

Scala

```
val myReady = Bool
foreach { val vecStream = Vec(Stream(Bool), 4)
  vecStream.foreach(_ready := myReady) // Connect to each stream.ready the myReady signal
}
val vecStream1 = Vec(Stream(Bool), 3)
val vecStream2 = Vec(Stream(Bool), 3)
zipped (vecStream1, vecStream2).zipped.foreach(_ >> _) // Connect all Streams of vecStream1 to vecStream2
val myBits = B"00110011"
reduce { val xorBits = myBits.reduce(_ ^ _) // XOR all bits
  val addresses = Vec(UInt(8 bits), 4)
  val key = UInt(8 bits)
  val hits = addresses.map(address => address === key) // hits is a Vector of Bool
}
map { val vecStream = Vec(Stream(Bool), 4)
  val andValid = srcStreams.map(_valid).reduce(_ && _) // AND all valid signals of the Stream together
}
```

Miscellaneous

```
String to Bits { val vecOfBits = Vec("Salut".map(c => B(c.toInt, 8 bits)))
}
val io = new Bundle {
  val pulse = in Bool
  val counter = out UInt(3 bits)
} .setName("")
val area_clkB = new ClockingArea(ClockDomain(io.clkB.io.rstB)) {
  val buf0 = RegNext(area_clkA.reg) init(False)
  addTag(crossClockDomain)
  val buf1 = RegNext(buf0) init(False)
} // Or by using a BufferCC
val area_clkB = new ClockingArea(clkB) {
  val buf1 = BufferCC(area_clkA.reg, False)
}
```