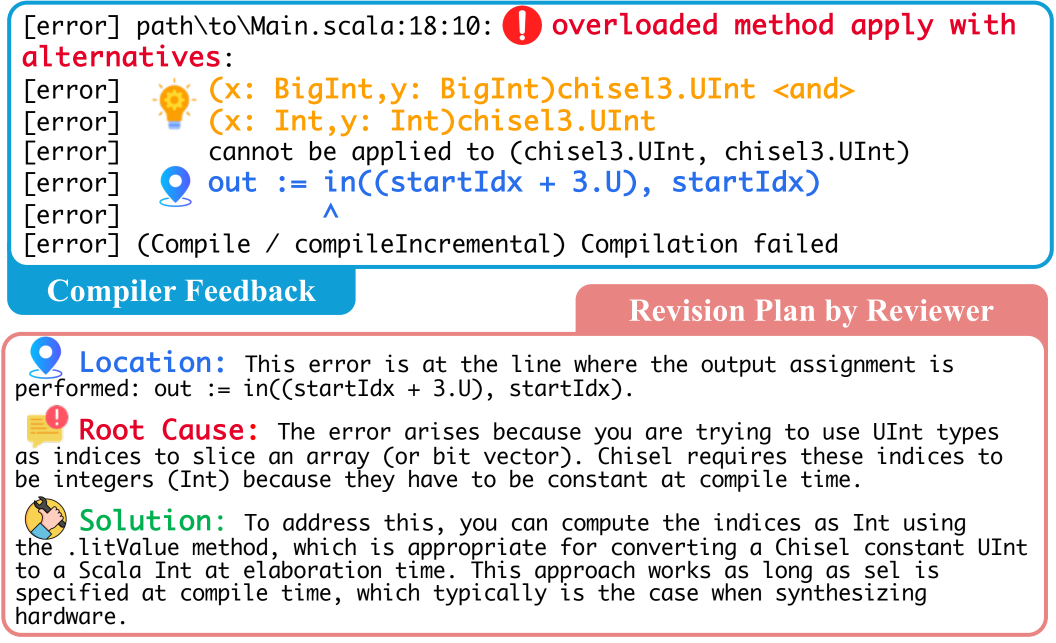

Compiler Feedback

```
[error] path\to\Main.scala:18:10: overloaded method apply with alternatives:
[error]      (x: BigInt,y: BigInt)chisel3.UInt <and>
[error]      (x: Int,y: Int)chisel3.UInt
[error]      cannot be applied to (chisel3.UInt, chisel3.UInt)
[error]      out := in((startIdx + 3.U), startIdx)
[error]             ^
[error] (Compile / compileIncremental) Compilation failed
```

Revision Plan by Reviewer

Location: This error is at the line where the output assignment is performed: out := in((startIdx + 3.U), startIdx).

Root Cause: The error arises because you are trying to use UInt types as indices to slice an array (or bit vector). Chisel requires these indices to be integers (Int) because they have to be constant at compile time.

Solution: To address this, you can compute the indices as Int using the .litValue method, which is appropriate for converting a Chisel constant UInt to a Scala Int at elaboration time. This approach works as long as sel is specified at compile time, which typically is the case when synthesizing hardware.