Artisan Universal Test Interface™

Introduction

UTI Functionality Issues

UTI Truth Table

Non-pipelined 1-Port SRAM UTI

Pipelined 1-Port SRAM UTI

Implementation and Test Examples

Conclusions
Introduction

- The Artisan Universal Test Interface™ (UTI™) is designed to work with the widest possible array of physical test interfaces to embedded SRAM.

- Function and pins are optimized to satisfy the requirements of diverse and unrelated customers.

- The UTI is tightly integrated by customizing design and layout in the critical path of the embedded memories.

- The interface is verified for, but not limited to 4 test styles:
  - Serial BIST
  - Full Scan
  - Parallel Test (“Mux Isolation”)
  - Serial Test
UTI Functionality Issues

- Conceptually, the UTI is simply test input muxes and scan flip-flops; however there are several control issues:

- The output (Q) bus requires tristate control during mission and test modes.

- A dedicated, shared test output bus is also required by some users: TQ

- UTI flip flops are gated by internal Global Timing Pulse (GTP).

- GTP activation is required during scan/serial test, even if chip enable input (TCEN) state is random.

- SRAM core write must be suppressed during serial shift/scan, even if the write enable input (WEN/TWEN) is random.

- The mission mode pipeline stage is re-used for scan/shift test activities.
### UTI Truth Table

<table>
<thead>
<tr>
<th>TIS</th>
<th>TMS</th>
<th>TOEN</th>
<th>OEN</th>
<th>Effective Address</th>
<th>Effective Data</th>
<th>Q</th>
<th>Pipeline Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>A</td>
<td>D</td>
<td>from memory</td>
<td>Qi</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>A</td>
<td>D</td>
<td>hi-Z</td>
<td>Qi</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>A</td>
<td>D</td>
<td>from pipeline</td>
<td>Di</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>A</td>
<td>D</td>
<td>hi-Z</td>
<td>Di</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>TA</td>
<td>TD</td>
<td>from memory</td>
<td>Qi</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>TA</td>
<td>TD</td>
<td>hi-Z</td>
<td>Qi</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>TA</td>
<td>TD</td>
<td>from pipeline</td>
<td>Di</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>TA</td>
<td>TD</td>
<td>hi-Z</td>
<td>Di</td>
</tr>
</tbody>
</table>
Pipelined 1-Port SRAM UTI
Implementation and Test Examples

Serial BIST Example

Controlled Inputs: TA[m-1:0], TD[0], CLK, TCEN, TWEN, TOEN, TMS, TIS

Strobed Outputs: Q[n-1]

Serial Chain:

- TD[0] is serial input
- Starting at Q[0], connect Q[i] to TD[i+1]
- Q[n-1] is serial output

It is possible to construct the serial chain by connecting TQ outputs to TD inputs. In the case of a non-pipelined memory, this modification would add an additional cycle of latency.

Serial BIST Test Example

1. Serially load test data word by performing repeated write operations. The outputs generated by each write operation are shifted and re-written by the subsequent write operation. Repeat for all memory locations.

2. Simultaneously shift out previous test data and shift in new test data by alternately performing read and write operations for each data bit. The results of each read operation are shifted and re-written by the subsequent write operation. Strobe the serial output after each iteration. Repeat for all memory locations.

3. Serially shift out final test data by alternately performing read and write operations for each data bit. The results of each read operation are shifted and re-written by the subsequent write operation. Repeat for all memory locations.

Repeat stage (2) as desired for different data patterns. Repeat entire serial BIST procedure for all memories.
Scan Test Example

Controlled Inputs: TA[m-1], CLK, OEN, TQOEN, TMS, TIS

Strobed Outputs: TQ[n-1]

Serial Chain:
- TA[m-1] is serial input
- Starting from ASQ[m-1], connect ASQ[i] to TA[i-1]
- Connect ASQ[0] to TWEN
- Connect WENSQ to TOEN
- Connect OENSQ to TCEN
- Connect CENSQ to TD[0]
- Starting at TQ[0], connect TQ[i] to TD[i+1]
- TQ[n-1] is serial output

Scan Test Output: Q[n-1:0]

Scan test is conducted in three stages:

1. Serially load the RAM scan chain elements with test data.

2. Perform a mission-mode cycle (also called parallel mode or functional mode). When the clock rises, the scan vector loaded in stage (1) supplies stimulus to elements downstream of the memory. At the same time, the input scan flip-flop slave stages are over-written by mission-mode inputs.

3. Serially shift out contents of UTI scan chain. Check the validity of scan data.
Parallel Test Example

Controlled Inputs: TA[m-1:0], TD[n-1:0], CLK, TCEN, TWEN, TQOEN, TMS, TIS
Strobed Outputs: TQ[n-1:0]

For parallel test, write cycles and read cycles are enabled through the test inputs and evaluated through test outputs. Parallel test is conducted in two stages:

1. Write memory directly, stimulating controlled inputs bit for bit. Repeat for all RAM locations.

2. Read memory directly, stimulating controlled inputs (except TD[n-1:0]) and checking strobed outputs bit for bit. Repeat for all RAM locations.

Repeat stages (1) and (2) as desired for different data patterns. Repeat entire parallel test for all memories.
Serial Test Example

Connect test circuit or stimulus to controlled inputs and strobed outputs as described below:

Controlled Inputs: TA[m-1], CLK, TCEN, TWEN, TOEN, OEN, TMS, TIS

Strobed Outputs: Q[n-1]

Serial Chain:  
- TA[m-1] is the serial input  
- Starting at ASQ[m-1], connect ASQ[i] to TA[i-1], continuing until TA[0]  
- Connect ASQ[0] to TD[0]  
- Starting at TQ[0], connect TQ[i] to TD[i+1], continuing until TD[n-1]  
- TQ[n-1] is the serial output

For serial test, write cycles and read cycles are enabled and evaluated through test inputs and the scan flip-flop chain. Serial test is conducted in two stages:

1. Serially load write data and address into UTI data and address flip-flops. Perform a write operation. Repeat for all RAM locations.

2. Serially load read address into UTI address flip-flops. Perform a read operation. Serially shift out read data from UTI data flip-flops. Check validity of read data.

Repeat stages (1) and (2) as desired for different data patterns. Repeat entire serial test for all memories.
Conclusions

- The integrated approach of the UTI increases SRAM design effort and perceived area, but lowers overall cost to chip speed and area.

- Generally, customer acceptance and use of the UTI is very high.

- Some customers have locally-developed embedded macro test interfaces which replicate the function of the UTI.

- Some customers with their own test interface methodology and some customers using small memories request the option to generate SRAM without UTI.

- The design, test, and support of views for various integrated test products requires non-trivial effort and close cooperation by the test vendors.

- There is a wide variance among end users in the effort spent to integrate test capability, and their awareness of the cost-saving potential of BIST and integrated test interfaces.