Introduction

- **Chemically Assembled Electronic Nanotechnology (CAEN):** proposed as a viable alternative to photolithography based silicon
- **High device densities:** $10^{10}$ gate-equivalents/cm$^2$ or more, against $10^7$ for CMOS
- **Extremely low cost of fabrication**
- **High defect densities:** up to 10% of components
  - (because we make it so)

**Problem:** to find a way to use defective chips
Using defective chips

- Use redundancy, as in memory chips
  - defect rates in CAEN devices too high
  - does not work for logic
- Use fault-tolerant circuit designs
  - large overheads (space and time)
  - needs hard upper bound on number of faults
  - circuit design is difficult
- Compose the fabrics of regular, repeating structures and use reconfiguration

We will use this last approach

---

Defect tolerance through reconfiguration

- Solution: suggested by reconfigurable FPGAs and Teramac custom computer
- Post-fabrication testing phase: locates and maps all defects
- Configurations routed around the defects
- Manufacturing time complexity traded-off for post-fabrication programming

We will call reconfigurable, CAEN based fabrics *nanoFabrics*
Routing around a defect

Requirements for testing

- The testing method used should not require access to individual fabric components
- It should scale with the number of defects
- It should scale with fabric size
- Testing should not become a bottleneck in the manufacturing process
Talk overview

- Introduction and motivation
- Our proposed solution
  - scaling with defect density
  - scaling with fabric size
- Simulations and Results
- Open Issues
- Conclusions

Testing method: overview

- **Test circuits** implementing a *chaotic* mathematical function
- Incorrect circuit output => defect!
- Correct circuit output => all its components are marked defect-free.

- Similarities with the **counterfeit coin problem**
  - however, they only find one coin!
- More importantly, **group testing**
Group testing

- Testing strategy which identifies +ves in a population by testing a group at a time
- Used for a wide-range of problems:
  - blood tests, product tests, multiple-access communication
  - more recently, in computational biology
- Has both adaptive and non-adaptive versions

- Constraints considered so far are different from ours
  - fewer number of +ves
  - possible to test individual members of population

Testing method: overview

- When are results analysed?
- Are tests adaptive or non-adaptive?
Test-circuits in action

Some terminology

- \( n \) components being tested
- Probability of defect \( p \)
- Each test circuit has \( k \) components
- Circuits arranged in various orientations, or tilings
- \% of good components recovered: yield

In the example,
- \( n=25 \)
- \( k=5 \)
- 2 tilings
- yield is 100%.
Assumptions

- **Permanent defects**
  - defective component always displays faulty behavior
  - defect in one component does not affect others
    - i.e., no short-circuits or stuck-at defects between wires
    - manufacturing process biased to ensure this
  - no Byzantine failures

- Defects in inter-connects: similar to defects in ordinary components

---

Assumptions (cont.)

- **Arbitrary, unlimited connectivity**
  - any component can be connected to any other, including non-adjacent ones
  - makes large number of tilings possible

- Above assumption: to simplify analysis
Scaling with defect density

- Expected $k*p$ defects/test-circuit
- Fewer defects/circuit: easier to locate
- We examine the following 3 cases:
  - $k*p \ll 1$
  - $k*p \approx 1$
  - $k*p \gg 1$
- Remember, $k$ cannot be too small

Low defect rates: $k*p \ll 1$ or $k*p \approx 1$

- Many test circuits have no defects
- Testing strategy:
  - configure test-circuits using a particular tiling
  - if any circuit’s output is correct, mark all components defect-free
  - repeat for many tilings
- Points to note:
  - tests are non-adaptive: all tilings known beforehand
  - no test-time “place-and-route” needed
Example with very low defect rate

Example with higher defect rate
Tilings required for low defect rates

![Graph showing the relationship between test circuit size and defect rates.](image)

Desired yield = 99%

Mahim Mishra

Carnegie Mellon

---

High defect rates: \( k \times p \gg 1 \)

- Many defects/test-circuit

- Finding a defect free circuit is extremely unlikely
  - e.g., for \( k=100, p=0.1 \), probability of finding a defect-free circuit = \( 1.76 \times 10^{-5} \)

- The previous approach does not work: something new is needed

Mahim Mishra

Carnegie Mellon
How can so many defects be located?

- Make $k$ smaller
  - $k*p$ is close to 1
  - may not be possible: no fine-grain access to components
  - increases test time

- Make the tester highly adaptive
  - tight feedback loop
  - result of each test determines configuration of next tester
  - will make testing very slow

- Use more powerful test circuits!

Making test circuits more powerful

- Use test-circuits which count defects
  - error in output depends directly on number of defects

- e.g., use error-correcting, fault-tolerant circuit designs

- These can return correct counts only up to a certain threshold
  - must indicate when threshold is crossed
  - use two different test circuits simultaneously!
New testing methodology

- Split into two phases:
  - probability-assignment phase
  - defect-location phase

- First phase: identifies components with high probability of being defect-free

- Second phase: tests these components further to pin-point defects
  - each phase: uses many different tilings

Probability-assignment phase

- Each component made a part of many different test circuits and defect counts are obtained

- Find probability of each component being good using Bayesian probabilistic analysis

- Discard components with low probability of being good
This works, but why?

- Intuitively, a defective component increases defect counts of all circuits it is a part of.

- If a component is part of many circuits with a high defect count, our analysis assigns it a low probability of being good.

- Precise mathematical model of this process: still under development.

---

Defect location phase

- Remaining components have low defect rate.

- Configure into test circuits, mark all the components good if circuit has no defects.

- Repeat for many different tilings.

- Everything left is marked bad.
Simulations

- For cases with low defect rates,
  - test-circuits gave 0-1 answers
  - measured yields for different number of tilings

- For cases with high defect rates,
  - test-circuits counted defects up to a certain threshold
  - measured yields obtained for different counting thresholds and different error rates

### Simulations with low defect densities

<table>
<thead>
<tr>
<th>Number of Tilings $t$</th>
<th>Expected Yield %</th>
<th>Achieved Yield %</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k=11$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p=0.009$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>91.36</td>
<td>91.34</td>
</tr>
<tr>
<td>2</td>
<td>99.25</td>
<td>99.29</td>
</tr>
<tr>
<td>$k=11$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p=0.09$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>38.94</td>
<td>38.05</td>
</tr>
<tr>
<td>2</td>
<td>62.72</td>
<td>62.05</td>
</tr>
<tr>
<td>5</td>
<td>91.51</td>
<td>91.17</td>
</tr>
<tr>
<td>10</td>
<td>99.28</td>
<td>99.24</td>
</tr>
</tbody>
</table>
Simulations with high defect densities

![Graph showing yield vs defect density for different defect densities](image)

*here, $k=101$, tilings used = 101*

---

Scaling with fabric size

- Each $k \times k$ piece of fabric requires
  - $O(k)$ tilings
  - therefore, $O(k)$ testing time
- Configure tested parts themselves as testers
  - reduces time on external tester
- Configure multiple testers simultaneously
- Wave-like progress of testing: total time needed is square root of fabric size

---

Mahim Mishra
Open issues

- **Accounting for limited fabric connectivity:**
  - we assume unlimited fabric connectivity
  - actual connectivity: will require lesser number of tilings

- **Using less restricted tilings:**
  - scalability of probability calculations needs to be checked

- **Accounting for real defect types and distributions:**
  - Byzantine defects
  - clustered defects
  - particular defect types such as *stuck-at* defects

---

More open issues

- **Exploring usability of alternative circuit types:**
  - Defect-counting circuits may be unrealizable
  - however, different, less powerful test circuits might also give useful information

- **Test circuit design:**
  - designing test circuits that satisfy our requirements will be a non-trivial task

- **Developing mathematical model of probability-assignment phase**
Conclusions

- CAEN-based computing fabrics with high defect densities can be used if we locate the defects and configure around them.

- To locate these defects, it is possible to devise a testing method which is scalable and has a high yield.

- Such a scalable testing method will require more powerful test circuits than are used currently.

Low defect rates: analysis

- If the desired yield is $y$ and the number of tilings required to achieve this is $t$,

  $$1 - \{1 - (1 - p)^{k-1}\}^t > y$$

  $$\Rightarrow t > \frac{\log (1 - y)}{\log \{1 - (1 - p)^{k-1}\}}$$

- For $k=10$ and $p=0.01$, a yield of at least 99% can be achieved with $t=2$, i.e., with only 2 tilings.
Medium defect rates: $k*p \approx 1$

- Expected 1 $(=k*p)$ defect/test-circuit
- About a third of the circuits are defect free
  - this is $(1 - p)^k \approx (1 - \frac{1}{k})^k \approx \frac{1}{e} \approx 0.35$
- Testing strategy used for the previous case works
- Caveat: many more tilings required
  - for $k=10$, $p=0.1$ and $y>99\%$, $t=10$

---

Probability calculation

If $A$ is the event of the component being good, and $B$ is the event of obtaining the defect counts $a_1, a_2, \ldots$ for it,

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A \cap B)}{P(A \cap B) + P(A^c \cap B)}$$

Simplification gives

$$P(A|B) = \frac{1}{1 + \frac{(1 - p)^{k-1}k^k}{p^{k-1}(k-a_1)(k-a_2)\ldots(k-a_k)}}$$
Scaling with fabric size (cont.)

- Testing proceeds in a wave through the fabric; the darker areas test and configure their adjacent lighter ones.
- Total time required equals the time for this wave to traverse the fabric, i.e., square root of the fabric size.