Debugging Functional Coverage Models
Get The Most Out of Your Cover Crosses

Mennatallah Amer, Amr Hany
Mentor, A Siemens Business
Problem

• Functional Coverage
  – Coverage from a system point of view
  – Cover variables, expressions and their combinations
  – Specify testplan in a simple and concise way
• But
  – Large number of cross bins
  – Covergroups contain hundreds of cover-crosses
  – Reports become too tedious to analyze
• How to identify large holes effectively?
Real Case Study

• 138 coverpoints
• 202 crosses
• Coverage: 72.9%
• Hit Percent: 30.79%
• There are a lot of holes
• From where to start?
Least Covered Cross

- **Coverage**: 3.3%, 9 coverpoints
- **Missed/Total Bins**: 464/ 480

<table>
<thead>
<tr>
<th>Opcode</th>
<th>priv_mode</th>
<th>debug</th>
<th>reg</th>
<th>usr_en</th>
<th>hdcr</th>
<th>tpmcr</th>
<th>hstr</th>
<th>excp</th>
<th>hit</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCR</td>
<td>ns_user</td>
<td>0</td>
<td>pmcr</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>null_req</td>
<td>0</td>
</tr>
<tr>
<td>MRC</td>
<td>ns_user</td>
<td>0</td>
<td>pmcr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>null_req</td>
<td>0</td>
</tr>
<tr>
<td>MCR</td>
<td>ns_user</td>
<td>0</td>
<td>pmcr</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>undef_req</td>
<td>0</td>
</tr>
<tr>
<td>MCR</td>
<td>ns_user</td>
<td>0</td>
<td>pmcr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>undef_req</td>
<td>0</td>
</tr>
<tr>
<td>MCR</td>
<td>ns_user</td>
<td>0</td>
<td>pmcr</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>undef_req</td>
<td>0</td>
</tr>
<tr>
<td>MRC</td>
<td>ns_user</td>
<td>0</td>
<td>pmcr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>undef_req</td>
<td>0</td>
</tr>
<tr>
<td>MCR</td>
<td>ns_user</td>
<td>0</td>
<td>pmcr</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>undef_req</td>
<td>0</td>
</tr>
<tr>
<td>MCR</td>
<td>ns_user</td>
<td>0</td>
<td>pmcr</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>undef_req</td>
<td>0</td>
</tr>
<tr>
<td>MRC</td>
<td>ns_user</td>
<td>0</td>
<td>pmcr</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>undef_req</td>
<td>0</td>
</tr>
<tr>
<td>MRC</td>
<td>ns_user</td>
<td>0</td>
<td>pmcr</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>undef_req</td>
<td>0</td>
</tr>
</tbody>
</table>

Rows only differ in a single coverpoint value
Automated Hole Analysis

- Coverage: 3.3%, 9 coverpoints
- Missed/Total Bins: 464/480
- Azatchi et al, 2006
- 26 distinct bucketed holes!

<table>
<thead>
<tr>
<th>Opcode</th>
<th>priv_mode</th>
<th>debug</th>
<th>reg</th>
<th>usr_en</th>
<th>hdr</th>
<th>tpmcr</th>
<th>hstr</th>
<th>excp</th>
<th>Missed</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>1</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>256</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>224</td>
</tr>
<tr>
<td>*</td>
<td>ns_user,ns_krn,*</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>1</td>
<td>*</td>
<td>176</td>
</tr>
<tr>
<td>*</td>
<td>ns_user,s_usr, *</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>1</td>
<td>*</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Top Hole, has 256 missed bins, when usr_en is 1

Holes can be sorted by missed bins

Larger holes, formed by aggregation of smaller projected hole. priv_mode has multiple values ns_user,ns_krn

Holes can be sorted by missed bins
Cool, But....

• Solution for a single cross
• 201 Crosses to go!

229 Cross pairs sharing up to 6 coverpoints!
Architecting Cross-Coverage

• Based on extensive discussions/iterations
• Each cross captures a desired behavior
• Overlaps exist due to:
  – Bins in each cross grouped in a different way
  – Marking various illegal combinations
  – Whole cross is banking another signal
• Crosses can be analyzed simultaneously:
  – Multi-Hole Analysis
  – Rank new aggregated reports
Projection

• Can not run analysis on all coverpoints of the selected crosses:
  – Computationally infeasible
## Multi- Hole Analysis

- **9 Crosses, Coverage: 11.08%, 5 coverpoints**
- **Missed/Total Bins: 15850/17570**

### A single Report/ 34 buckets

<table>
<thead>
<tr>
<th>debug</th>
<th>reg</th>
<th>priv_mode</th>
<th>excp</th>
<th>lock</th>
<th>Missed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>1</td>
<td>3976</td>
</tr>
<tr>
<td>1</td>
<td>*</td>
<td>ns_user,ns_krnl...</td>
<td>*</td>
<td></td>
<td>3788</td>
</tr>
<tr>
<td>1</td>
<td>osdlr,prcr,...</td>
<td></td>
<td></td>
<td></td>
<td>2640</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>osdlr,prcr,...</td>
<td>ns_mon</td>
<td></td>
<td>2313</td>
</tr>
<tr>
<td>1</td>
<td>vcr,dsmcr,..</td>
<td>*</td>
<td>null_req</td>
<td></td>
<td>1344</td>
</tr>
<tr>
<td>1</td>
<td>*</td>
<td>ns_hyp,s_user</td>
<td>undef_req</td>
<td></td>
<td>1188</td>
</tr>
<tr>
<td>1</td>
<td>jidr,dttrx,...</td>
<td>*</td>
<td>undef_req</td>
<td></td>
<td>1022</td>
</tr>
<tr>
<td>1</td>
<td>teecr,teehbr,...</td>
<td>ns_hyp</td>
<td>*</td>
<td></td>
<td>844</td>
</tr>
</tbody>
</table>

Top common hole having 3976 missed bins

**Is this the best ranking?**
Hole Effect

- **9 Crosses**, Coverage: 11.08%, 5 coverpoints
- **Missed/Total Bins**: 15850/17570
- Cross bins [56-7936]
- **Hole Effect** proportional to coverage increase

\[
\text{covInc}_c = \frac{\text{miss}_c}{\text{bin}_c}
\]

\[
\text{HoleEffect}_c = \frac{\sum_{c \in C} w_c \cdot \text{covInc}_c}{\sum_{c \in C} w_c}
\]
### Multi-Hole Analysis

- **9 Crosses, Coverage:** 11.08%, 5 coverpoints
- **Missed/Total Bins:** 15850/17570
- **A single Report/ 34 buckets**
- **Hole Effect**

<table>
<thead>
<tr>
<th>debug</th>
<th>reg</th>
<th>priv_mode</th>
<th>excp</th>
<th>lock</th>
<th>Missed</th>
<th>Hole Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>osd1r, pccr, ...</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>2640</td>
<td>26.17</td>
</tr>
<tr>
<td>1</td>
<td>*</td>
<td>ns_user, ns_krnl...</td>
<td>*</td>
<td>*</td>
<td>3788</td>
<td>20.10</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td></td>
<td>*</td>
<td>1</td>
<td>3976</td>
<td>15.00</td>
</tr>
</tbody>
</table>

**Hole having largest number of bins ranked 3rd, having hole effect to 15%**

**New top hole would cause a coverage increase of 26.17%**
Summary

• Aggregated report
  – Easy to rank/analyze
  – Identify global holes

• Computationally feasible
  – Independent of number of crosses
  – Cost of a single hole analysis run

• Hole Effect for ranking top holes

• Save time and effort of analyzing complete coverage model
What's next?

• Automate the start of the analysis:
  – Automatically select cover point/crosses
  – Leverage clustering algorithms
  – No human-driven factor

• Bridge the gap between complex cover groups and analysis engine
References


• A. Yehia, "Faster coverage closure: Runtime guidance of Constrained Random stimuli by collected," Electronics, Communications and Photonics Conference (SIECPC), May 2013

Questions?