ScalaHDL

by Yao Li

Outline

- Exceptions
- Future Plan

Exceptions

- no such module
- wrong number of arguments
- undeclared registers
- not enough bits length
- illegal argument

Exceptions

- no such module:

```
trait Adder extends ScalaHDL {

  defMod.adder('clk, 'rst, 'a, 'b, 'z) {

    …

  }

}


object Main extends Adder {

  def main(args: Array[String]) {

    …

    println(convert('adder1, clk, rst, a, b, z))

  }

}
```

```
(run-main)
ScalaHDL.Core.NoSuchModuleException: module
adder1 not found!
ScalaHDL.Core.NoSuchModuleException: module
adder1 not found!
at
ScalaHDL.Core.ScalaHDL.convert(SHDLlib.scala:
952)
```

Exceptions

- wrong number of arguments:

```
trait Adder extends ScalaHDL {

  defMod.adder('clk, 'rst, 'a, 'b, 'z) {

    …

  }

}


object Main extends Adder {

  def main(args: Array[String]) {

    …

    println(convert('adder, rst, a, b, z))

  }

}
```

```
[error] (run-main)
ScalaHDL.Core.WrongNumberOfArgumentsException
: wrong number of arguments for module adder,
expecting 5, get 4.
ScalaHDL.Core.WrongNumberOfArgumentsException
: wrong number of arguments for module adder,
expecting 5, get 4.
at
ScalaHDL.Core.HDLModule.mapArgs(SHDLlib.scala
:783)
at
ScalaHDL.Core.HDLModule.convert(SHDLlib.scala
:834)
at
ScalaHDL.Core.ScalaHDL.convert(SHDLlib.scala:
954)
```

Exceptions

- undeclared register:

```
trait Adder extends ScalaHDL {

  defMod.adder('clk, 'rst, 'a, 'b, 'z) {

    'x := 1

    …

  }

}


object Main extends Adder {

  def main(args: Array[String]) {

    …

    println(convert('adder, clk, rst, a, b, z))

  }

}
```

```
[error] (run-main)
ScalaHDL.Core.UndeclaredRegisterException:
register x is not declared!
ScalaHDL.Core.UndeclaredRegisterException:
register x is not declared!
at
ScalaHDL.Core.HDLIdent.<init>(SHDLlib.scala:
423)
at
ScalaHDL.Core.ScalaHDL.toHDLType(SHDLlib.scal
a:913)
at
ScalaHDL.Core.ScalaHDL.sym2HDLType(SHDLlib.sc
ala:851)
```

Exceptions

- not enough bits length:

```
trait Adder extends ScalaHDL {

  defMod.adder('clk, 'rst, 'a, 'b, 'z) {

    'z := 1000

    …

  }

}


object Main extends Adder {

  def main(args: Array[String]) {

    val z = signed(0, 6)

    …

    println(convert('adder, clk, rst, a, b, z))

  }

}
```

```
[error] (run-main)
ScalaHDL.Core.NotEnoughBitsException: not
enough bits to hold value 1000 in z, require
11, get 6
ScalaHDL.Core.NotEnoughBitsException: not
enough bits to hold value 1000 in z, require
11, get 6
at
ScalaHDL.Core.DataType.Signed.checkValid(Sign
al.scala:205)
```

Exceptions

- illegal argument:

```
trait Adder extends ScalaHDL {

  defMod.adder('clk, 'rst, 'a, 'b, 'z) {

    'z := -1

    …

  }

}


object Main extends Adder {

  def main(args: Array[String]) {

    val z = unsigned(0, 6)

    …

    println(convert('adder, clk, rst, a, b, z))

  }

}
```

```
[error] (run-main)
java.lang.IllegalArgumentException: the value
cannot be less than 0
java.lang.IllegalArgumentException: the value
cannot be less than 0
at
ScalaHDL.Core.DataType.Unsigned.checkValid(Si
gnal.scala:132)
```

Future Plan

- log and warning?
- polished syntax
- tests

Any Question?

Thanks!