Traces in ScalaHDL

by yao Li

Contents

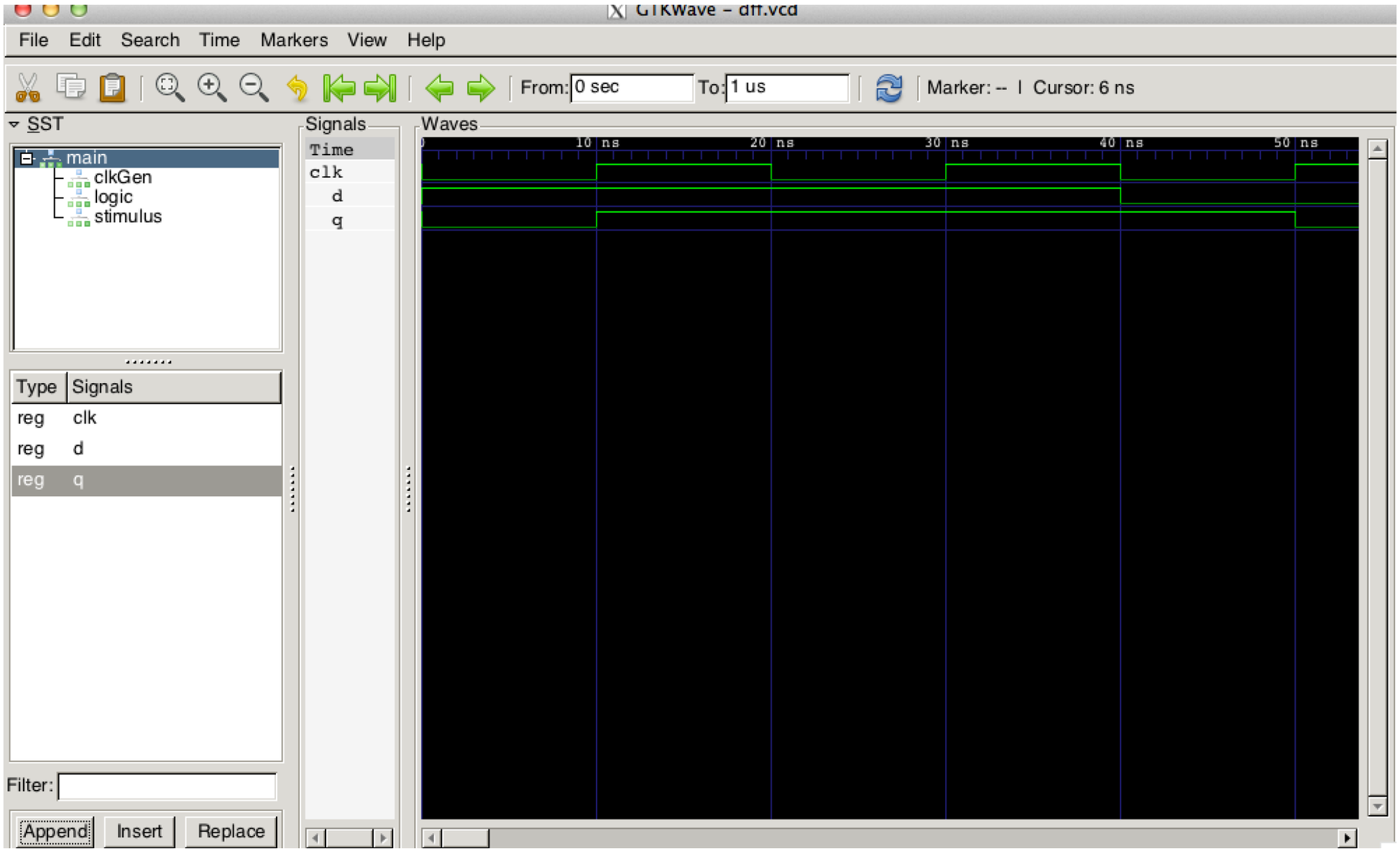
- Traces
  - usage
  - implementation
- Tests
- Follow-up Works

Traces: Usage

```
object Main extends ScalaHDL {

  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)

}
```

```
def main(args: Array[String]) {

  val q = Signal(0, 1)

  val d = Signal(1, 1)

  val clk = Signal(0, 1)

  val sim = Simulator(this,

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

    module('clkGen, clk),

    module('stimulus, d, clk))

  sim.simulate(1000, "dff.vcd")

  sim.stop()

 }

}
```

Traces: VCD


GTKWave - dff.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 1 us
Marker: -- | Cursor: 6 ns
SST
main
clkGen
logic
stimulus
Type Signals
reg clk
reg d
reg q
Filter:
Append Insert Replace
Signals
Time
clk
d
q
Waves
10 ns
20 ns
30 ns
40 ns
50 ns

Implementation: Naming

- Use **dsinfo** library (https://bitbucket.org/inkytonik/dsinfo/overview).

- The Signal is used in this way:
  - `val q = Signal(0, 1)`
- But the Signal is actually defined as:
  - `class Signal(val name: String, var _value: Int, _bits: Int)`

Implementation: Naming

```
object SignalMaker {

  def Signal(value: Int, bits: Int): Signal =

    macro SignalMaker.makeSignalWithName


  def mkSignal(name: String, value: Int, bits: Int): Signal =

    new Signal(name, value, bits)


  def makeSignalWithName(c: Context)(value: c.Expr[Int], bits: c.Expr[Int]) =

    makeCallWithName(c, "SignalMaker.mkSignal")

}
```

Tests

- Tests for wire and schedule methods of Simulator have been written. Some bug were found through testing (and fixed).

Follow-up Works

- Enable a module to use other modules (currently not supported).
- Test bench.
- Finish some unfinished parts of previous work.

Any Questions?

Thanks!