



`current_block`

```
...  
cond := Assign ...
```

BB0

Ablauf der Codeerzeugung:

- ▶ Bedingung generieren
`self.rvalue(ifStmt.cond)`

```
...  
cond := Assign ...
```

BB0

BB1

BB2

BB3

Ablauf der Codeerzeugung:

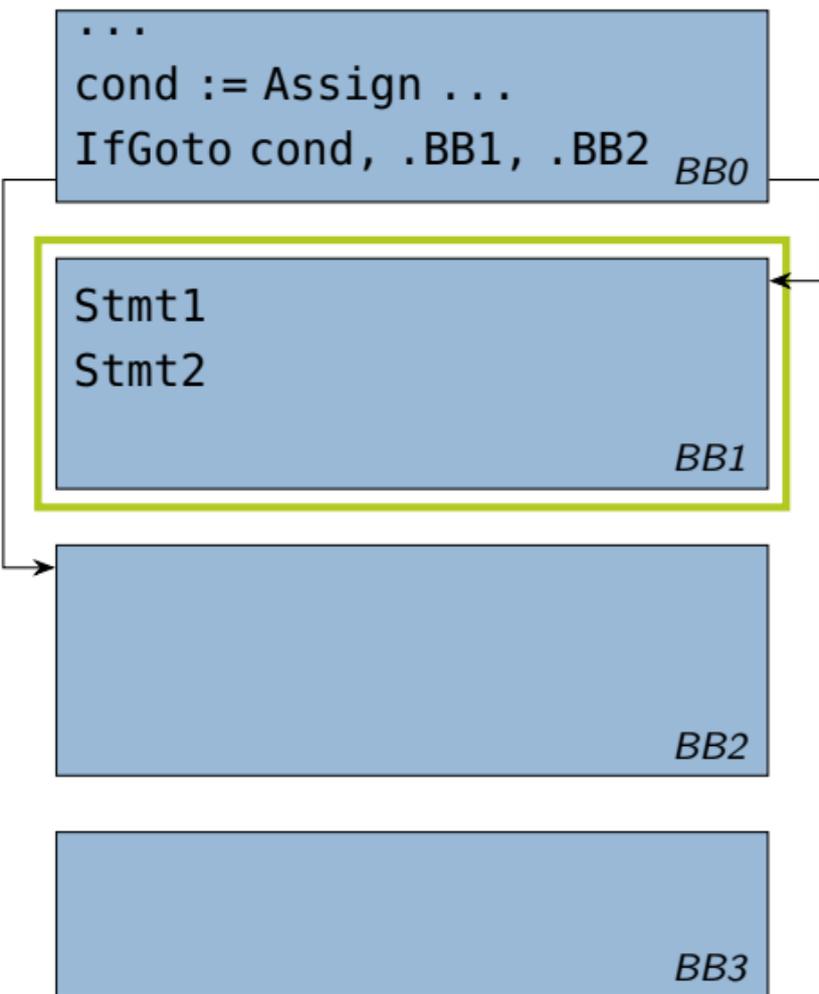
- ▶ Bedingung generieren
`self.rvalue(ifStmt.cond)`
- ▶ Blöcke für Then-/Else-Teil, sowie den Sequenzierungsblock erstellen

```
...  
cond := Assign ...  
IfGoto cond, .BB1, .BB2 BB0
```

```
graph TD; BB0[BB0] --> BB1[BB1]; BB0 --> BB2[BB2]; BB1 --> BB2; BB2 --> BB3[BB3];
```

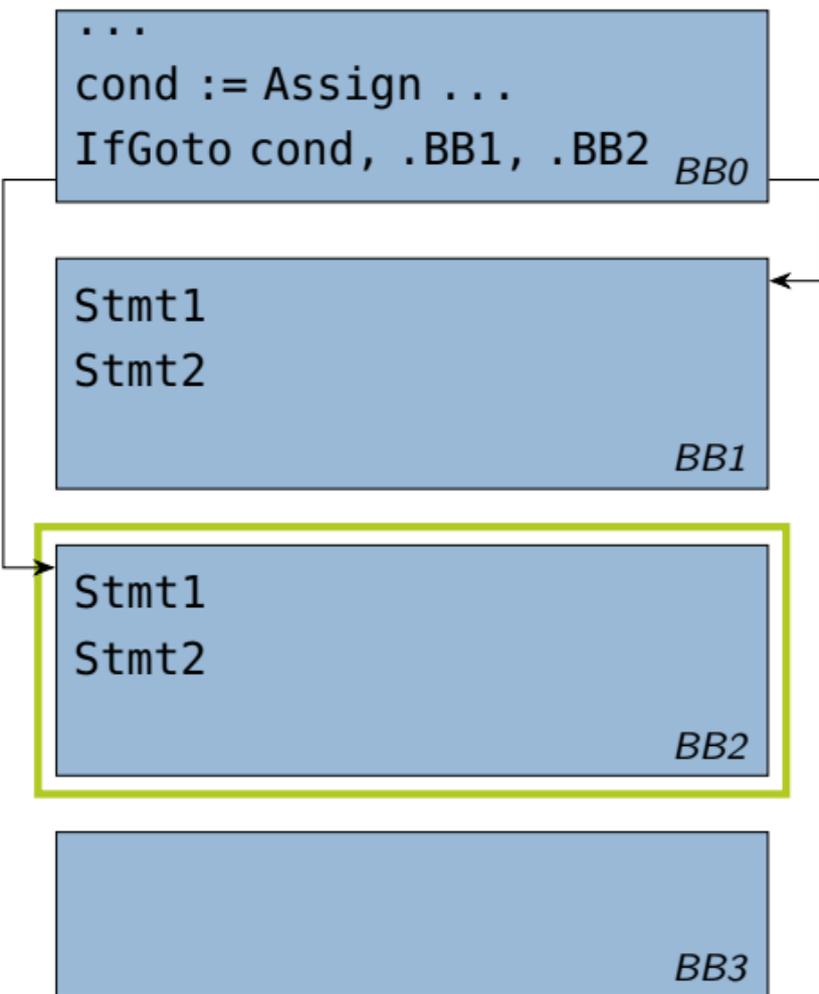
Ablauf der Codeerzeugung:

- ▶ Bedingung generieren
`self.rvalue(ifStmt.cond)`
- ▶ Blöcke für Then-/Else-Teil, sowie den Sequenzierungsblock erstellen
- ▶ Bedingte Kontrollflussverzeigung



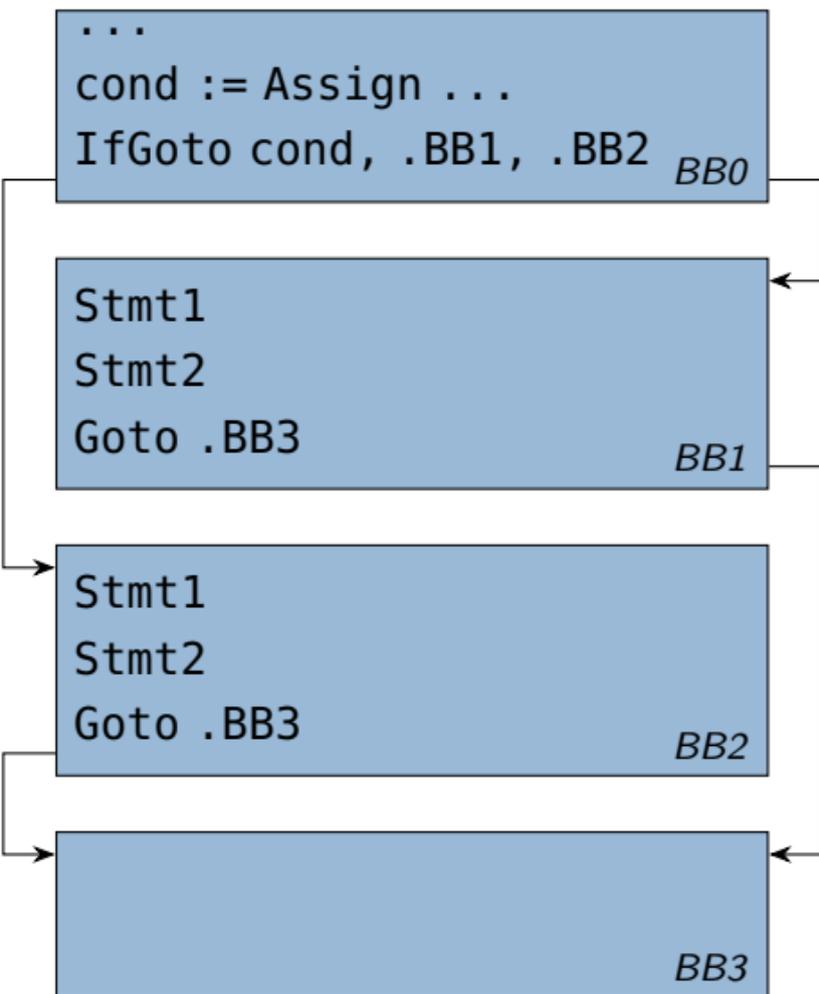
Ablauf der Codeerzeugung:

- ▶ Bedingung generieren
`self.rvalue(ifStmt.cond)`
- ▶ Blöcke für Then-/Else-Teil, sowie den Sequenzierungsblock erstellen
- ▶ Bedingte Kontrollflussverzeigung
- ▶ Then-Block generieren
`self.visit(ifStmt.then_block)`



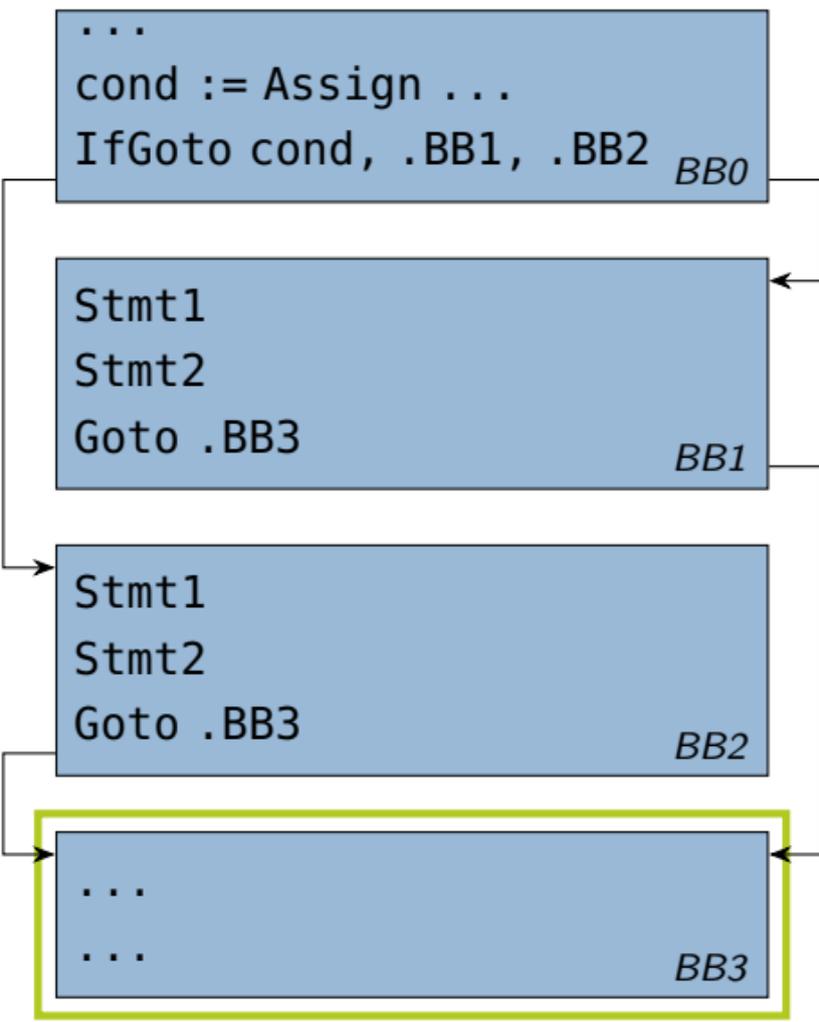
Ablauf der Codeerzeugung:

- ▶ Bedingung generieren
`self.rvalue(ifStmt.cond)`
- ▶ Blöcke für Then-/Else-Teil, sowie den Sequenzierungsblock erstellen
- ▶ Bedingte Kontrollflussverzeigung
- ▶ Then-Block generieren
`self.visit(ifStmt.then_block)`
- ▶ Else-Block generieren
`self.visit(ifStmt.else_block)`



Ablauf der Codeerzeugung:

- ▶ Bedingung generieren
`self.rvalue(ifStmt.cond)`
- ▶ Blöcke für Then-/Else-Teil, sowie den Sequenzierungsblock erstellen
- ▶ Bedingte Kontrollflussverzeigung
- ▶ Then-Block generieren
`self.visit(ifStmt.then_block)`
- ▶ Else-Block generieren
`self.visit(ifStmt.else_block)`
- ▶ Kontrollfluss wieder zusammenführen



Ablauf der Codeerzeugung:

- ▶ Bedingung generieren
`self.rvalue(ifStmt.cond)`
- ▶ Blöcke für Then-/Else-Teil, sowie den Sequenzierungsblock erstellen
- ▶ Bedingte Kontrollflussverzeigung
- ▶ Then-Block generieren
`self.visit(ifStmt.then_block)`
- ▶ Else-Block generieren
`self.visit(ifStmt.else_block)`
- ▶ Kontrollfluss wieder zusammenführen
- ▶ `current_block`-Invariante wiederherstellen