T
F
T
F
def
1
2
3
T
F

```
entry:
 %unread_marker = getelementptr inbounds %struct.jpeg_decompress_struct,
... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 72
 %0 = load i32, i32* %unread_marker, align 4, !tbaa !2
 %err = getelementptr inbounds %struct.jpeg_decompress_struct,
... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 0
 %1 = load %struct.jpeg_error_mgr*, %struct.jpeg_error_mgr** %err, align 8,
... !tbaa !10
 %msg_code = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %1, i64 0, i32 5
 store i32 117, i32* %msg_code, align 8, !tbaa !11
 %arrayidx = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %1, i64 0, i32 6, i32 0, i64 0
 store i32 %0, i32* %arrayidx, align 4, !tbaa !14
 %arrayidx5 = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %1, i64 0, i32 6, i32 0, i64 1
 store i32 %desired, i32* %arrayidx5, align 4, !tbaa !14
 %emit_message = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %1, i64 0, i32 1
 %2 = load void (%struct.jpeg_common_struct*, i32)*, void
... (%struct.jpeg_common_struct*, i32)** %emit_message, align 8, !tbaa !15
 %3 = bitcast %struct.jpeg_decompress_struct* %cinfo to
... %struct.jpeg_common_struct*
 tail call void %2(%struct.jpeg_common_struct* %3, i32 -1) #4
 %add = add nsw i32 %desired, 1
 %and = and i32 %add, 7
 %add11 = or i32 %and, 208
 %add14 = add nsw i32 %desired, 2
 %and15 = and i32 %add14, 7
 %add16 = or i32 %and15, 208
 %sub = add i32 %desired, 7
 %and20 = and i32 %sub, 7
 %add21 = or i32 %and20, 208
 %sub24 = add i32 %desired, 6
 %and25 = and i32 %sub24, 7
 %add26 = or i32 %and25, 208
 br label %for.cond.outer

for.cond.outer:
 %marker.0.ph = phi i32 [ %13, %if.end48 ], [ %0, %entry ]
 %cmp = icmp slt i32 %marker.0.ph, 192
 %cmp12 = icmp eq i32 %marker.0.ph, %add11
 %cmp17 = icmp eq i32 %marker.0.ph, %add16
 %cmp22 = icmp eq i32 %marker.0.ph, %add21
 %cmp27 = icmp eq i32 %marker.0.ph, %add26
 br i1 %cmp, label %for.cond.outer.split.us, label %for.cond.outer.split

for.cond.outer.split:
 %6 = and i32 %marker.0.ph, -8
 %7 = icmp eq i32 %6, 208
 br i1 %7, label %if.else10.us83.preheader, label %for.cond.outer.split.split

if.else10.us83.preheader:
 %brmerge = or i1 %cmp12, %cmp17
 %8 = or i1 %cmp22, %cmp27
 %.. = select i1 %8, i32 2, i32 1
 %action.0.us88 = select i1 %brmerge, i32 3, i32 %..
 br label %if.else10.us83

if.else10.us83:
 %9 = load %struct.jpeg_error_mgr*, %struct.jpeg_error_mgr** %err, align 8,
... !tbaa !10
 %msg_code34.us89 = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %9, i64 0, i32 5
 store i32 96, i32* %msg_code34.us89, align 8, !tbaa !11
 %arrayidx38.us90 = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %9, i64 0, i32 6, i32 0, i64 0
 store i32 %marker.0.ph, i32* %arrayidx38.us90, align 4, !tbaa !14
 %arrayidx42.us91 = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %9, i64 0, i32 6, i32 0, i64 1
 store i32 %action.0.us88, i32* %arrayidx42.us91, align 4, !tbaa !14
 %emit_message44.us92 = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %9, i64 0, i32 1
 %10 = load void (%struct.jpeg_common_struct*, i32)*, void
... (%struct.jpeg_common_struct*, i32)** %emit_message44.us92, align 8, !tbaa !15
 tail call void %10(%struct.jpeg_common_struct* %3, i32 4) #4
 switch i32 %action.0.us88, label %if.else10.us83 [
 i32 1, label %sw.bb
 i32 2, label %sw.bb46.loopexit
 i32 3, label %cleanup.loopexit
 ]

for.cond.outer.split.us:
 %4 = load %struct.jpeg_error_mgr*, %struct.jpeg_error_mgr** %err, align 8,
... !tbaa !10
 %msg_code34.us.us = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %4, i64 0, i32 5
 store i32 96, i32* %msg_code34.us.us, align 8, !tbaa !11
 %arrayidx38.us.us = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %4, i64 0, i32 6, i32 0, i64 0
 store i32 %marker.0.ph, i32* %arrayidx38.us.us, align 4, !tbaa !14
 %arrayidx42.us.us = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %4, i64 0, i32 6, i32 0, i64 1
 store i32 2, i32* %arrayidx42.us.us, align 4, !tbaa !14
 %emit_message44.us.us = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %4, i64 0, i32 1
 %5 = load void (%struct.jpeg_common_struct*, i32)*, void
... (%struct.jpeg_common_struct*, i32)** %emit_message44.us.us, align 8, !tbaa !15
 tail call void %5(%struct.jpeg_common_struct* %3, i32 4) #4
 br label %sw.bb46

sw.bb46.loopexit:
 br label %sw.bb46

cleanup.loopexit:
 br label %cleanup

for.cond.outer.split.split:
 %marker.0.ph.lcssa = phi i32 [ %marker.0.ph, %for.cond.outer.split ]
 %11 = load %struct.jpeg_error_mgr*, %struct.jpeg_error_mgr** %err, align 8,
... !tbaa !10
 %msg_code34.us101 = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %11, i64 0, i32 5
 store i32 96, i32* %msg_code34.us101, align 8, !tbaa !11
 %arrayidx38.us102 = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %11, i64 0, i32 6, i32 0, i64 0
 store i32 %marker.0.ph.lcssa, i32* %arrayidx38.us102, align 4, !tbaa !14
 %arrayidx42.us103 = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %11, i64 0, i32 6, i32 0, i64 1
 store i32 3, i32* %arrayidx42.us103, align 4, !tbaa !14
 %emit_message44.us104 = getelementptr inbounds %struct.jpeg_error_mgr,
... %struct.jpeg_error_mgr* %11, i64 0, i32 1
 %12 = load void (%struct.jpeg_common_struct*, i32)*, void
... (%struct.jpeg_common_struct*, i32)** %emit_message44.us104, align 8, !tbaa !15
 tail call void %12(%struct.jpeg_common_struct* %3, i32 4) #4
 br label %cleanup

sw.bb46:
 %call = tail call fastcc i32 @next_marker(%struct.jpeg_decompress_struct*
... nonnull %cinfo)
 %tobool = icmp eq i32 %call, 0
 br i1 %tobool, label %cleanup.loopexit129, label %if.end48

sw.bb:
 store i32 0, i32* %unread_marker, align 4, !tbaa !2
 br label %cleanup

cleanup.loopexit129:
 br label %cleanup

if.end48:
 %13 = load i32, i32* %unread_marker, align 4, !tbaa !2
 br label %for.cond.outer

cleanup:
 %retval.0 = phi i32 [ 1, %sw.bb ], [ 1, %for.cond.outer.split.split ], [ 1,
... %cleanup.loopexit ], [ 0, %cleanup.loopexit129 ]
 ret i32 %retval.0
```

CFG for 'jpeg_resync_to_restart' function