Space Compaction Multiple Outputs
We need to treat the general case of a \textit{k-output} circuit.

There are several possibilities:
- Multiplex the \textit{k} outputs of the CUT.

The multiplexer compacts the responses of each PO one at a time.
\textit{k} times slower but the $2^{-N}$ aliasing probability is reduced when multiple POs are tested independently.
Space Compaction Multiple Outputs

- Bellmac uses both *parity* and *signature analysis* compaction.

For example, given the error responses:

<table>
<thead>
<tr>
<th>Patterns</th>
<th>E1</th>
<th>E2</th>
<th>E3</th>
<th>E4</th>
<th>Parity</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>T2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>T4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

The "parity" polynomial, \( X^4 + 1 \), is then feed to the LFSR, which is divided by \( P(X) = X^4 + X + 1 \).

This yields a remainder of \( R(X) = X \).
Space Compaction Multiple Outputs
  • Parallel Signature Analysis (Multiple Input Signature Register or MISR).

This scheme is equivalent $k$ single input SAs but with the input stream
shifted in time, $M(X) = M_0(X) + XM_1(X) + \ldots + X^kM_k(X)$.

The error polynomial of the four outputs is $E(X) = E_1(X) + XE_2(X) + X^2E_3(X) + X^3E_4(X)$, which is divided by the $P(X)$ yielding a remainder of $X^3 + X + 1$.

Error Responses

Note that the aliasing probability of the MISR is still $2^{-N}$ for an $N$-stage SA. When the number of outputs, $k$, of the CUT is $> N$, parity/MUX can be used.
Random Pattern Resistant Faults

The effectiveness of any test can be measured by:

- It’s fault coverage
- It’s length
- It’s hardware requirements
- It’s data storage requirements

PR tests generated according to previous methods are usually long and result in unacceptable fault coverage:

**Saturation** follows the rapid increase in fault coverage.
Random Pattern Resistant Faults

$\Delta FC$ represents the hard-to-detect faults by random patterns (RPR).

The fault coverage can be improved by reducing the aliasing probability. However, the main source of difficulty is that some faults are detected by only a couple, possibly one, patterns.

The root of the problem: Under PR pattern generation, all FFs have equal probability of generating a 1 or 0.

However, detection probabilities for faults in gates do not follow this distribution, e.g., only 1 pattern detects an $SA0$ on an input to a 6-input NOR.

Minimal SAF tests

More 1s

1101
1011
0111

More 0s

0010
0100
1000

6 patterns (of 32 exhaustive patterns) give 100%

Note 1s and 0s do not occur uniformly.
Random Pattern Resistant Faults

Weighted PR TPG assigns weights to the PIs, the probability that 1 should be assigned to a PI.

Weight assignment can be based on circuit structure analysis or fault detection probabilities.

Although coverage is improved, there are still hard-to-detect faults. This results from fan-out, e.g., an input common to the AND and OR gate is assigned a weight that favors one over the other.

Multiple weights is a solution but adds hardware.

Other solutions: test point insertion, reseeding the LFSR and multiple polynomial LFSRs add hardware, impact performance and/or require long tests.

Mixed-mode approach uses deterministic patterns stored in ROM or via bit-fixing/flippering from LFSR patterns for RPR faults.

No good solutions, deterministic patterns are typically applied via scan path.
BIST Architectures

The LFSR and SA can be on-chip or off-chip, and as indicated, logic BIST typically combines PR testing with scan and boundary-scan.

Autonomous Test

Circuit is partitioned using MUXs or sensitization method.

Each is tested independently using the same LFSR and MISR.
BIST Architectures

**Circular BIST:** For register-based architectures, self-test shift registers (STSR).

Three phases to the test: **Initialization:** all STSR and FFs. **Test mode:** all STSR act as LFSR and MISR. **Response Eval:** STSRs are compared with fault-free value.

MISR using all STSR has characteristic polynomial $1 + X^N$.

Text shows another version.
BIST Architectures

**BILBO** (Built-In Logic Blocks Observer): BIST + Scan Path.

Combines TPG and response compression in a single unit (designed for bus-oriented systems).

It uses existing FFs on-chip for PR TPG and SA.

C₁ and C₂ configure as a **shift register** for scan (00), an **LFSR** (00), **MISR** (10) a **Normal** (11).
BIST Architectures

BILBO test scenario:

Normal

Test mode

Each combo block is tested one at a time. For testing Combo-1, Reg 1 configured as PRTPG (LFSR) and Reg 2 configured as MISR.

So testing Combo-1 involves configuring BILBO as a **MISR**.

Afterwards, testing Combo-2 involves configuring BILBO as an **LFSR**.
BIST Architectures

**Random Test Socket:** Combines scan and BIST.

All PIs are connected to the taps of LFSR #1 and all POs to the MISR. FFs are scannable and form a Shift Register (SR).

SI is driven by LFSR #2 while SO is connected to the SSA.

![BIST Architecture Diagram](image)

1) Load SR with pattern from LFSR #2
2) Apply pattern using LFSR #1 to PIs.
3) Clock to latch response in SRs.
4) Capture results in MISR (SE = 0).
5) Scan out SR into SSA.

(Steps 1 and 5 can be overlapped).

Called "test per scan" instead of "test per clk" since shifting is necessary.

Note, LFSR 1 and 2 can be combined as well as the MISR and SSA.

Adv: low-cost ATPG, Disadv: overhead and long test times.
BIST Architectures

**STUMPS**: Self-Test Using MISR and Parallel Shift reg. sequence generator.

Originally proposed to reduce overhead of LFSR/MISR for application to testing multi-chip boards, each of which has only the SRs.

Can also be used on a single chip with **multiple scan chains**.

Inputs to all scan chains provided by multiple-output LFSR.

In order to break linear dependency, **Phase shifters** (XOR gates) added.