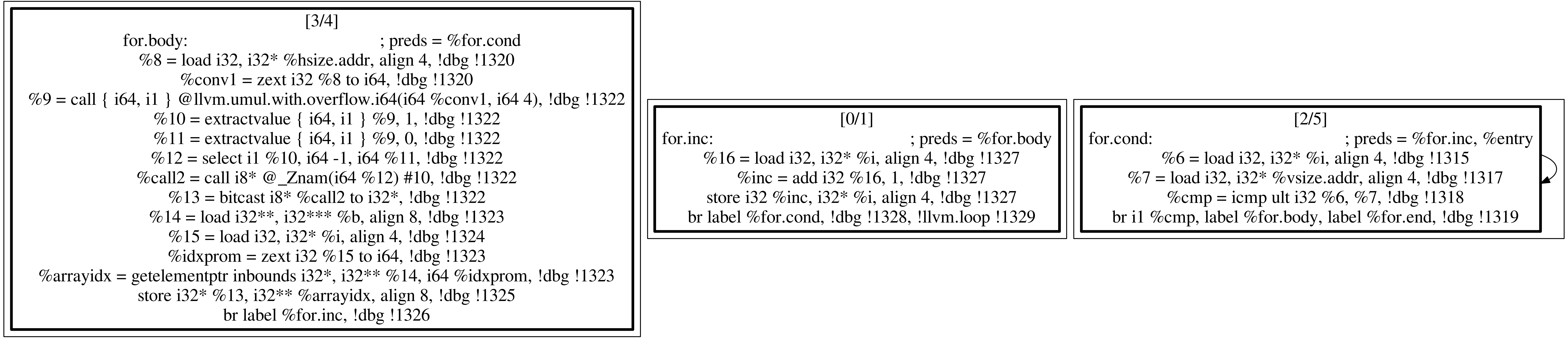

```
[3/4]
for.body:                                         ; preds = %for.cond
  %8 = load i32, i32* %hsize.addr, align 4, !dbg !1320
  %conv1 = zext i32 %8 to i64, !dbg !1320
  %9 = call { i64, i1 } @llvm.umul.with.overflow.i64(i64 %conv1, i64 4), !dbg !1322
  %10 = extractvalue { i64, i1 } %9, 1, !dbg !1322
  %11 = extractvalue { i64, i1 } %9, 0, !dbg !1322
  %12 = select i1 %10, i64 -1, i64 %11, !dbg !1322
  %call2 = call i8* @_Znam(i64 %12) #10, !dbg !1322
  %13 = bitcast i8* %call2 to i32*, !dbg !1322
  %14 = load i32**, i32*** %b, align 8, !dbg !1323
  %15 = load i32, i32* %i, align 4, !dbg !1324
  %idxprom = zext i32 %15 to i64, !dbg !1323
  %arrayidx = getelementptr inbounds i32*, i32** %14, i64 %idxprom, !dbg !1323
  store i32* %13, i32** %arrayidx, align 8, !dbg !1325
  br label %for.inc, !dbg !1326
```

```
[0/1]
for.inc:                                          ; preds = %for.body
  %16 = load i32, i32* %i, align 4, !dbg !1327
  %inc = add i32 %16, 1, !dbg !1327
  store i32 %inc, i32* %i, align 4, !dbg !1327
  br label %for.cond, !dbg !1328, !llvm.loop !1329
```

```
[2/5]
for.cond:                                         ; preds = %for.inc, %entry
  %6 = load i32, i32* %i, align 4, !dbg !1315
  %7 = load i32, i32* %vsize.addr, align 4, !dbg !1317
  %cmp = icmp ult i32 %6, %7, !dbg !1318
  br i1 %cmp, label %for.body, label %for.end, !dbg !1319
```