Building a Test Plan

Alon Shtepel

Yaniv Barash
Test Plan vs. Verification Plan

- **Verification Plan** (Environment Architecture):
  - Derived from System and Macro Spec.
  - High level description.
  - Targets, goals, feasibility

- **Test Plan**:
  - Detailed list
  - Derived from all documents.
Test Plan Specifications

- Resources:
  - MRD, System Spec., Hardware Specs (Macro and Micro)
  - Designers notes.
  - Common tests for all chips.
  - Experience.
Test Plan Specifications (2)

- Consider:
  - IPs, Reused blocks, New blocks
  - Verification methodology - Random/direct
  - Granularity – System, ASIC, Block, Unit.
  - Verification capabilities
Documentation Flow

- Features List
  - Environment Architecture
  - Direct Tests Plan
  - Coverage Plan
  - Direct Tests Definitions
Format

- Different generators / checkers
- High Level / Low level
- Priorities
- Monitoring capabilities
<table>
<thead>
<tr>
<th>No</th>
<th>chapter</th>
<th>No</th>
<th>sub chapter</th>
<th>description</th>
<th>Prto</th>
<th>Gen tool</th>
<th>checked By</th>
<th>Coverage</th>
<th>Direct tests</th>
<th>Test Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ingress</td>
<td>1.1</td>
<td>Interfaces</td>
<td>Ingress &lt;-&gt; Express interface. Cover all relevant options of data in to the unit</td>
<td>1</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Prs_tests</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2</td>
<td>Counters</td>
<td>Counters compare at End of Test Counters reading during test</td>
<td>2</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Prs_tests</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.3</td>
<td>Errors</td>
<td>Oversize, undersize, jabber, fragments, wrong CRC etc.</td>
<td>3</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Prs_tests</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.4</td>
<td>Interrupts</td>
<td>Check each interrupt bit + mask + sum by causing the relevant scenario. Check more than one interrupt indication without CPU reading.</td>
<td>4</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Prs_tests</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.5</td>
<td>Arbirs, files, CMs</td>
<td></td>
<td>5</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Prs_tests</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.6</td>
<td>Debug bus</td>
<td></td>
<td>6</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Prs_tests</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.7</td>
<td>RegFile</td>
<td>Access to all registers and tables for read and write</td>
<td>7</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Prs_tests</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.8</td>
<td>Access in bursts</td>
<td>Access to non exist registers</td>
<td>8</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Prs_tests</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.9</td>
<td>clock domains</td>
<td>Data path</td>
<td>9</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Prs_tests</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.10</td>
<td>MO configurations</td>
<td>Access to memory</td>
<td>10</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>Prs_tests</td>
<td>v</td>
</tr>
</tbody>
</table>
Definition

- **When:**
  - Must we wait until Arch is closed? (features)
  - Design status - Started? Finished? Other? (components – fifos, arbiters, SMs etc.)

- **Who:**
  - Arch, µArch, Design, Verification, Other?
  - Split between several engineers
Review

- Reviewers:
  - Arch, µArch, PL, Design, Verification (All?), Others?

- Risk evaluation
  - Priority = risk x feature necessity

- Depending on the status of:
  - Architecture, Design, Verification

- Re-review
  - Code completion? RTL Freeze? Tape out?
The End

- Thank You!