**Using Portable Stimulus to Verify an LTE Base-Station Switch**

Adnan Hamid (adnan@brekersystems.com)
Breker Verification Systems, 1879 Lundy Avenue, Ste 126, San Jose, CA 95131
www.brekersystems.com

### Objective

Generate software driven verification (SDV) test cases in "C" for an LTE Base Station Switch

- Complex design supporting many possible use models
- Particular focus on dependency management across tasks and resources
- Create single verification model that can scale from simple to complex tests
- Crate portable stimulus model that can generate tests for simulation, emulation and post-silicon
- Crate self-checking test cases that are easy to debug
- Prove coverage closure on system use cases

### LTE Base Station Switch Design

<table>
<thead>
<tr>
<th>CPU 1</th>
<th>CPU 2</th>
<th>CPU 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Dispatcher</td>
<td>Task Dispatcher</td>
<td>Task Dispatcher</td>
</tr>
<tr>
<td>DMA Engine</td>
<td>DMA Engine</td>
<td>DMA Engine</td>
</tr>
<tr>
<td>Memory Controller</td>
<td>Memory Controller</td>
<td>Memory Controller</td>
</tr>
<tr>
<td>Memory</td>
<td>Memory</td>
<td>Memory</td>
</tr>
</tbody>
</table>

- CPUs configure Task Dispatcher with 1000's of tasks
- Task Dispatcher schedules tasks on Taskers as resources become available
- Each task has many possible formats with different paths through system
- Each task has many complex dependencies on other tasks and resources
- Task Dispatcher must manage dependencies across tasks and resources
- Task Dispatcher must track progress and completion of tasks to free resources
- One Task Dispatcher and multiple Taskers in each cluster
- Multiple clusters in full chip

### Verification Requirements

**Complexity Requirements**

- Individual tasks have many possible formats and many possible resource dependencies
- Need to test complex dependency scenarios across tasks, task dispatchers and taskers (Figure 2).
- Need to generate complex data structures in memory
- Need to manage memory across multiple regions
- Task Dispatchers have complex configurations with many variations

**Scalability Requirements**

- Scaling of generated tests from simple to complex
- Scaling of system under test from sub-system to full-chip

**Portability Requirements**

- Single source of tests must run on simulation, emulation, post-silicon
- Support different test case mechanics for each platform
  - e.g. back-door byte-by-byte memory checking in simulation, but front-door checksum-based checking in post-silicon

**Checking and Debug Requirements**

- Automatic generation of self-checking tests
- Track progress of each task through life-cycle
  - All tasks completed
  - Tasks completed in dependency order
- Predict and check results in memories and registers
- Interactive debug to identify and root-cause errors
  - must work in simulation / emulation / post-silicon

**Coverage Requirements**

- Prove coverage closure on all use cases

### Test Generation Tool

- Use Breker TrekSoC product as tool of choice for scalable portable stimulus (Figure 3)
- Algorithmic graph-based scenario models to capture verification space

### Method

- Common scenario model used at all stages of verification (Figure 4)
- TrekSoC tool configured for each environment
  - complexity of test to generate
  - number of CPUs, Task Dispatchers and Taskers
  - Number and size of memory regions
  - Communication mechanism between test and TrekBox

### Results

**Scenario Model Construction**

- Algorithmic graph-based portable stimulus scenario model "blind-built" from spec before simulation available
- First generated test running within hours of initial bring-up!
  - graph-based models allow simple creation of complex test scenarios
  - test cases correct by construction, even with 1000's of interacting tasks and complex Task Dispatcher programming cases

**Test Complexity Scaling**

- Seamless scaling of test complexity from single task to 1000's of inter-dependent tasks, and from single CPU/Task Dispatcher/Tasker to multiple CPU/Task Dispatchers/Taskers across multiple clusters

**Portable Stimulus**

- Target specific self-checking tests generated for simulation and emulation with different test case mechanics
  - Easy to replay failing emulator cases on simulation
  - Waiting for silicon to generate post-silicon tests

**Checking and Debug**

- Task dependency graph generated from tool (Figure 2) used to help follow test case intent
- Design errors rapidly identification and root-caused, with common debug view across all platforms

**Coverage Closure**

- Path coverage and automatic coverage closure of graph-based models provide metrics for use case coverage
  - cannot be measured with RTL functional coverage

### Realized Benefits

- Bugs Found
  - Hardware, software, and environment bugs found and fixed at each level of application

- Effort & Cost Savings
  - Automated generation of 1000's of tests for emulation/Si
  - Could not achieve this coverage with manually written tests

---

For more information on this verification process, see www.brekersystems.com