Circuit Capture of Chisel Part 2. Naming

Boyang Han
yqszxx@gmail.com

7/29/20

Problems involved

- Correctness
- Naming
- Debug info

What is circuit capture?

```scala
import chisel3._

class DecCounter extends Module {
  val io = IO(new Bundle{
    val value = Output(UInt(4.W))
  })

  val counter = RegInit(0.U(4.W))

  io.value := counter

  when (counter === 9.U) {
    counter := 0.U
  } otherwise {
    counter := counter + 1.U
  }
}
```

```
circuit DecCounter :
  module DecCounter :
    input clock : Clock
    input reset : UInt<1>
    output io : { value : UInt<4>}

    reg counter : UInt<4>, clock with :
      reset => (reset, UInt<4>("h0")) @[main.scala 8:24]
    io.value <= counter @[main.scala 10:12]
    node _T = eq(counter, UInt<4>("h9")) @[main.scala 12:17]
    when _T : @[main.scala 12:26]
      counter <= UInt<1>("h0") @[main.scala 13:13]
    else :
      node _T_1 = add(counter, UInt<1>("h1")) @[main.scala 15:24]
      node _T_2 = tail(_T_1, 1) @[main.scala 15:24]
      counter <= _T_2 @[main.scala 15:13]
```

Naming

```scala
import chisel3._

class DecCounter extends Module {
  val io = IO(new Bundle{
    val value = Output(UInt(4.W))
  })

  val counter = RegInit(0.U(4.W))

  io.value := counter

  when (counter === 9.U) {
    counter := 0.U
  } otherwise {
    counter := counter + 1.U
  }
}
```

```scala
(new ChiselStage).execute(
  Array(
    "-X", "verilog",
  ),
  Seq(ChiselGeneratorAnnotation(() => new DecCounter))
)
```

```scala
411 def build[T <: RawModule](f: => T): (Circuit, T) = {
... ...
414   val mod = f
... ...
421 }
```

src\main\scala\chisel3\internal\Builder.scala @ 4a0e828

```scala
30 def do_apply[T <: BaseModule](bc: => T)
31                              (...)
32                                   : T = {
..   ...
67   val component = module.generateComponent()
..   ...
76 }
```

src\main\scala\chisel3\Module.scala @ 4a0e828

```scala
61 private[chisel3] override def generateComponent(): Component = {
..   ...
65   val names = nameIds(classOf[RawModule])
..   ...
124 }
```

src\main\scala\chisel3\RawModule.scala @ 4a0e828

```scala
301 protected def nameIds(rootClass: Class[_]): HashMap[HasId, String] = {
...   ...
318   for (m <- getPublicFields(rootClass)) {
319     Builder.nameRecursively(cleanName(m.getName), m.invoke(this), name)
320   }
...   ...
323 }
```

src\main\scala\chisel3\Module.scala @ 4a0e828

Thanks!

Boyang Han
yqszxx@gmail.com