

```

entry:
  %br_state = alloca %struct.bitread_working_state, align 16
  %entropy1 = getelementptr inbounds %struct.jpeg_decompress_struct,
  ... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 79
  %0 = bitcast %struct.jpeg_entropy_decoder** %entropy1 to
  ... %struct.phuff_entropy_decoder**
  %1 = load %struct.phuff_entropy_decoder*, %struct.phuff_entropy_decoder**
  ... %0, align 8, !tbaa !2
  %A1 = getelementptr inbounds %struct.jpeg_decompress_struct,
  ... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 71
  %2 = bitcast i32* %A1 to i64*
  %3 = load i64, i64* %2, align 8
  %4 = trunc i64 %3 to i32
  %sh1 = shl i32 1, %4
  %5 = bitcast %struct.bitread_working_state* %br_state to i8*
  call void @llvm.lifetime.start(i64 56, i8* %5) #3
  %restart_interval = getelementptr inbounds %struct.jpeg_decompress_struct,
  ... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 49
  %6 = load i32, i32* %restart_interval, align 8, !tbaa !10
  %tobool = icmp eq i32 %6, 0
  %7 = lshr i64 %3, 32
  %8 = trunc i64 %7 to i32
  br i1 %tobool, label %if.end6, label %if.then

```

```

if.then:
  %restarts_to_go = getelementptr inbounds %struct.phuff_entropy_decoder,
  ... %struct.phuff_entropy_decoder* %1, i64 0, i32 3
  %9 = load i32, i32* %restarts_to_go, align 4, !tbaa !11
  %cmp = icmp eq i32 %9, 0
  br i1 %cmp, label %if.then2, label %if.end6

```

```

if.then2:
  %bits_left.i = getelementptr inbounds %struct.phuff_entropy_decoder,
  ... %struct.phuff_entropy_decoder* %1, i64 0, i32 1, i32 1
  %10 = load i32, i32* %bits_left.i, align 8, !tbaa !17
  %div.i = sdiv i32 %10, 8
  %marker.i = getelementptr inbounds %struct.jpeg_decompress_struct,
  ... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 78
  %11 = load %struct.jpeg_marker_reader*, %struct.jpeg_marker_reader**
  ... %marker.i, align 8, !tbaa !18
  %discarded_bytes.i = getelementptr inbounds %struct.jpeg_marker_reader,
  ... %struct.jpeg_marker_reader* %11, i64 0, i32 8
  %12 = load i32, i32* %discarded_bytes.i, align 4, !tbaa !19
  %add.i = add i32 %12, %div.i
  store i32 %add.i, i32* %discarded_bytes.i, align 4, !tbaa !19
  store i32 0, i32* %bits_left.i, align 8, !tbaa !17
  %read_restart_marker.i = getelementptr inbounds %struct.jpeg_marker_reader,
  ... %struct.jpeg_marker_reader* %11, i64 0, i32 2
  %13 = load i32 (%struct.jpeg_decompress_struct*), i32
  ... (%struct.jpeg_decompress_struct*)* %read_restart_marker.i, align 8, !tbaa !21
  %call.i = tail call i32 @13(%struct.jpeg_decompress_struct* nonnull %cinfo)
  ... #3
  %tobool.i = icmp eq i32 %call.i, 0
  br i1 %tobool.i, label %cleanup, label %for.cond.preheader.i

```

```

for.cond.preheader.i:
  %comps_in_scan.i = getelementptr inbounds %struct.jpeg_decompress_struct,
  ... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 62
  %14 = load i32, i32* %comps_in_scan.i, align 8, !tbaa !22
  %cmp22.i = icmp sgt i32 %14, 0
  br i1 %cmp22.i, label %for.body.i.preheader, label %process_restart.exit

```

```

for.body.i.preheader:
  br label %for.body.i

```

```

for.body.i:
  %indvars.iv.i = phi i64 [ %indvars.iv.next.i, %for.body.i ], [ 0,
  ... %for.body.i.preheader ]
  %arrayidx.i = getelementptr inbounds %struct.phuff_entropy_decoder,
  ... %struct.phuff_entropy_decoder* %1, i64 0, i32 2, i32 1, i64 %indvars.iv.i
  store i32 0, i32* %arrayidx.i, align 4, !tbaa !23
  %indvars.iv.next.i = add nuw nsw i64 %indvars.iv.i, 1
  %15 = load i32, i32* %comps_in_scan.i, align 8, !tbaa !22
  %16 = sext i32 %15 to i64
  %cmp.i = icmp slt i64 %indvars.iv.next.i, %16
  br i1 %cmp.i, label %for.body.i, label %process_restart.exit.loopexit

```

```

process_restart.exit.loopexit:
  br label %process_restart.exit

```

```

process_restart.exit:
  %EOBRUN.i = getelementptr inbounds %struct.phuff_entropy_decoder,
  ... %struct.phuff_entropy_decoder* %1, i64 0, i32 2, i32 0
  store i32 0, i32* %EOBRUN.i, align 8, !tbaa !24
  %17 = load i32, i32* %restart_interval, align 8, !tbaa !10
  store i32 %17, i32* %restarts_to_go, align 4, !tbaa !11
  %printed_eod.i = getelementptr inbounds %struct.phuff_entropy_decoder,
  ... %struct.phuff_entropy_decoder* %1, i64 0, i32 1, i32 2
  store i32 0, i32* %printed_eod.i, align 4, !tbaa !25
  %unread_marker.ph1.trans.insert = getelementptr inbounds
  ... %struct.jpeg_decompress_struct, %struct.jpeg_decompress_struct* %cinfo, i64
  ... 0, i32 72
  %pre = load i32, i32* %unread_marker.ph1.trans.insert, align 4, !tbaa !26
  br label %if.end6

```

```

if.end6:
  %18 = phi i32 [ %pre, %process_restart.exit ], [ %8, %entry ], [ %8,
  ... %if.then ]
  %cinfo7 = getelementptr inbounds %struct.bitread_working_state,
  ... %struct.bitread_working_state* %br_state, i64 0, i32 5
  store %struct.jpeg_decompress_struct* %cinfo7, align 8, !tbaa !27
  %src = getelementptr inbounds %struct.jpeg_decompress_struct,
  ... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 5
  %19 = bitcast %struct.jpeg_source_mgr** %src to i64**
  %20 = load i64*, i64** %19, align 8, !tbaa !29
  %21 = bitcast i64* %20 to <2 x i64>*
  %22 = load <2 x i64>, <2 x i64>* %21, align 8, !tbaa !30
  %23 = bitcast %struct.bitread_working_state* %br_state to <2 x i64>*
  store <2 x i64> %22, <2 x i64>* %23, align 16, !tbaa !30
  %unread_marker = getelementptr inbounds %struct.jpeg_decompress_struct,
  ... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 72
  %unread_marker11 = getelementptr inbounds %struct.bitread_working_state,
  ... %struct.bitread_working_state* %br_state, i64 0, i32 2
  store i32 %18, i32* %unread_marker11, align 16, !tbaa !31
  %get_buffer12 = getelementptr inbounds %struct.phuff_entropy_decoder,
  ... %struct.phuff_entropy_decoder* %1, i64 0, i32 1, i32 0
  %24 = load i64, i64* %get_buffer12, align 8, !tbaa !32
  %bits_left14 = getelementptr inbounds %struct.phuff_entropy_decoder,
  ... %struct.phuff_entropy_decoder* %1, i64 0, i32 1, i32 1
  %25 = load i32, i32* %bits_left14, align 8, !tbaa !17
  %printed_eod = getelementptr inbounds %struct.phuff_entropy_decoder,
  ... %struct.phuff_entropy_decoder* %1, i64 0, i32 1, i32 2
  %printed_eod_ptr = getelementptr inbounds %struct.bitread_working_state,
  ... %struct.bitread_working_state* %br_state, i64 0, i32 6
  store i32* %printed_eod, i32** %printed_eod_ptr, align 16, !tbaa !33
  %blocks_in_MCU = getelementptr inbounds %struct.jpeg_decompress_struct,
  ... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 66
  %26 = load i32, i32* %blocks_in_MCU, align 8, !tbaa !34
  %cmp1687 = icmp sgt i32 %26, 0
  br i1 %cmp1687, label %for.body.lr.ph, label %for.end

```

```

for.body.lr.ph:
  %get_buffer23 = getelementptr inbounds %struct.bitread_working_state,
  ... %struct.bitread_working_state* %br_state, i64 0, i32 3
  %bits_left24 = getelementptr inbounds %struct.bitread_working_state,
  ... %struct.bitread_working_state* %br_state, i64 0, i32 4
  br label %for.body

```

```

for.body:
  %indvars.iv = phi i64 [ 0, %for.body.lr.ph ], [ %indvars.iv.next, %for.inc ]
  %bits_left.090 = phi i32 [ %25, %for.body.lr.ph ], [ %sub, %for.inc ]
  %get_buffer.089 = phi i64 [ %24, %for.body.lr.ph ], [ %get_buffer.1,
  ... %for.inc ]
  %arrayidx = getelementptr inbounds [64 x i16]*, [64 x i16]** %MCU_data, i64
  ... %indvars.iv
  %27 = load [64 x i16]*, [64 x i16]** %arrayidx, align 8, !tbaa !35
  %cmp17 = icmp slt i32 %bits_left.090, 1
  br i1 %cmp17, label %if.then18, label %if.end25

```

```

if.then18:
  %call119 = call i32 @jpeg_fill_bit_buffer(%struct.bitread_working_state*
  ... nonnull %br_state, i64 %get_buffer.089, i32 %bits_left.090, i32 1) #3
  %tobool20 = icmp eq i32 %call119, 0
  br i1 %tobool20, label %cleanup.loopexit, label %if.end22

```

```

if.end22:
  %28 = load i64, i64* %get_buffer23, align 8, !tbaa !36
  %29 = load i32, i32* %bits_left24, align 16, !tbaa !37
  br label %if.end25

```

```

if.end25:
  %get_buffer.1 = phi i64 [ %28, %if.end22 ], [ %get_buffer.089, %for.body ]
  %bits_left.1 = phi i32 [ %29, %if.end22 ], [ %bits_left.090, %for.body ]
  %sub = add nsw i32 %bits_left.1, -1
  %sh_prom = sext i32 %sub to i64
  %30 = shl i64 1, %sh_prom
  %and82 = and i64 %30, %get_buffer.1
  %tobool26 = icmp eq i64 %and82, 0
  br i1 %tobool26, label %for.inc, label %if.then27

```

```

if.then27:
  %arrayidx28 = getelementptr inbounds [64 x i16], [64 x i16]* %27, i64 0, i64
  ... 0
  %31 = load i16, i16* %arrayidx28, align 2, !tbaa !38
  %conv2983 = sext i16 %31 to i32
  %or = or i32 %conv2983, %sh1
  %conv30 = trunc i32 %or to i16
  store i16 %conv30, i16* %arrayidx28, align 2, !tbaa !38
  br label %for.inc

```

```

for.inc:
  %indvars.iv.next = add nuw nsw i64 %indvars.iv, 1
  %32 = load i32, i32* %blocks_in_MCU, align 8, !tbaa !34
  %33 = sext i32 %32 to i64
  %cmp16 = icmp slt i64 %indvars.iv.next, %33
  br i1 %cmp16, label %for.body, label %for.end.loopexit

```

```

for.end.loopexit:
  %sub.lessa = phi i32 [ %sub, %for.inc ]
  %get_buffer.1.lessa = phi i64 [ %get_buffer.1, %for.inc ]
  %pre93 = load i64*, i64** %19, align 8, !tbaa !29
  %34 = bitcast %struct.bitread_working_state* %br_state to <2 x i64>*
  %35 = load <2 x i64>, <2 x i64>* %34, align 16, !tbaa !30
  %pre95 = load i32, i32* %unread_marker11, align 16, !tbaa !31
  br label %for.end

```

```

for.end:
  %36 = phi i32 [ %8, %if.end6 ], [ %pre95, %for.end.loopexit ]
  %in = phi i64* [ %20, %if.end6 ], [ %pre93, %for.end.loopexit ]
  %bits_left.0.lessa = phi i32 [ %25, %if.end6 ], [ %sub.lessa,
  ... %for.end.loopexit ]
  %get_buffer.0.lessa = phi i64 [ %24, %if.end6 ], [ %get_buffer.1.lessa,
  ... %for.end.loopexit ]
  %37 = phi <2 x i64> [ %22, %if.end6 ], [ %35, %for.end.loopexit ]
  %38 = bitcast i64* %in to <2 x i64>*
  store <2 x i64> %37, <2 x i64>* %38, align 8, !tbaa !30
  store i32 %36, i32* %unread_marker, align 4, !tbaa !26
  store i64 %get_buffer.0.lessa, i64* %get_buffer12, align 8, !tbaa !32
  store i32 %bits_left.0.lessa, i32* %bits_left14, align 8, !tbaa !17
  %restarts_to_go44 = getelementptr inbounds %struct.phuff_entropy_decoder,
  ... %struct.phuff_entropy_decoder* %1, i64 0, i32 3
  %39 = load i32, i32* %restarts_to_go44, align 4, !tbaa !11
  %dec = add i32 %39, -1
  store i32 %dec, i32* %restarts_to_go44, align 4, !tbaa !11
  br label %cleanup

```

```

cleanup:
  %retval.0 = phi i32 [ 1, %for.end ], [ 0, %if.then2 ], [ 0,
  ... %cleanup.loopexit ]
  call void @llvm.lifetime.end(i64 56, i8* %5) #3
  ret i32 %retval.0

```

CFG for 'decode\_mcu\_DC\_refine' function