Register and Memory Test

Draco™ 3.0
May 1, 2006

TAREK VERIFICATION SYSTEMS, LLC

Contents

1 Register Access Mode Test (_regTestMode = 1)

Unless specified explicitly, four set of test vectors are applied to each register, which are 0xa5a5...a5a5, 0x5a5a...5a5a, 0xffff...ffff, and random value.

1.1 RO Test

1. force the register with test vector
2. bus-read the register, and compare the data with the expected value
3. if peek is allowed, and compare the data with the expected value in order to verify the RTL path
4. release the register

1.2 WO Test

1. bus-read of the register must be default value of 0 (Note: The scenario may not be true depending upon design. In this case, the register needs customized test with testability of 0)
2. bus-write and read: read value must be default value of 0 (Note: The scenario may not be true depending upon design. In this case, the register needs customized test with testability of 0)

(if poke is allowed)

1. peek and compare the data with test vector to verify the RTL path

1.3 RC Test

(if poke is allowed)

1. poke test vector
2. bus-read, and compare the data with the expected value to verify the RTL path
3. bus-read to ensure the read value is 0

(if poke is allowed)

1. bus-read
2. bus-read to ensure the read value is 0

1.4 WC Test

(if poke is allowed)

1. bus write
2. after CLR_DELAY cycles, peek, and verify that peek value is 0 (verify the RTL path, and writing)

(if poke is not allowed, then cannot test)

1.5 RCW Test

1. bus-write test vector
2. bus-read and compare the data with the expected value
3. if poke is allowed, then peek data and compare the data with the expected value
4. bus-read to ensure the read value is 0

1.6 RW Test

1. bus-write test vector
2. bus-read and compare the data with the expected value
3. if poke is allowed, then peek data and compare the data with the expected value

1.7 RWC Test

(if poke is allowed)

1. poke test vector
2. bus read, and compare test vector vs. read value (Read test)
3. bus write test vector, and peek, and compare peek value vs. test vector (Write test)
4. after CLR_DELAY cycles, peek and verify that peek value is 0 (Clear test)

(if poke is not allowed)

1. bus write test vector
2. bus write test vector again
3. bus read, and read value = 0

1.8 RWC0 Test

(if poke is allowed)

1. poke test vector
2. bus read, and compare test vector vs. read value (Read test)
3. bus write the test vector again, and peek, and compare peek value vs. test vector (Write test)
4. after CLR_DELAY cycles, peek, and verify that peek value is the same as test vector (Clear test)

(if poke is not allowed)

1. bus write test vector
2. after CLR_DELAY cycles, read, and verify that the read value is 0

1.9 RWC1 Test

(if poke is allowed)

1. poke test vector
2. bus read, and compare test vector vs. read value (Read test)
3. bus write the inverse of test vector and peek and compare (Write test)
4. after CLR_DELAY cycles, peek, and verify that the peek value is 0 (Clear test)

(if poke is not allowed)

1. bus write test vector
2. after CLR_DELAY cycles, read, and verify that the read value is 0

1.10 RCWC Test

(if poke is allowed)

1. poke test vector
2. bus read, and compare with test vector
3. after CLR_DELAY cycles, peek, and compare with 0 (RC test)
4. poke test vector
5. write the same test vector
6. after CLR_DELAY cycles, peek, and compare with 0 (WC test)

(if poke is not allowed)

1. bus write
2. after CLR_DELAY cycles, bus read and compare with 0

2 Register Reset Test (_regTestMode = 2)

1. peek the register value if allowed, otherwise bus-read the register value
2. compare the data with reset value

3 Register Sweeping (_regTestMode = 3)

Register sweeping test verifies the register address decoding block. Unique value is written to each RW register sequentially, and the read the value of each register in order to compare with the written value. Registers with the testability of 2 are excluded from register sweeping tests, because they control bus-transactions directly.

4 Memory Test

There are two modes for selecting rows for memory test. If testability of the memory is 1, then randomly select rows for memory test. If the testability is 2, then test all the rows.
For each selected row,

4.1 RW Test

1. bus-write the test vector into the row
2. bus-read and compare the data with the expected value
3. if poke is allowed, then peek and compare the data with the expected value

4.2 RO Test

(if poke is allowed)

1. poke test value
2. bus-read and compare

4.3 RC Test

(if poke is allowed)

1. poke test value
2. bus-read and compare with the applied test value
3. bus-read again to make sure the clean memory entry

(if poke is not allowed)

1. bus-read
2. bus-read again to make sure the clean memory entry