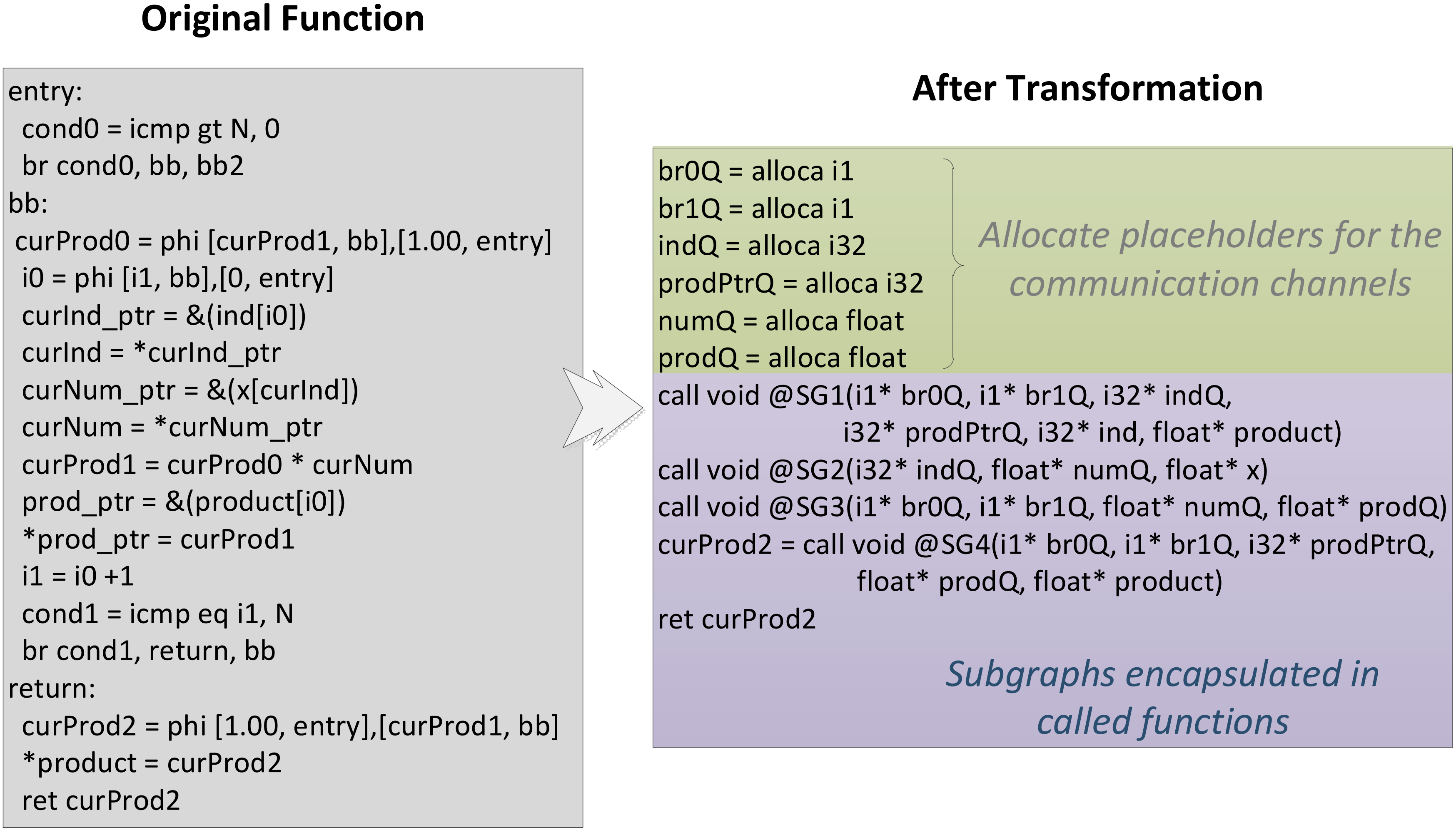

Original Function

```
entry:
  cond0 = icmp gt N, 0
  br cond0, bb, bb2
bb:
 curProd0 = phi [curProd1, bb],[1.00, entry]
  i0 = phi [i1, bb],[0, entry]
  curInd_ptr = &(ind[i0])
  curInd = *curInd_ptr
  curNum_ptr = &(x[curInd])
  curNum = *curNum_ptr
  curProd1 = curProd0 * curNum
  prod_ptr = &(product[i0])
  *prod_ptr = curProd1
  i1 = i0 +1
  cond1 = icmp eq i1, N
  br cond1, return, bb
return:
  curProd2 = phi [1.00, entry],[curProd1, bb]
  *product = curProd2
  ret curProd2
```

After Transformation

```
br0Q = alloca i1
br1Q = alloca i1
indQ = alloca i32
prodPtrQ = alloca i32
numQ = alloca float
prodQ = alloca float
```

Allocate placeholders for the communication channels

```
call void @SG1(i1* br0Q, i1* br1Q, i32* indQ,
                     i32* prodPtrQ, i32* ind, float* product)
call void @SG2(i32* indQ, float* numQ, float* x)
call void @SG3(i1* br0Q, i1* br1Q, float* numQ, float* prodQ)
curProd2 = call void @SG4(i1* br0Q, i1* br1Q, i32* prodPtrQ,
                     float* prodQ, float* product)
ret curProd2
```

Subgraphs encapsulated in called functions