ScalaHDL

by Yao Li

Contents

Using Modules in a Module

Using Modules in a Module

```
sync('clk is 1)

defMod.logic('d, 'q, 'clk) {

  'q := 'd

}


delay(10)

defMod.clkGen('clk) {

  'cycle('clk)

}
```

```
sync('clk is 0)

defMod.stimulus('d, 'clk) {

  'd := Random.nextInt(2)

}


defMod.cycle('x) {

  'x := not('x)

}
```

Using Modules in a Module

- Not intended to support recursion (and features like “generate” in Verilog).

Test Bench

Test Bench Example 1

```
class MainTest extends FunSuite with
TestHelper {

  test("test dff 1") {

    val q = signal(0)

    val d = signal(1)

    val clk = signal(0)

    val sim = Simulator(Main,

      module('logic, d, q, clk),

      module('clkGen, clk),

      module('stimulus, d, clk))


    sim.simulate(10)

    assert(clk === 1)

    for (i <- 1 to 100) {

      sim.continue(10)

      assert(clk === 0)

      sim.continue(10)

      assert(clk === 1)

      assert(q.value === d.value)

    }

    sim.stop()

  }

}
```

Test Bench Example 2

```
class MainTest extends FunSuite with
TestHelper {

  test("test dff 2") {

    val q = signal(0)

    val d = signal(1)

    val clk = signal(0)

    val sim = Simulator(Main,

      module('logic, d, q, clk),

      module('clkGen, clk),

      module('stimulus, d, clk))

    sim since 0 to 1000 every 20 run {

      assert(clk === 0)

    }

    sim since 10 to 1000 every 20 run {

      assert(clk === 1)

      assert(q.value === d.value)

    }

    sim test

  }

}
```

Equality

- Given two signal `a` and `b`, what is `a == b` supposed to mean?
  - `a.value == b.value`
  - `a.value == b.value && a.bits == b.bits`
  - `a.value == b.value && a.bits == b.bits && a.name == b.name`
  - `a.hashcode == b.hashcode`
- For now, in ScalaHDL:
  - `a == n`, where `a` is a `Signal`, `n` is a `Int`, means `a.value == n`
  - `a == b`, where both `a` and `b` are `Signal`s, means all the fields of `a` and `b` are equal

Follow-up Work

Follow-up Work

- More on syntax specifications.
  - e.g. the “if” problem.
- More tests.

Any Questions?

Thanks!