```
entry:
 %cmp = icmp slt i32 %bits_left, %min_bits
 br i1 %cmp, label %if.then, label %if.end4
```

```
if.then:
 %call = tail call i32 @jpeg_fill_bit_buffer(%struct.bitread_working_state*
... %state, i64 %get_buffer, i32 %bits_left, i32 %min_bits)
 %tobool = icmp eq i32 %call, 0
 br i1 %tobool, label %cleanup, label %if.end
```

```
if.end:
 %get_buffer2 = getelementptr inbounds %struct.bitread_working_state,
... %struct.bitread_working_state* %state, i64 0, i32 3
 %0 = load i64, i64* %get_buffer2, align 8, !tbaa !2
 %bits_left3 = getelementptr inbounds %struct.bitread_working_state,
... %struct.bitread_working_state* %state, i64 0, i32 4
 %1 = load i32, i32* %bits_left3, align 8, !tbaa !9
 br label %if.end4
```

```
if.end4:
 %bits_left.addr.0 = phi i32 [ %1, %if.end ], [ %bits_left, %entry ]
 %get_buffer.addr.0 = phi i64 [ %0, %if.end ], [ %get_buffer, %entry ]
 %sub = sub nsw i32 %bits_left.addr.0, %min_bits
 %sh_prom = zext i32 %sub to i64
 %shr = ashr i64 %get_buffer.addr.0, %sh_prom
 %conv = trunc i64 %shr to i32
 %shl = shl i32 1, %min_bits
 %sub5 = add nsw i32 %shl, -1
 %and = and i32 %conv, %sub5
 %conv6 = sext i32 %and to i64
 %idxprom88 = sext i32 %min_bits to i64
 %arrayidx89 = getelementptr inbounds %struct.d_derived_tbl,
... %struct.d_derived_tbl* %htbl, i64 0, i32 1, i64 %idxprom88
 %2 = load i64, i64* %arrayidx89, align 8, !tbaa !10
 %cmp790 = icmp sgt i64 %conv6, %2
 %get_buffer17 = getelementptr inbounds %struct.bitread_working_state,
... %struct.bitread_working_state* %state, i64 0, i32 3
 %bits_left18 = getelementptr inbounds %struct.bitread_working_state,
... %struct.bitread_working_state* %state, i64 0, i32 4
 br i1 %cmp790, label %while.body.preheader, label %while.end
```

```
while.body.preheader:
 br label %while.body
```

```
while.body:
 %code.094 = phi i64 [ %or, %if.end19 ], [ %conv6, %while.body.preheader ]
 %l.093 = phi i32 [ %inc, %if.end19 ], [ %min_bits, %while.body.preheader ]
 %get_buffer.addr.192 = phi i64 [ %get_buffer.addr.2, %if.end19 ], [
... %get_buffer.addr.0, %while.body.preheader ]
 %bits_left.addr.191 = phi i32 [ %sub20, %if.end19 ], [ %sub,
... %while.body.preheader ]
 %shl9 = shl i64 %code.094, 1
 %cmp10 = icmp slt i32 %bits_left.addr.191, 1
 br i1 %cmp10, label %if.then12, label %if.end19
```

```
if.then12:
 %call13 = tail call i32 @jpeg_fill_bit_buffer(%struct.bitread_working_state*
... %state, i64 %get_buffer.addr.192, i32 %bits_left.addr.191, i32 1)
 %tobool14 = icmp eq i32 %call13, 0
 br i1 %tobool14, label %cleanup.loopexit, label %if.end16
```

```
if.end16:
 %3 = load i64, i64* %get_buffer17, align 8, !tbaa !2
 %4 = load i32, i32* %bits_left18, align 8, !tbaa !9
 br label %if.end19
```

```
cleanup.loopexit:
 br label %cleanup
```

```
if.end19:
 %bits_left.addr.2 = phi i32 [ %4, %if.end16 ], [ %bits_left.addr.191,
... %while.body ]
 %get_buffer.addr.2 = phi i64 [ %3, %if.end16 ], [ %get_buffer.addr.192,
... %while.body ]
 %sub20 = add nsw i32 %bits_left.addr.2, -1
 %sh_prom21 = zext i32 %sub20 to i64
 %shr2281 = lshr i64 %get_buffer.addr.2, %sh_prom21
 %and24 = and i64 %shr2281, 1
 %or = or i64 %and24, %shl9
 %inc = add nsw i32 %l.093, 1
 %idxprom = sext i32 %inc to i64
 %arrayidx = getelementptr inbounds %struct.d_derived_tbl,
... %struct.d_derived_tbl* %htbl, i64 0, i32 1, i64 %idxprom
 %5 = load i64, i64* %arrayidx, align 8, !tbaa !10
 %cmp7 = icmp sgt i64 %or, %5
 br i1 %cmp7, label %while.body, label %while.end.loopexit
```

```
while.end.loopexit:
 %idxprom.lcssa104 = phi i64 [ %idxprom, %if.end19 ]
 %inc.lcssa = phi i32 [ %inc, %if.end19 ]
 %or.lcssa = phi i64 [ %or, %if.end19 ]
 %sub20.lcssa = phi i32 [ %sub20, %if.end19 ]
 %get_buffer.addr.2.lcssa = phi i64 [ %get_buffer.addr.2, %if.end19 ]
 br label %while.end
```

```
while.end:
 %idxprom.lcssa = phi i64 [ %idxprom88, %if.end4 ], [ %idxprom.lcssa104,
... %while.end.loopexit ]
 %code.0.lcssa = phi i64 [ %conv6, %if.end4 ], [ %or.lcssa,
... %while.end.loopexit ]
 %l.0.lcssa = phi i32 [ %min_bits, %if.end4 ], [ %inc.lcssa,
... %while.end.loopexit ]
 %get_buffer.addr.1.lcssa = phi i64 [ %get_buffer.addr.0, %if.end4 ], [
... %get_buffer.addr.2.lcssa, %while.end.loopexit ]
 %bits_left.addr.1.lcssa = phi i32 [ %sub, %if.end4 ], [ %sub20.lcssa,
... %while.end.loopexit ]
 store i64 %get_buffer.addr.1.lcssa, i64* %get_buffer17, align 8, !tbaa !2
 store i32 %bits_left.addr.1.lcssa, i32* %bits_left18, align 8, !tbaa !9
 %cmp28 = icmp sgt i32 %l.0.lcssa, 16
 br i1 %cmp28, label %if.then30, label %if.end34
```

```
if.then30:
 %cinfo = getelementptr inbounds %struct.bitread_working_state,
... %struct.bitread_working_state* %state, i64 0, i32 5
 %6 = load %struct.jpeg_decompress_struct*, %struct.jpeg_decompress_struct**
... %cinfo, align 8, !tbaa !11
 %err = getelementptr inbounds %struct.jpeg_decompress_struct,
... %struct.jpeg_decompress_struct* %6, i64 0, i32 0
 %7 = load %struct.jpeg_error_mgr*, %struct.jpeg_error_mgr** %err, align 8,
... !tbaa !12
 %msg_code = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %7, i64 0, i32 5
 store i32 114, i32* %msg_code, align 8, !tbaa !16
 %emit_message = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %7, i64 0, i32 1
 %8 = load void (%struct.jpeg_common_struct*, i32)*, void
... (%struct.jpeg_common_struct*, i32)** %emit_message, align 8, !tbaa !18
 %9 = bitcast %struct.jpeg_decompress_struct* %6 to
... %struct.jpeg_common_struct*
 tail call void %8(%struct.jpeg_common_struct* %9, i32 -1) #2
 br label %cleanup
```

```
if.end34:
 %arrayidx36 = getelementptr inbounds %struct.d_derived_tbl,
... %struct.d_derived_tbl* %htbl, i64 0, i32 2, i64 %idxprom.lcssa
 %10 = load i32, i32* %arrayidx36, align 4, !tbaa !19
 %arrayidx38 = getelementptr inbounds %struct.d_derived_tbl,
... %struct.d_derived_tbl* %htbl, i64 0, i32 0, i64 %idxprom.lcssa
 %11 = load i64, i64* %arrayidx38, align 8, !tbaa !10
 %sub39 = sub nsw i64 %code.0.lcssa, %11
 %conv40 = trunc i64 %sub39 to i32
 %add = add nsw i32 %conv40, %10
 %idxprom41 = sext i32 %add to i64
 %pub = getelementptr inbounds %struct.d_derived_tbl, %struct.d_derived_tbl*
... %htbl, i64 0, i32 3
 %12 = load %struct.JHUFF_TBL*, %struct.JHUFF_TBL** %pub, align 8, !tbaa !20
 %arrayidx42 = getelementptr inbounds %struct.JHUFF_TBL, %struct.JHUFF_TBL*
... %12, i64 0, i32 1, i64 %idxprom41
 %13 = load i8, i8* %arrayidx42, align 1, !tbaa !22
 %conv43 = zext i8 %13 to i32
 br label %cleanup
```

```
cleanup:
 %retval.0 = phi i32 [ 0, %if.then30 ], [ %conv43, %if.end34 ], [ -1,
... %if.then ], [ -1, %cleanup.loopexit ]
 ret i32 %retval.0
```

CFG for 'jpeg_huff_decode' function