may lead to an effective short to the ground, weakening the signals and affecting their ability to perform the intended functions. Referring to Fig. 7.1a, the interwire capacitance C_I can easily exceed the load plus parasitic capacitance C_L for long buses, affecting power dissipation, speed, and signal integrity.



(a) Interwire capacitance (b) On-chip twisted pair

Fig. 7.1 Source of crosstalk problems, and a mitigation method.

7.2 Shielding via Enclosures

Materials and techniques exist for shielding hardware from a variety of external influences such as static electricity, electromagnetic interference, or radiation. Many advanced shielding methods have been developed for use with spacecraft computers that may be subjected to extreme temperatures and other harsh environmental conditions. Noteworthy among adverse conditions affecting electronic systems in space is bombardment by high energy atomic particles.

As VLSI circuit features shrink, the radiation problem, formerly problematic only during space missions, affects the proper operation of electronics even on earth. We will discuss methods for dealing with the radiation problem in Sections 7.3 and 7.4.

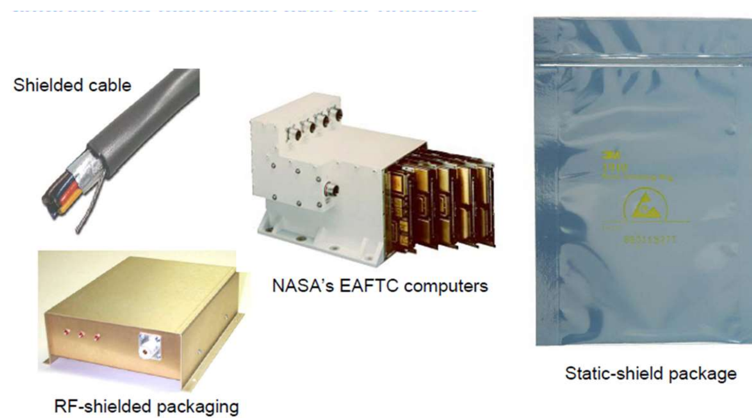


Fig. 7.encl Shielding via specialized enclosures or packaging.

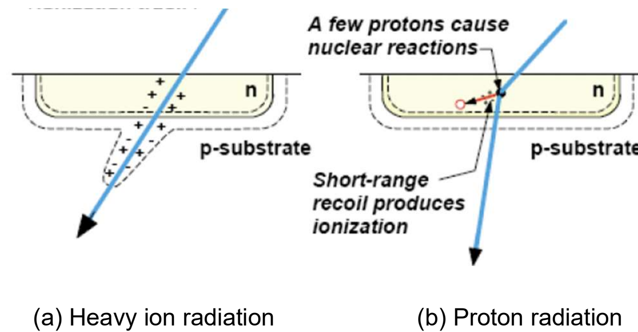
7.3 The Radiation Problem

Computing and other electronic equipment can be affected by radiation of two kinds: electromagnetic and particle.

Effects of electromagnetic radiation can be easily eradicated. Ultraviolet (UV) radiation is nonpenetrating and thus easily stopped. Both X-ray and gamma radiations can be absorbed by atoms with heavy nuclei, such as lead. Other defensive measures include the use of thick layer of suitably reinforced concrete, as used, for example, in building nuclear reactors.

Particle radiation comes in a variety of forms. Alpha particles (helium nuclei) are the least penetrating so that even paper stops them. Beta particles (electrons) are somewhat more penetrating, requiring the use of aluminium sheets. Neutron radiation is more diffictult to deal with, requiring rather bulky shielding. Finally cosmic radiation comes into play for space electronics. Besides primary radiation of the kinds just cited, secondary radiation, arising from the interaction of primary radiation and material used for shielding, is also of some interest.

As integrated circuits shrink in size, the damage done by high-energy particles, such as protons or heavy ions, can be significant. Radiation ionizes the oxide, creating electrons and holes; the electrons then flow out, creating a positive charge which leads to current leak across the channel. It also decreases the threshold voltage, which affects timing and other operational parameters. It has been estimated that a one-way mission to Mars exposes the electronics to about 1000 kilorad of radiation in total, which is near the limit of what is now tolerable by advanced space electronics.



(a) Heavy ion radiation (b) Proton radiation

Fig. 7.rad Effects of heavy-ion and proton radiations on electronics. [Source: http://parts.jpl.nasa.gov/docs/Radcrs_Final.pdf]

The most common negative impacts of radiation, and the associated terminology, are as follows:

Single-event upset (SEU): A single ion changing the state of a memory or register bit; multiple bits being affected is possible, but rare.

Single-event latchup (SEL) or snapback: A heavy ion or a high-energy particle shorting the power source to substrate (high currents may result).

Single-event transient (SET): The discharge of collected charge from an ionization event creating a spurious signal.

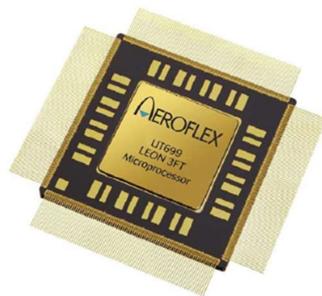
Single-event induced burnout (SEB): A drain-source voltage exceeding the breakdown threshold of the parasitic structures.

Single-event gate rupture (SEGR): A heavy ion hitting the gate region, combined with applied high voltage, as in EEPROMs, creates breakdown.

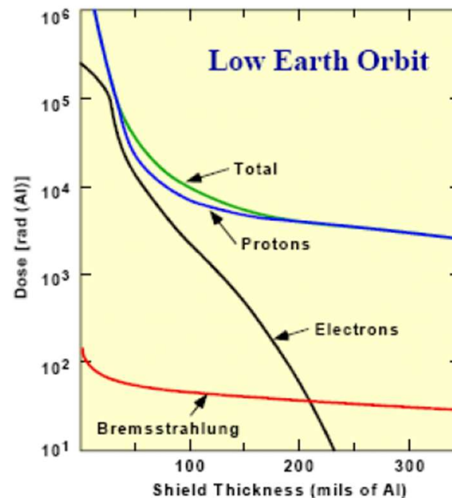
7.4 Radiation Hardening

Radiation hardening is accomplished by a variety of methods, applied from the device and circuit levels all the way to system level. At the device and component level, four approaches are noteworthy. First, instead of common, and fairly inexpensive, semiconductor substrate, an insulating or wide-band substrate may be used. Second, sensitive parts may be replaced with more rugged, functionally equivalent components. For example SRAMs may be used instead of DRAMs, a strategy that is quite effective but implies a nontrivial added cost. Third, the chip or package containing the circuit may be shielded through the use of more resilient material in the chip's composition. Fourth, the packaging ma be made radioactive-resistant, an approach that is not as effective against proton radiation as other kinds of radiation, but the ability of the packaging to slow down the particles, if not completely stop them, may be valuable when used in conjuction with other methods.

At the fault level, circuit duplication/triplication, along with comparison/voting, can be used to guard against radiation-caused deviations. One level further up, we can use error codes to detect or correct any incorrect value produced. Finally, at the system and application levels, a number of strategies, including on-line or periodic testing, liveness checks, and frequent system resets can help guard against radiation-caused problems.



(a) Packaging example



(b) Packaging ffectiveness

Fig. 7.pack Packaging and its effectiveness against radiation. [Source: http://parts.jpl.nasa.gov/docs/Radcrs_Final.pdf]

One important point to keep in mind about enclosures used to mitigate radiation effects is that proper care must be taken about the choice of material. Because of the possibility of secondary radiation (radiation of a different kind produced as a result of the primary radiation interacting with the packaging material), improper packaging may actually do more harm than good in protecting against radiation effects.

7.5 Vibrations, Shocks, and Spills

Besides radiation, a variety of other envioronmental conditions can affect the proper functioning of computer equipment. Vibrations, shocks, and spills constitute some of the major categories of such conditions.

Vibration can be a problem when a computer system is installed in a car, truck, train, boat, airplane, or space capsule (basically, anything that moves or spins). Certain factory or process-control installations are also prone to excessive vibrations that may cause loose connections to become undone or various mechanical parts to break down from stress. Systems can be ruggedized to tolerate vibration by initial stress testing (screening out products that are prone to fail when exposed to vibration) and use of special casing that absorbs or neutralizes the unwanted movements.

Shock is experienced from rough handling of devices or exposure to impact, as in an accidental drop or car crash. Various levels of protection against shock can be provided, ranging from special “skins” added to ordinary devices (as in phone or tablet cases/shells) to total product redesign from specs. Many modern portable devices have built-in sensors that detect the acceleration resulting from an impact or drop and initiate protective actions, such as saving the state or securing the disk.

(a) Ruggedized phone (b) Ruggedized laptop (c) Ruggedized disk drive

Fig. 7.rugged Ruggedized phone (Casio G-Shock), laptop (Panasonic Toughbook), and external disk drive (LaCie/Hitachi).

Protection against spills, and waterproofing in general, is technically quite simple, but of course adds to the product cost. Watches and cameras have been marketed in waterproof versions for many decades. The same methods can be applied to smartphones, laptops, and other electronic devices. As mechanical moving parts, bottons, and levers disappear from our devices, the task of waterproofing becomes simpler.

Laptop computers that have been partially ruggedized against shock and spills and are aimed for use by children have been in existence for several years now. Other environmental conditions against which protection may be sought include electrical noise (needed for use in some industrial environments; see Section 7.1), radiation (see Section 7.3), and heat.

7.6 Current Practice and Trends

This section has not yet been written. The following paragraphs contain some of the points to be made.

Abstract of [Nemo97]: Single-event upset (SEU) tolerance for commercial 1Mbit SRAMs, 4Mbit SRAMs, 16Mbit DRAMs and 64Mbit DRAMs was evaluated by irradiation tests using high-energy heavy ions with an LET range between 4.0 and 60.6 MeV/(mg/cm2). The threshold LET and the saturated cross-section were determined for each device from the LET dependence of the SEU cross-section. We show these test results and describe the SEU tolerance of highly integrated memory devices in connection with their structures and fabrication processes. The SEU rates in actual space were also calculated for these devices.

Abstract of [Karn04]: Radiation-induced single event upsets (SEUs) pose a major challenge for the design of memories and logic circuits in high-performance microprocessors in technologies beyond 90nm. Historically, we have considered power-performance-area trade offs. There is a need to include the soft error rate (SER) as another design parameter. In this paper, we present radiation particle interactions with silicon, charge collection effects, soft errors, and their effect on VLSI circuits. We also discuss the impact of SEUs on system reliability. We describe an accelerated measurement of SERs using a high-intensity neutron beam, the characterization of SERs in sequential logic cells, and technology scaling trends. Finally, some directions for future research are given.

Abstract of [Worm05]: Systems-on-Chip (SoC) design involves several challenges, stemming from the extreme miniaturization of the physical features and from the large number of devices and wires on a chip. Since most SoCs are used within embedded systems, specific concerns are increasingly related to correct, reliable, and robust operation. We believe that in the future most SoCs will be assembled by using large-scale macro-cells and interconnected by means of on-chip networks. We examine some physical properties of on-chip interconnect busses, with the goal of achieving fast, reliable, and low-energy communication. These objectives are reached by dynamically scaling down the voltage swing, while ensuring data integrity-in spite of the decreased signal to noise ratio-by means of encoding and retransmission schemes. In particular, we describe a closed-loop voltage swing controller that samples the error retransmission rate

to determine the operational voltage swing. We present a control policy which achieves our goals with minimal complexity; such simplicity is demonstrated by implementing the policy in a synthesizable controller. Such a controller is an embodiment of a self-calibrating circuit that compensates for significant manufacturing parameter deviations and environmental variations. Experimental results show that energy savings amount up to 42%, while at the same time meeting performance requirements.

Problems

7.1 Designs with improved noise immunity

As devices and interconnects are scaled down, integrated-circuits become more vulnerable to noise. Many techniques have been proposed for reducing the vulnerability (increasing the immunity) to noise in such circuits. Study the twin-transistor method to improve noise tolerance [Bala01] and write a 2-page report about the essence of the method and the domain of its applicability.

7.2 Effects of radiation on logic circuits

Read the paper [Poli11] and answer the following qustions.

a. How does radiation strike affect the output of a NAND gate with both inputs being 1?

b. Discuss the effect of radiation strike on the operation of a bit-serial adder.

c. Repeat part b for a ripple-carry adder.

d. Which of the adders of parts b and c is more likely to produce an erroneous sum due to radiation?

7.3 Selective hardening

Read the paper [Poli11a] and summarize its key ideas in one typset page. Single-space the text and include only one figure from the paper that you deem most important to conveying its main message.

7.4 Wave attacks

In addition to protecting computer systems against radiation and other electromagnetic waves caused by specific environments and random phenomena in them, we should also be concerned with malicious attacks that take advantage of system sensitivities to such external interference to force crashes or to compromise security and data privacy. Study the latter problem and prepare a single-spaced 2-page report on the range of threats and possible remedies.

7.4 Rugged laptops for space applications

Many technological advances have their roots in the safety and ruggedness requirements of space flight. The GRiD (Graphical Retrieval Information Display) Compass, the first laptop in orbit, had a 21.6-cm bright plasma display and was used by NASA on Space Shuttle missions through the early 1990s. The rugged 4.5-kg laptop, costing $8150 at the time, reportedly survived the 1986 Space Shuttle Challenger crash. Use Internet sources to discover the various design methods used to build the GRiD Compass and present your findings in a 2-page report.

References and Further Readings

[Bala01] Balamurugan, G. and N. R. Shanbhag, "The Twin-Transistor Noise-Tolerant Dynamic Circuit Technique," *IEEE J. Solid-State Circuits*, Vol. 36, No. 2, pp. 273-280, February 2001.

[Carm99] Carmichael, C., E. Fuller, P. Blain, and M. Caffrey, "SEU Mitigation Techniques for Virtex FPGAs in Space Applications," *Proc. MAPLD Int'l Conf.*, 1999.

[Clar16] Clark, L. T., D. W. Patterson, C. Ramamurthy, and K. E. Holbert, "An Embedded Microprocessor Radiation Hardened by Microarchitecture and Circuits," *IEEE Trans. Computers*, Vol. 65, No. 2, pp. 382-395, February 2016

[Duan09] Duan, C., V. Cordero, and S. P. Khatri, "Efficient On-Chip Crosstalk Avoidance CODEC Design," *IEEE Trans. VLSI Systems*, Vol. 17, No. 4, pp. 551-560, April 2009.

[Edmo00] Edmonds, L. D., C. E. Barnes, and L. Z. Scheick, "An Introduction to Space Radiation Effects on Microelectronics," JPL Publication 00-06, 83 pp., May 2000. [Available on-line at: https://snebulos.mit.edu/projects/reference/NASA-Generic/JPL-00-06.pdf]

[Gatt16] Gatti, U., C. Calligaro, E. Pikhay, and Y. Roizin, "Radiation-Hardening Methodologies for Flash ADC," *Analog Integrated Circuits and Signal Processing*, Vol. 87, No. 2, pp. 141-154, May 2016.

[Geet09] Geetha, S., K. K. Satheesh Kumar, C. R. K. Rao, M. Vijayan, and D. C. Trivedi, "EMI Shielding: Methods and Materials—A Review," *J. Applied Polymer Science*, Vol. 112, No. 4, pp. 2073-2086, 2009.

[John98] Johnston, A. H., "Radiation Effects in Advanced Microelectronics Technologies," *IEEE Trans. Nuclear Science*, Vol. 45, No. 3, pp. 1339-1354, June 1998.

[JPL] Jet Propulsion Lab., NASA, "Space Radiation Effects on Microelectronics," Short course slides, undated, available online at: http://parts.jpl.nasa.gov/docs/Radcrs_Final.pdf

[Karn04] Karnik, T., "Characterization of Soft Errors Caused by Single Event Upsets in CMOS Processes," *IEEE Trans. Dependable and Secure Computing*, Vol. 1, No. 2, pp. 128-143, April-June 2004.

[Kern88] Kerns, S. E., *et al.*, "The Design of Radiation-Hardened ICs for Space: A Compendium of Approaches," *Proc. IEEE*, Vol. 76, No. 11, pp. 1470-1509, November 1988.

[Khat01] Khatri, S. P., R. K. Brayton, and A. L. Sangiovanni-Vincentelli, *Cross-Talk Noise Immune VLSI Design Using Regular Layout Fabrics*, Kluwer, 2001.

[Ma89] Ma, T. P. and P. V. Dressendorfer, *Ionizing Radiation Effects in MOS Devices and Circuits*, Wiley, 1989.

[Nemo97] Nemoto, N., et al., "Evaluation of Single-Event Upset Tolerance on Recent Commercial Memory ICs" *Proc. 3rd ESA Electronic Components Conf.*, April 1997.

[Poli11] Polian, I., J. P. Hayes, S. M. Reddy, and B. Becker, "Modeling and Mitigating Transient Errors in Logic Circuits," *IEEE Trans. Dependable and Secure Computing*, Vol. 8, No. 4, pp. 537-547, July/August 2011.

[Poli11a] Polian, I. and J. P. Hayes, "Selective Hardening: Toward Cost-Effective Error Tolerance," *IEEE Design & Test of Computers*, Vol. 28, No. 3, pp. 54-63, May-June 2011.

[Worm05] Worm, F., P. Ienne, P. Thiran, and G. De Micheli, "A Robust Self-Calibrating Transmission Scheme for on-Chip Networks," *IEEE Trans. VLSI Systems*, Vol. 13, No. 1, pp. 126-139, January 2005.

[Yu09] Yu, H., L. He, and M.-C. F. Chang, "Robust On-Chip Signaling by Staggered and Twisted Bundle," *IEEE Design & Test of Computers*, Vol. 26, No. 5, pp. 92-104, September/October 2009.

8 Yield Enhancement

"As the soft yield of water cleaves obstinate stone, so to yield with life solves the insolvable: To yield, I have learned, is to come back again."

Lao-Tzu

". . . never give in except to convictions of honour and good sense. Never yield to force; never yield to the apparently overwhelming might of the enemy."

Winston Churchill

Topics in This Chapter

8.1. Yield Models
8.2. Redundancy for Yield Enhancement
8.3. Floor-Planning and Routing
8.4. Improving Memory Yield
8.5. Regular Processor Arrays
8.6. Impact of Process Variations

In Section 5.2, we introduced the notion of yield and explained why a small deterioration in defect density is amplified in the way it affects the final product cost. Despite significant strides in improving the design and manufacturing processes for integrated circuits, yield has presented a greater challenge with each generation of denser and more complex devices. Due to the significant impact of yield on cost, modern production technologies for electronic devices incorporate provisions for detecting and circumventing defects of various kinds to reduce the need for discarding slightly defective parts. In this chapter we review defect detection and circumvention methods that are particularly suitable for the goal of yield enhancement in digital circuits and storage products.

8.1 Yield Models

Yield models are combinatorial in nature and range from primitive to highly sophisticated. Let us begin with a very simple example.

Example 8.1: Modeling of yield Consider a square chip area of side 1 cm filled with parallel, equally spaced nodes with width and separation of 1 μm. Assume there are an average of 10 random defects per cm^2. Defects are of the extra-material kind, with 80% being small defects of diameter 0.5 μm and 20% being larger defects of diameter 1.5 μm. What is the expected yield of this simple chip?

Solution: The expected number of defects on the chip is 10 (8 small, 2 large). Small defects cannot lead to shorts, so we can ignore them. A large defect leads to a short if its center is within a 0.5-μm band halfway between two neighboring nodes. So we need to compute the probability of at least 1 large defect appearing within a critical area of 0.25 cm^2, given an average of 2 such defects in 1 cm^2. Because each of the 2 defects falls in the critical area with probability 1/4, the probability of having at least 1 large defect in that area is $1 - (3/4)^2 = 7/16$, giving a yield of $9/16 \cong 56\%$.

Most yield models in practical use are based on defect distributions that provide information about the frequencies and sizes of defects of various kinds. They then take the exact circuit layout or some rough model of it into account in deriving critical areas for each defect type/size. The ratio of the critical area to the total area is a measure of the sensitivity of the particular layout to the corresponding defect type/size.

8.2 Redundancy for Yield Enhancement

Consider a device consisting of n identical cells. A simple-minded strategy for yield improvement is to provide s spare cells so that we still have at least n good cells in the presence of up to s defective ones. Such an approach appears to lend itself to modeling as an n-out-of-$(n + s)$ system. However, such a model would be appropriate only if any defective cell is replaceable with any one of the spares. Placement of the cells on the chip and connectivity among them may make such an arbitrary replacement impossible. For example, replacement may have to be done in blocks (such as rows or columns), instead of single cells. Such restrictions would lead to a significant reduction in the resulting yield compared with what the n-out-of-$(n + s)$ model predicts.

8.3 Floor-Planning and Routing

In automated tools for electronic circuit design, floorplanning and routing stages affect the resulting yield. Thus, VLSI layout must be done with defect patterns and their impacts in mind. Designers can mitigate the effect of extra- and missing-material defects by adjusting the rules for floorplanning and routing. For example, wider wires are less sensitive to missing-material defects and narrower wires are less likely to be shorted to others by extra material, given the same center-to-center separation (Fig. 7.matdef). The examples above indicate that designers face a complex array of optimizations and trade-offs, as they must strike a balance with regard to sensitivity to different defect types.

Different chip layout/routing designs differ in their sensitivity to various defect classes. Because of defect clustering, one good idea is to place blocks with similar sensitivities to defects apart from each other.

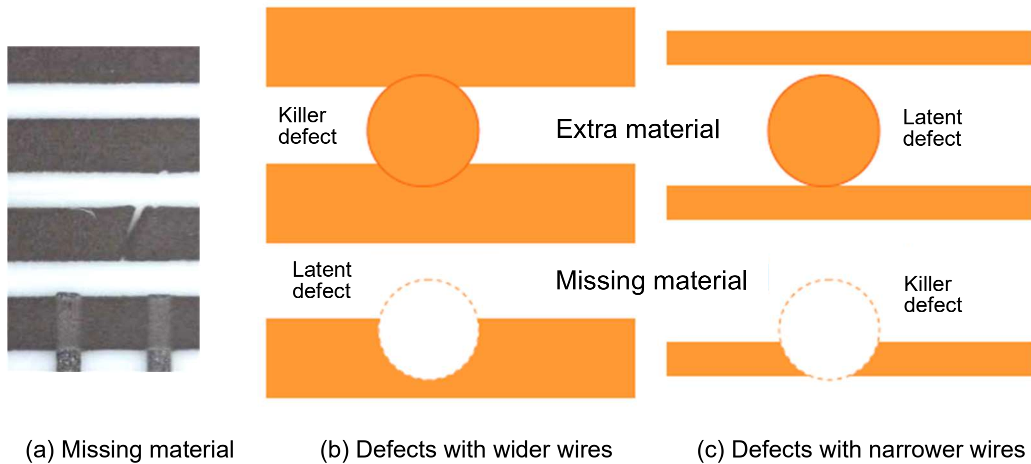


Fig. 7.matdef A real missing-material defect and simplified modeling of both extra- and missing-material defects as circles.

One approach to modeling the impact of defects on yield is to derive critical areas in the layout where the presence of a defect of a given size would disable the circuit. The gray regions in Fig. 7.critarea represents the results of such an effort for small extra-material defects of a specific diameter and larger defect of the size shown. The small defect is seen to be noncritical in most areas, causing shorts between wires/vias only in the small, fairly narrow gray regions shown. So, there is a relatively low probability that the small defects would lead to an unusable chip. The larger defect, on the other hand, can lead to shorts when centered in a significant portion of the circuit segment shown, making it a killer defect with high probability.

The fraction of the chip area that is critical with respect to various defect sizes, combined with information on the distribution of such defects (Fig. 5.8), allows us to compute the overall probability that the chip will be rendered unusable by extra-material or missing-material defects. If changes in the layout cannot improve an unreasonably low yield, then redundancy techniques, discussed elsewhere in this chapter, might be called for.

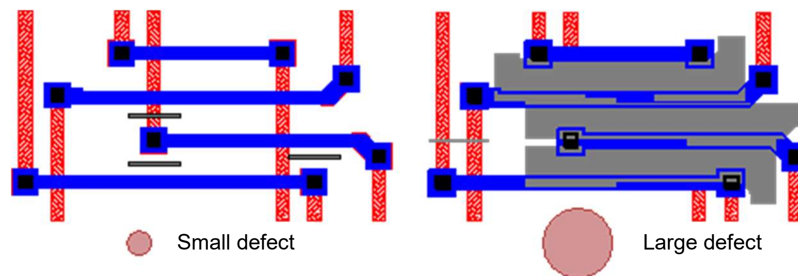


Fig. 7.critarea Critical areas for different defect sizes.

8.4 Improving Memory Yield

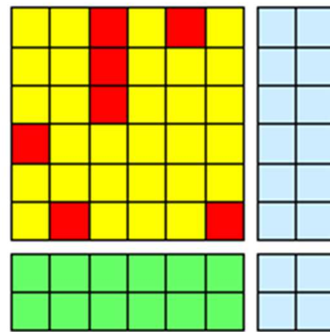
Systems-on-chips (SOCs) found in many modern electronic products consist mostly of memory, so improving memory yield is important to the overall yield.

The most common redundancy scheme applied to semiconductor memories is the provision of spare rows and/or columns. The memory array in Fig. 8.memsrc-a has been provided with 2 spare rows and 2 spare columns. In general, the number of spares need not be the same in the two dimensions and the spares need not be global in the sense of spanning all rows/columns. For example, the 6-row memory array of Fig. 8.memsrc-a may be divided into two 3-row banks, with each bank having its own pair of spare columns of length 3. Similarly, the spare rows may be segmented when the banks have fewer columns than the entire memory array.

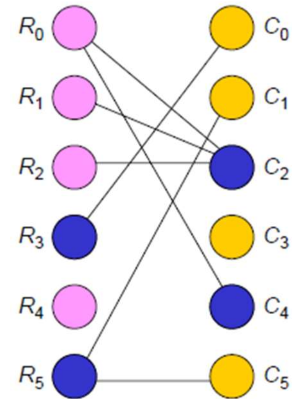
Replacement of defective rows/columns with spares reqires the incorporation of switching mechanisms around the memory array so as to route accesses to the defective entities to an assigned spare. [Elaborate on switching schemes.]

Given a particular pattern of defective memory cells (bits), such as the dark cells in Fig. memsrc-a, we would like to know if the available spare resources are adequate for circumventing the defects. In other words, we want to find out an assignment of spare rows and columns to the defective cells so that all defects are covered, if it exists, or to conclude that the circumvention capacity of the system has been exceeded. For the example set of 7 defects in Fig. memsrc-a, the assignment can be readily found by inspection: use the spare columns to cover the defects in columns 2 and 4 (numbering from 0) and the spare rows to circumvent the defects in rows 3 and 5. We may also be interested to find the optimal assignment, that is, one that used a minimal number of spare resources, in case more defects must be circumvented in future.

The assignment problem discussed in the preceding paragraph is NP-complete in general. We can conclude quite easily that any pattern of n defects can be circumvented by using n spare rows and columns in any combination (r spare rows and $n - r$ spare columns, for any r). This lower bound is also an upper bound in the worst case when no two defects are in the same row or column. In most cases, however, we can do better, circumventing significantly more than $r + c$ defects, given r spare rows and c spare columns (7 defects, with 2 spare rows and 2 spare columns in the example above).



(a) Memory array, with spare rows/columns



(b) A representation of the defect pattern

Fig. 8.memsrc Memory array with spare rows and column, and the bipartite graph representing the defect pattern shown.

The spare row/column assignment problem can be converted to a graph problem as follows. Construct a bipartite graph, with vertices on the two sides corresponding to rows (R_0-R_{r-1}) and columns (C_0-C_{c-1}) in the memory array and edges connecting pairs of vertices R_i and C_j if there is a defect in row i, column j. The assignment problem is then selecting r vertices among the R_i and c vertices among the C_j such that every edge is incident to a selected node.

8.5 Regular Processor Arrays

Processor arrays of 1 or 2 dimensions can be embedded with switches in order to circumvent a defective processor. In a 1D scheme, exemplified by Fig. 6.array1D, any number of defective processors can be circumvented by appropriate setting of the switch states. Assuming the switches and defect detection to be perfect, the resulting system can be modeled as an n-out-of-(n + s) system. Defective switches can be circumvented by either embedding them in processor cells (Fig. 6.array1D-b) or by using a redundant switching network in which defective switches can be bypassed.

The latter approach for bypassing a single defective switch is depicted in Fig. 8.red1Dsw, where we see that even though the switching cell 3′ is inoperative, communication between neighbors among the remaining processors via the switching network is not interrupted. However, the unusability of the switching cell 3′ also makes processor 2 inaccessible. The use of distributed switching, as shown in Fig. 6.array1D-b, obviates the need for considering redundancy schemes for the switches and for more complicated models with separate provisions for switching reliability.

Whereas the 1D arrays discussed above have no limit on the array size and the number of spares, the 2D array reconfiguration schemes of Fig. 6.arr2D1 are more constrained, owing to the more rigid connectivity requirements between processor nodes. Referring to Fig. 6.arr2D2-b, we note that a particular pattern of defects can be circumvented if straight, nonintersecting, nonoverlapping paths (the solid arrows) can be drawn from the spare row or column elements to each defective element. The 7 "compensation paths," shown as heavy arrows in Fig. 6.arr2D2-b, do not intersect or overlap, indicating that the 7 defective processors can be circumvented, as demonstrated in the same figure.

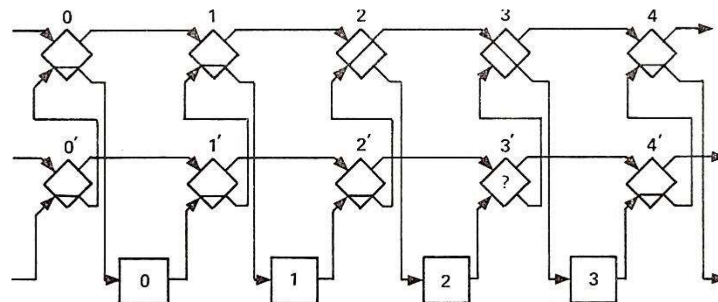


Fig. 8.red1Dsw Reconfigurable 1D array with redundant switching.

A natural question at this point is the maximum circumvention capacity of the reconfigurable 2D array in the worst case. It is easy to show that with 1 spare row and 1 spare column, no more than 3 defective processors can be circumvented in the worst case. Four defective processors, if they appear in a 2×2 block, will defeat the scheme, because one of the processors will be "behind" others as far as the spare nodes are concerned. In fact, you should be able to come up with a 3-processor defect pattern that is also noncircumventable.

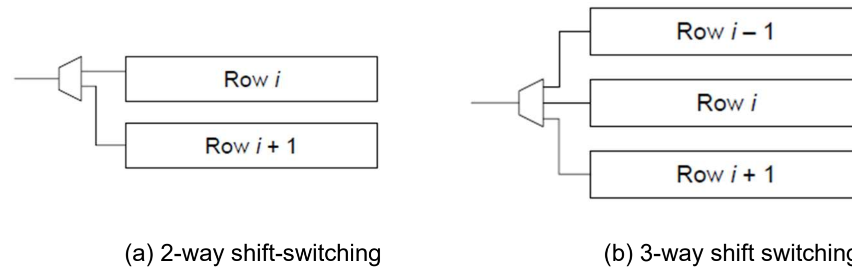
The discussion above is based on the assumption of 2-way shift-switching at the edges of the array, so that a row/column is either connected straight through or with a one-position shift downward/rightward (Fig. 8.shift-a). It is also possible to use 3-way shift-switching at the edges (Fig. 8.shift-b), which would allow the spare row/column to be viewed as being on either side of the array, depending on the defect pattern. This added flexibility would improve the worst-case noncircumventable defect pattern to 4 processors, thus improving yield.

An $m \times m$ array must then be modeled as an $(m^2 - 2)$-out-of-m system with regard to reliability or yield, assuming 2-way shift-switching at the boundaries and as an $(m^2 - 3)$-out-of-m system with 3-way shift-switching. These reliability models are simple but highly pessimistic. Precise modeling is more complicated, requiring that we enumerate all the possible defect patterns that can/cannot be circumvented and compute a propbability for each case.

Example 6.x: Reliability modeling for procressor arrays To be provided based on [Parh19].

Solution: To be provided.

We can go beyond the 3-defect limit for reconfigurable 2D arrays by providing spare rows and columns on all array boundaries, that is, spare rows at the top and bottom and spare columns on either side. Figuring out the worst-case defect pattern in this case is left as an exercise.



(a) 2-way shift-switching (b) 3-way shift switching

Fig. 6.shift Shift-switching at the edges of a reconfigurable 2D array.

As in most engineering problems, the optimal solution method for a particular application and defect model may be a mix of the various method available to us. Intutively, we can think of the effectiveness as the hybrid approach as being due to each method having some weaknesses that are covered by the other(s).

A good example of a hybrid approach is provided by the problem of memory yield enhancement. IBM's 16 Mb memory chip [ref cite] used 16 spare rows and 24 spare columns in each of its 4 quadrants, along with a single-error-correcting code that attached 9 check bits to each 128-bit data word (or 137 data word? Check the source). Furthermore, bits assigned to the same word were separated by 8 bit positions, making it less likely for a single defect to affect more than 1 bit in the same word. As shown in Fig. 6.rcs-ecc, the combination of row/column sparing and single-error-correcting code leads to a significant improvement in yield for memory devices.

The effectiveness of the hybrid approach just discussed can be explained thus. Row and column spares are very effective for large numbers of defects when they cluster along rows and columns. Use of error-correcting codes is capable of overcoming widespread random defects when each word does not have too many defective cells. The latter weakness is covered by row spares that allow us to circumvent the entire word (and several other words in the same row).

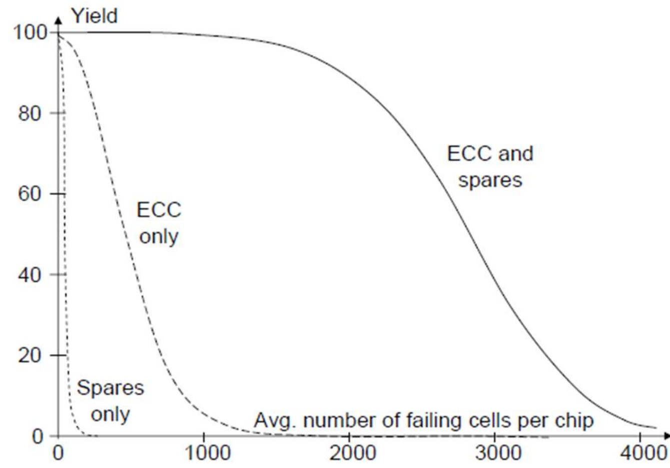


Fig. 6.rcs-ecc Memory yield as a function of the average number of defective cells with sparing and error-correcting code.

8.6 Impact of Process Variations

As devices and interconnects are scaled down, integrated-circuits become more error-prone and vulnerable to both external influences and internal interference. One important reason for such errors and vulnerabilities is manufacturing process variations [Ghos10]. Process imperfections lead to transistors, wires, and other circuit elements to have imperfect shapes, something that can be considered mild defects. When circuit elements are relatively large, small imperfections do not cause serious variations in electrical properties, such as resistance or capacitance. However, for a tiny element, a small irregularity in shape can translate to relatively large variations in electrical parameters, as well as large variations between supposedly-identical elements.

The same mechanisms that make process variations more serious in modern VLSI than in previous generations of circtuis also may lead to massive numbers of defects or new kinds of defects that have not been observed before [Breu04], [Siri04].

Example 8.y: Effect of process variations on wire resistance Consider a wire of width 100 nm on an IC chip. Process variations may cause the width of the wire along up to half of its length to become either as small as 50 nm or as large as 150 nm. Quantify the change in the wire's resistance in the worst case.

Solution: Assuming no variation in the thickness (depth) of the wire, the resistance is inversely proportional to wire width, doubling when the width is halved. In the worst case, half of the wire will have its original resistance $R/2$ and the other half will have resistance ranging from R to $R/3$. Thus, the total resistance of the wire will range from $R/2 + R = 3R/2$ to $R/2 + R/3 = 5R/6$, placing the variations relative to the original resistance R in the interval [–17%, +50%], or a factor of 1.8 separating the maximum and minimum resistance values.

Problems

8.1 Memory array with spare rows and columns

Consider an $m \times m$ memory array with r spare rows and c spare columns, where $m >> \max(r, c)$.

a. What is the smallest number of memory-cell defects that would need all of the redundant resources to overcome, that is, it cannot be circumvented if we reduce r or c by 1?

b. What is the largest number of memory cell defects that can be circumvented with the given spare rows and columns?

8.2 Memory array with spare rows and columns

We learned in Section 8.4 that the spare row/column assignment problem for memory arrays with arbitrary numbers of spare rows and columns is NP-complete.

a. The NP-completeness result applies to the general problem, not to any specific instance. Show that when there is only one spare row and one spare column, the problem can be solved efficiently.

b. Present the result of part a in the form of a spare row/column assignment algorithm and supply an argument for its correctness.

c. Derive the running time of the algorithm of part b as a function of the side length n of a square memory array.

8.3 Memory array with spare rows and columns

Consider a memory array with spare rows and/or columns for yield enhancement. Which of the following statements, if any, is correct?

a. Using a spare rows and b spare columns is preferable to using $a + b$ spare rows or columns.

b. Using $2c$ spare rows and $2c$ spare columns for a $2n \times 2n$ memory array is preferable to dividing the memory array into four quadrants and providing each of the quadrants with c spare rows and c spare columns.

8.4 Yield with defect circumvention

The manufacturing process for an integrated circuit die produces an average of 2.5 defects, with defect distribution $p(k)$, probability of having k defects, given in the following table. Approximately 40% of the defects are killer defects.

k	0	1	2	3	4	5	6
$p(k)$	0.11	0.18	0.25	0.20	0.13	0.08	0.05

a. What is the expected yield of this process?

b. We provide reconfiguration mechanisms on the die to allow the circumvention of up to two defects. Assuming that the reconfiguration logic is always free from defects and that it requires negligible area, what is the new expected yield?

8.5 Reconfigurable 2D processor arrays

We concluded that a 2D processor array with 1 spare row and 1 spare column can circumvent up to 2 defective processors in the worst case.

a. What is the guaranteed number of circumventable defects in a 2D processor array with spare rows/columns on all four sides?

b. Is it advantageous to provide more than one spare row or column on each side of the array?

c. Would increasing the defect tolerance capability change if both spare rows and spare columns are on one side of the array?

8.6 Reconfigurable 2D processor arrays

Intro

a.

b.

8.7 Critical areas for small and large defects

Consider Fig. 7.critarea, in which examples of small and large defects and their corresponding critical areas in connection with a specific circuit layout are shown. How would the criticl areas change under the following defect sizes? Only approximate answers are needed.

a. Small defects halve in diameter.

b. Small defects double in diameter.

c. Large defects halve in diameter.

d. Large defects double in diameter.

8.8 Reliability modeling of reconfigurable linear arrays

Using the modeling approach of [Parh19], set up appropriate reliability models for the processor arrays depicted in Figs. 6.array1Da and 8.red1Dsw and compare the two schemes with reasonable assumptions about the model parameters.

References and Further Readings

[Breu04] Breuer, M. A., S. K. Gupta, and T. M. Mak, “Defect and Error Tolerance in the Presence of Massive Numbers of Defects,” *IEEE Design & Test of Computers*, Vol. 21, No. 3, pp. 216-227, May-June 2004. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1302088&isnumber=28929

[Chia07] Chiang, C. C. and J. Kawa, *Design for Manufacturability and Yield for Nano-Scale CMOS*, Springer, 2007.

[Cici95] Ciciani, B., *Manufacturing Yield Evaluation of VLSI/WSI Systems*, IEEE Computer Society Press, 1995.

[Ghai09] Ghaida, R. S., K. Doniger, and P. Zarkesh-Ha, “Random Yield Prediction Based on a Stochastic Layout Sensitivity Model,” *IEEE Trans. Semiconductor Manufacturing*, Vol. 22, No. 3, pp. 329-337, August 2009.

[Ghos10] Ghosh, S. and K. Roy, “Parameter Variation Tolerance and Error Resiliency: New Design Paradigm for the Nanoscale Era,” *Proc. IEEE*, Vol. 98, No. 10, pp. 1718-1751, 2010.

[Huan03] Huang, C.-T., C.-F. Wu, J.-F. Li, and C.-W. Wu, “Built-in Redundancy Analysis for Memory Yield Improvement,” *IEEE Trans. Reliability*, Vol. 52, No. 4, pp. 386-399, December 2003.

[Kore98] Koren, I. and Z. Koren, “Defect-Tolerant VLSI Circuits: Techniques and Yield Analysis,” *Proc. IEEE*, Vol. 86, No. 9, pp. 1817-1836, September 1998.

[Kore07] Koren, I., and C. M. Krishna, *Fault-Tolerant Systems*, Morgan Kaufmann, 2007 (Chapter 8, pp. 249-283).

[Lu06] Lu, S. K., Y.-C. Tsai, C.-H. Hsu, A. Pao, K. Chiu, and E. Chen, “Efficient Built-in Redundancy Analysis for Embedded Memories with 2-D Redundancy,” *IEEE Trans. VLSI Systems*, Vol. 14, No. 1, pp. 34-42, January 2006.

[Parh19] Parhami, B., “Reliability Inversion: A Cautionary Tale,” *IEEE Computer*, Vol. 53, No. 6, pp. 28-33, June 2020.

[Parh20] Parhami, B., “Reliability and Modelability Advantages of Distributed Switching for Reconfigurable 2D Processor Arrays,” *Proc. 11th Annual IEEE Information Technology, Electronics and Mobile Communication Conf.*, November 2020, to appear.

[Royc90] Roychowdhury, V. P., J. Bruck, and T. Kailath, “Efficient Algorithms for Reconfiguration in VLSI/WSI Arrays,” *IEEE Trans. Computers*, Vol. 39, No. 4, pp. 480-489, April 1990.

[Sega00] Segal, J., L. Milor, and Y.-K. Peng, “Reducing Baseline Defect Density Through Modeling Random Defect-Limited Yield,” *IEEE Micro*, January 2000. [Available on-line at: http://micromagazine.fabtech.org/archive/00/01/segal.html]

[Siri04] Sirisantana, N., B. C. Paul, and K. Roy, “Enhancing Yield at the End of the Technology Roadmap,” *IEEE Design & Test of Computers*, Vol. 21, No. 6, pp. 563-571, November-December 2004.

[Stru05] Strukov, D. B. and K. K. Likharev, “Prospects for Terabit-Scale Nanoelectronic Memories,” *Nanotechnology*, Vol. 16, No. 1, pp. 137-148, January 2005.

[Zhan95] Zhang, J. C. and M. A. Styblinski, *Yield and Variability Optimization of Integrated Circuits*, Kluwer, 1995.