

```

for.body8.preheader:
    %cquantize1 = getelementptr inbounds %struct.jpeg_decompress_struct,
... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 83
    %0 = bitcast %struct.jpeg_color_quantizer** %cquantize1 to
... %struct.my_cquantizer**
    %1 = load %struct.my_cquantizer*, %struct.my_cquantizer** %0, align 8, !tbaa
... !2
    %mem = getelementptr inbounds %struct.jpeg_decompress_struct,
... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 1
    %2 = load %struct.jpeg_memory_mgr*, %struct.jpeg_memory_mgr** %mem, align 8,
... !tbaa !10
    %alloc_small = getelementptr inbounds %struct.jpeg_memory_mgr,
... %struct.jpeg_memory_mgr* %2, i64 0, i32 0
    %3 = load i8* (%struct.jpeg_common_struct*, i32, i64)*, i8*
... (%struct.jpeg_common_struct*, i32, i64)** %alloc_small, align 8, !tbaa !11
    %4 = bitcast %struct.jpeg_decompress_struct* %cinfo to
... %struct.jpeg_common_struct*
    %call = tail call i8* %3(%struct.jpeg_common_struct* %4, i32 1, i64 2044) #5
    %add.ptr = getelementptr inbounds i8, i8* %call, i64 1020
    %5 = bitcast i8* %add.ptr to i32*
    %error_limiter = getelementptr inbounds %struct.my_cquantizer,
... %struct.my_cquantizer* %1, i64 0, i32 7
    %6 = bitcast i32** %error_limiter to i8**
    store i8* %add.ptr, i8** %6, align 8, !tbaa !14
    %arrayidx.1 = getelementptr inbounds i8, i8* %call, i64 1024
    %arrayidx4.3 = getelementptr inbounds i8, i8* %call, i64 1008
    %7 = bitcast i8* %arrayidx4.3 to <4 x i32>*
    store <4 x i32> <i32 -3, i32 -2, i32 -1, i32 0>, <4 x i32>* %7, align 4,
... !tbaa !17
    %8 = bitcast i8* %arrayidx.1 to <4 x i32>*
    store <4 x i32> <i32 1, i32 2, i32 3, i32 4>, <4 x i32>* %8, align 4, !tbaa
... !17
    %arrayidx.5 = getelementptr inbounds i8, i8* %call, i64 1040
    %arrayidx4.7 = getelementptr inbounds i8, i8* %call, i64 992
    %9 = bitcast i8* %arrayidx4.7 to <4 x i32>*
    store <4 x i32> <i32 -7, i32 -6, i32 -5, i32 -4>, <4 x i32>* %9, align 4,
... !tbaa !17
    %10 = bitcast i8* %arrayidx.5 to <4 x i32>*
    store <4 x i32> <i32 5, i32 6, i32 7, i32 8>, <4 x i32>* %10, align 4, !tbaa
... !17
    %arrayidx.9 = getelementptr inbounds i8, i8* %call, i64 1056
    %arrayidx4.11 = getelementptr inbounds i8, i8* %call, i64 976
    %11 = bitcast i8* %arrayidx4.11 to <4 x i32>*
    store <4 x i32> <i32 -11, i32 -10, i32 -9, i32 -8>, <4 x i32>* %11, align 4,
... !tbaa !17
    %12 = bitcast i8* %arrayidx.9 to <4 x i32>*
    store <4 x i32> <i32 9, i32 10, i32 11, i32 12>, <4 x i32>* %12, align 4,
... !tbaa !17
    %arrayidx.13 = getelementptr inbounds i8, i8* %call, i64 1072
    %13 = bitcast i8* %arrayidx.13 to i32*
    store i32 13, i32* %13, align 4, !tbaa !17
    %arrayidx.14 = getelementptr inbounds i8, i8* %call, i64 1076
    %14 = bitcast i8* %arrayidx.14 to i32*
    store i32 14, i32* %14, align 4, !tbaa !17
    %arrayidx.15 = getelementptr inbounds i8, i8* %call, i64 1080
    %15 = bitcast i8* %arrayidx.15 to i32*
    store i32 15, i32* %15, align 4, !tbaa !17
    %arrayidx4.15 = getelementptr inbounds i8, i8* %call, i64 960
    %16 = bitcast i8* %arrayidx4.15 to <4 x i32>*
    store <4 x i32> <i32 -15, i32 -14, i32 -13, i32 -12>, <4 x i32>* %16, align
... 4, !tbaa !17
    br label %for.body8

```

```

for.body8:
    %indvars.iv71 = phi i64 [ 16, %for.body8.preheader ], [
... %indvars.iv.next72.1, %for.body8 ]
    %out.166 = phi i32 [ 16, %for.body8.preheader ], [ %add.1, %for.body8 ]
    %arrayidx10 = getelementptr inbounds i32, i32* %5, i64 %indvars.iv71
    store i32 %out.166, i32* %arrayidx10, align 4, !tbaa !17
    %sub11 = sub nsw i32 0, %out.166
    %17 = sub nsw i64 0, %indvars.iv71
    %arrayidx14 = getelementptr inbounds i32, i32* %5, i64 %17
    store i32 %sub11, i32* %arrayidx14, align 4, !tbaa !17
    %indvars.iv.next72 = or i64 %indvars.iv71, 1
    %arrayidx10.1 = getelementptr inbounds i32, i32* %5, i64 %indvars.iv.next72
    store i32 %out.166, i32* %arrayidx10.1, align 4, !tbaa !17
    %sub11.1 = sub nsw i32 0, %out.166
    %18 = xor i64 %indvars.iv71, -1
    %arrayidx14.1 = getelementptr inbounds i32, i32* %5, i64 %18
    store i32 %sub11.1, i32* %arrayidx14.1, align 4, !tbaa !17
    %indvars.iv.next72.1 = add nsw i64 %indvars.iv71, 2
    %19 = trunc i64 %indvars.iv.next72.1 to i32
    %add.1 = add nsw i32 %out.166, 1
    %exitcond75.1 = icmp eq i32 %19, 48
    br i1 %exitcond75.1, label %for.body20.lr.ph, label %for.body8

```

T

F

```

for.body20.lr.ph:
    %add.1.lcssa = phi i32 [ %add.1, %for.body8 ]
    %out.166.lcssa = phi i32 [ %out.166, %for.body8 ]
    %sub23 = xor i32 %out.166.lcssa, -1
    br label %for.body20

```

```

for.body20:
    %indvars.iv = phi i64 [ 48, %for.body20.lr.ph ], [ %indvars.iv.next.3,
... %for.body20 ]
    %arrayidx22 = getelementptr inbounds i32, i32* %5, i64 %indvars.iv
    store i32 %add.1.lcssa, i32* %arrayidx22, align 4, !tbaa !17
    %20 = sub nsw i64 0, %indvars.iv
    %arrayidx26 = getelementptr inbounds i32, i32* %5, i64 %20
    store i32 %sub23, i32* %arrayidx26, align 4, !tbaa !17
    %indvars.iv.next = or i64 %indvars.iv, 1
    %arrayidx22.1 = getelementptr inbounds i32, i32* %5, i64 %indvars.iv.next
    store i32 %add.1.lcssa, i32* %arrayidx22.1, align 4, !tbaa !17
    %21 = xor i64 %indvars.iv, -1
    %arrayidx26.1 = getelementptr inbounds i32, i32* %5, i64 %21
    store i32 %sub23, i32* %arrayidx26.1, align 4, !tbaa !17
    %indvars.iv.next.1 = or i64 %indvars.iv, 2
    %arrayidx22.2 = getelementptr inbounds i32, i32* %5, i64 %indvars.iv.next.1
    store i32 %add.1.lcssa, i32* %arrayidx22.2, align 4, !tbaa !17
    %22 = sub i64 -2, %indvars.iv
    %arrayidx26.2 = getelementptr inbounds i32, i32* %5, i64 %22
    store i32 %sub23, i32* %arrayidx26.2, align 4, !tbaa !17
    %indvars.iv.next.2 = or i64 %indvars.iv, 3
    %arrayidx22.3 = getelementptr inbounds i32, i32* %5, i64 %indvars.iv.next.2
    store i32 %add.1.lcssa, i32* %arrayidx22.3, align 4, !tbaa !17
    %23 = sub i64 -3, %indvars.iv
    %arrayidx26.3 = getelementptr inbounds i32, i32* %5, i64 %23
    store i32 %sub23, i32* %arrayidx26.3, align 4, !tbaa !17
    %indvars.iv.next.3 = add nsw i64 %indvars.iv, 4
    %lfr.wideiv.3 = trunc i64 %indvars.iv.next.3 to i32
    %exitcond.3 = icmp eq i32 %lfr.wideiv.3, 256
    br i1 %exitcond.3, label %for.end29, label %for.body20

```

T

F

```

for.end29:
    ret void

```

CFG for 'init_error_limit' function