

```
entry:
  %0 = bitcast %struct.djpeg_dest_struct* %info to %struct.bmp_dest_struct*
  %mem = getelementptr inbounds %struct.jpeg_decompress_struct,
  ... %struct.jpeg_decompress_struct.%info, i64 0, i32 1
  %1 = load %struct.jpeg_memory_mgr*, %struct.jpeg_memory_mgr**, %mem, align 8,
  ... !bbaa 12
  %access_virt_sarray = getelementptr inbounds %struct.jpeg_memory_mgr,
  ... %struct.jpeg_memory_mgr*, %1, i64 0, i32 7
  %2 = load i8**, !8** (%struct.jpeg_common_struct*, %struct.jvirt_sarray_control*,
  ... %struct.djpeg_dest_struct*, %struct.jpeg_common_struct*,
  ... %struct.jvirt_sarray_control*, i32, i32, i32)** %access_virt_sarray, align 8,
  ... !bbaa 110
  %3 = bitcast %struct.jpeg_decompress_struct* %cinfo to
  ... %struct.jpeg_common_struct*
  %whole_image = getelementptr inbounds %struct.djpeg_dest_struct,
  ... %struct.djpeg_dest_struct* %dinfo, i64 1, i32 1
  %4 = bitcast void (%struct.jpeg_decompress_struct*,
  ... %struct.djpeg_dest_struct*, i32)** %whole_image to
  ... %struct.jvirt_sarray_control**
  %5 = load %struct.jvirt_sarray_control*, %struct.jvirt_sarray_control**, %4,
  ... align 8, !bbaa 115
  %cur_output_row = getelementptr inbounds %struct.bmp_dest_struct,
  ... %struct.bmp_dest_struct* %0, i64 0, i32 6
  %6 = load i32, i32** %cur_output_row, align 4, !bbaa 116
  %call = tail call @8** (%2,%struct.jpeg_common_struct* %3,
  ... %struct.jvirt_sarray_control* %5, i32 %6, i32 1, i32 1) @6
  %7 = load i32, i32** %cur_output_row, align 4, !bbaa 116
  %inc = add i32 %7, 1
  store i32 %inc, i32** %cur_output_row, align 4, !bbaa 116
  %8 = load i8**, !8** %call, align 8, !bbaa 117
  %output_width = getelementptr inbounds %struct.jpeg_decompress_struct,
  ... %struct.jpeg_decompress_struct* %cinfo, i64 0, i32 26
  %9 = load i32, i32** %output_width, align 8, !bbaa 118
  %cmp24 = icmp eq i32 %9, 0
  br i1 %cmp24, label %for.end, label %for.body.preheader
```

```
for.body.preheader:
  %buffer = getelementptr inbounds %struct.djpeg_dest_struct,
  ... %struct.djpeg_dest_struct* %dinfo, i64 0, i32 4
  %10 = load i8**, !8** %buffer, align 8, !bbaa 119
  %11 = load i8**, !8** %10, align 8, !bbaa 117
  %12 = add i32 %9, -1
  %13 = zext i32 %12 to i64
  %14 = add nuw nsw i64 %13, 1
  %15 = zext i32 %12 to i64
  %16 = add nuw nsw i64 %15, 1
  %min.its.checked = icmp ult i64 %16, 32
  br i1 %min.its.checked, label %for.body.preheader162, label
  ... %min.its.checked
```

```
min.its.checked:
  %n.vee = and i64 %16, 8589934560
  %cmp.zero = icmp eq i64 %n.vee, 0
  br i1 %cmp.zero, label %for.body.preheader162, label %vector.memcheck
```

```
vector.memcheck:
  %17 = add i32 %9, -1
  %18 = zext i32 %17 to i64
  %scevgep28 = getelementptr i8, !8* %8, i64 %18
  %scevgep29 = getelementptr i8, !8* %8, !11, i64 %18
  %bound0 = icmp ult i8* %8, %scevgep29
  %bound1 = icmp ult i8* %11, %scevgep28
  %memcheck.conflict = and i1 %bound0, %bound1
  %cast.erd = true i64 %n.vee to i32
  %ind.end = sub i32 %9, %cast.erd
  %ind.end31 = getelementptr i8, !8* %8, i64 %n.vee
  %ind.end33 = getelementptr i8, !8* %11, i64 %n.vee
  br i1 %memcheck.conflict, label %for.body.preheader162, label
  ... %vector.body.preheader
```

```
vector.body.preheader:
  %19 = add i32 %9, -1
  %20 = zext i32 %19 to i64
  %21 = add nuw nsw i64 %20, 1
  %22 = and i64 %21, 8589934560
  %23 = add nsw i64 %22, -32
  %24 = lshr exact i64 %23, 5
  %25 = add nuw nsw i64 %24, 1
  %strater163 = and i64 %25, 3
  %icmp.mod164 = icmp eq i64 %strater163, 0
  br i1 %icmp.mod164, label %vector.body.preheader.split, label
  ... %vector.body.prol.preheader
```

```
vector.body.prol.preheader:
  br label %vector.body.prol
```

```
vector.body.prol:
  %index.prol = phi i64 [ %index.next.prol, %vector.body.prol ], [ 0,
  ... %vector.body.prol.preheader ]
  %prol.iter165 = phi i64 [ %prol.iter165.sub, %vector.body.prol ], [
  ... %strater163, %vector.body.prol.preheader ]
  %next.gep.prol = getelementptr i8, !8* %8, i64 %index.prol
  %next.gep97.prol = getelementptr i8, !8* %11, i64 %index.prol
  %26 = bitcast i8* %next.gep97.prol to <16 x i8*>
  %wide.load.prol = load <16 x i8>, <16 x i8*> %26, align 1, !bbaa 120
  %27 = getelementptr i8, !8* %next.gep97.prol, i64 16
  %28 = bitcast i8* %27 to <16 x i8*>
  %wide.load161.prol = load <16 x i8>, <16 x i8*> %28, align 1, !bbaa 120
  %29 = bitcast i8* %next.gep.prol to <16 x i8*>
  store <16 x i8> %wide.load.prol, <16 x i8*> %29, align 1, !bbaa 120
  %30 = getelementptr i8, !8* %next.gep.prol, i64 16
  %31 = bitcast i8* %30 to <16 x i8*>
  store <16 x i8> %wide.load161.prol, <16 x i8*> %31, align 1, !bbaa 120
  %index.next.prol = add i64 %index.prol, 32
  %prol.iter165.sub = add i64 %prol.iter165, -1
  %prol.iter165.cmp = icmp eq i64 %prol.iter165.sub, 0
  br i1 %prol.iter165.cmp, label %vector.body.preheader.split.loopexit, label
  ... %vector.body.prol, !llvm.loop 121
```

```
vector.body.preheader.split.loopexit:
  %index.next.prol.lessa = phi i64 [ 0, %index.next.prol, %vector.body.prol ],
  br label %vector.body.preheader.split
```

```
vector.body.preheader.split:
  %index.unr = phi i64 [ 0, %vector.body.preheader ], [
  ... %index.next.prol.lessa, %vector.body.preheader.split.loopexit ]
  %32 = icmp ult i64 %32, 96
  br i1 %32, label %middle.block, label %vector.body.preheader.split.split
```

```
vector.body.preheader.split.split:
  br label %vector.body
```

```
vector.body:
  %index = phi i64 [ %index.unr, %vector.body.preheader.split.split ], [
  ... %index.next.3, %vector.body ]
  %next.gep = getelementptr i8, !8* %8, i64 %index
  %next.gep97 = getelementptr i8, !8* %11, i64 %index
  %33 = bitcast i8* %next.gep97 to <16 x i8*>
  %wide.load = load <16 x i8>, <16 x i8*> %33, align 1, !bbaa 120
  %34 = getelementptr i8, !8* %next.gep97, i64 16
  %35 = bitcast i8* %34 to <16 x i8*>
  %wide.load161 = load <16 x i8>, <16 x i8*> %35, align 1, !bbaa 120
  %36 = bitcast i8* %next.gep to <16 x i8*>
  store <16 x i8> %wide.load, <16 x i8*> %36, align 1, !bbaa 120
  %37 = getelementptr i8, !8* %next.gep, i64 16
  %38 = bitcast i8* %37 to <16 x i8*>
  store <16 x i8> %wide.load161, <16 x i8*> %38, align 1, !bbaa 120
  %next.gep.1 = getelementptr i8, !8* %8, i64 %index.next
  %next.gep97.1 = getelementptr i8, !8* %11, i64 %index.next
  %39 = bitcast i8* %next.gep97.1 to <16 x i8*>
  %wide.load.1 = load <16 x i8>, <16 x i8*> %39, align 1, !bbaa 120
  %40 = getelementptr i8, !8* %next.gep97.1, i64 16
  %41 = bitcast i8* %40 to <16 x i8*>
  %wide.load161.1 = load <16 x i8>, <16 x i8*> %41, align 1, !bbaa 120
  %42 = bitcast i8* %next.gep.1 to <16 x i8*>
  store <16 x i8> %wide.load.1, <16 x i8*> %42, align 1, !bbaa 120
  %43 = getelementptr i8, !8* %next.gep.1, i64 16
  %44 = bitcast i8* %43 to <16 x i8*>
  store <16 x i8> %wide.load161.1, <16 x i8*> %44, align 1, !bbaa 120
  %index.next.1 = add i64 %index, 64
  %next.gep.2 = getelementptr i8, !8* %8, i64 %index.next.1
  %next.gep97.2 = getelementptr i8, !8* %11, i64 %index.next.1
  %45 = bitcast i8* %next.gep97.2 to <16 x i8*>
  %wide.load.2 = load <16 x i8>, <16 x i8*> %45, align 1, !bbaa 120
  %46 = getelementptr i8, !8* %next.gep97.2, i64 16
  %47 = bitcast i8* %46 to <16 x i8*>
  %wide.load161.2 = load <16 x i8>, <16 x i8*> %47, align 1, !bbaa 120
  %48 = bitcast i8* %next.gep.2 to <16 x i8*>
  store <16 x i8> %wide.load.2, <16 x i8*> %48, align 1, !bbaa 120
  %49 = getelementptr i8, !8* %next.gep.2, i64 16
  %50 = bitcast i8* %49 to <16 x i8*>
  store <16 x i8> %wide.load161.2, <16 x i8*> %50, align 1, !bbaa 120
  %index.next.2 = add i64 %index.next.1, 64
  %next.gep.3 = getelementptr i8, !8* %8, i64 %index.next.2
  %next.gep97.3 = getelementptr i8, !8* %11, i64 %index.next.2
  %51 = bitcast i8* %next.gep97.3 to <16 x i8*>
  %wide.load.3 = load <16 x i8>, <16 x i8*> %51, align 1, !bbaa 120
  %52 = getelementptr i8, !8* %next.gep97.3, i64 16
  %53 = bitcast i8* %52 to <16 x i8*>
  %wide.load161.3 = load <16 x i8>, <16 x i8*> %53, align 1, !bbaa 120
  %54 = bitcast i8* %next.gep.3 to <16 x i8*>
  store <16 x i8> %wide.load.3, <16 x i8*> %54, align 1, !bbaa 120
  %55 = getelementptr i8, !8* %next.gep.3, i64 16
  %56 = bitcast i8* %55 to <16 x i8*>
  store <16 x i8> %wide.load161.3, <16 x i8*> %56, align 1, !bbaa 120
  %index.next.3 = add i64 %index.next.2, 64
  %57 = icmp eq i64 %index.next.3, %n.vee
  br i1 %57, label %middle.block.unr.lessa, label %vector.body, !llvm.loop 123
```

```
middle.block.unr.lessa:
  br label %middle.block
```

```
middle.block:
  %cmp.n = icmp eq i64 %16, %n.vee
  br i1 %cmp.n, label %for.end.loopexit, label %for.body.preheader162
```

```
for.body.preheader162:
  %col.027.ph = phi i32 [ %9, %vector.memcheck ], [ %9, %min.its.checked ],
  [ %9, %for.body.preheader ], [ %ind.end, %middle.block ]
  %outpr.026.ph = phi i8* [ %8, %vector.memcheck ], [ %8, %min.its.checked
  ... ], [ %8, %for.body.preheader ], [ %ind.end31, %middle.block ]
  %inpr.025.ph = phi i8* [ %11, %vector.memcheck ], [ %11, %min.its.checked
  ... ], [ %11, %for.body.preheader ], [ %ind.end33, %middle.block ]
  %58 = add i32 %col.027.ph, -8
  %strater = and i32 %col.027.ph, 7
  %icmp.mod = icmp eq i32 %strater, 0
  br i1 %icmp.mod, label %for.body.preheader162.split, label
  ... %for.body.prol.preheader
```

```
for.body.prol.preheader:
  br label %for.body.prol
```

```
for.body.prol:
  %col.027.prol = phi i32 [ %dec.prol, %for.body.prol ], [ %col.027.ph,
  ... %for.body.prol.preheader ]
  %outpr.026.prol = phi i8* [ %indec.ptr3.prol, %for.body.prol ], [
  %outpr.026.ph, %for.body.prol.preheader ]
  %inpr.025.prol = phi i8* [ %indec.ptr.prol, %for.body.prol ], [
  ... %inpr.025.ph, %for.body.prol.preheader ]
  %prol.iter = phi i32 [ %prol.iter.sub, %for.body.prol ], [ %strater,
  ... %for.body.prol.preheader ]
  %indec.ptr.prol = getelementptr inbounds i8, !8* %inpr.025.prol, i64 1
  %59 = load i8, !8* %inpr.025.prol, align 1, !bbaa 120
  %indec.ptr3.prol = getelementptr inbounds i8, !8* %outpr.026.prol, i64 1
  store i8 %59, !8* %outpr.026.prol, align 1, !bbaa 120
  %dec.prol = add i32 %col.027.prol, -1
  %prol.iter.sub = add i32 %prol.iter, -1
  %prol.iter.cmp = icmp eq i32 %prol.iter.sub, 0
  br i1 %prol.iter.cmp, label %for.body.preheader162.split.loopexit, label
  ... %for.body.prol, !llvm.loop 126
```

```
for.body.preheader162.split.loopexit:
  %dec.prol.lessa = phi i32 [ %dec.prol, %for.body.prol ],
  %indec.ptr3.prol.lessa = phi i8* [ %indec.ptr3.prol, %for.body.prol ],
  %indec.ptr.prol.lessa = phi i8* [ %indec.ptr.prol, %for.body.prol ],
  br label %for.body.preheader162.split
```

```
for.body.preheader162.split:
  %col.027.unr = phi i32 [ %col.027.ph, %for.body.preheader162 ], [
  ... %dec.prol.lessa, %for.body.preheader162.split.loopexit ]
  %outpr.026.unr = phi i8* [ %outpr.026.ph, %for.body.preheader162 ], [
  ... %indec.ptr3.prol.lessa, %for.body.preheader162.split.loopexit ]
  %inpr.025.unr = phi i8* [ %inpr.025.ph, %for.body.preheader162 ], [
  ... %indec.ptr.prol.lessa, %for.body.preheader162.split.loopexit ]
  %60 = icmp ult i32 %58, 7
  br i1 %60, label %for.end.loopexit.loopexit, label
  ... %for.body.preheader162.split.split
```

```
for.body.preheader162.split.split:
  br label %for.body
```

```
for.body:
  %col.027 = phi i32 [ %col.027.unr, %for.body.preheader162.split.split ], [
  ... %dec.7, %for.body ]
  %outpr.026 = phi i8* [ %outpr.026.unr, %for.body.preheader162.split.split
  ... ], [ %indec.ptr3.7, %for.body ]
  %inpr.025 = phi i8* [ %inpr.025.unr, %for.body.preheader162.split.split ],
  ... [ %indec.ptr, %for.body ]
  %indec.ptr = getelementptr inbounds i8, !8* %inpr.025, i64 1
  %61 = load i8, !8* %inpr.025, align 1, !bbaa 120
  %indec.ptr3 = getelementptr inbounds i8, !8* %outpr.026, i64 1
  store i8 %61, !8* %outpr.026, align 1, !bbaa 120
  %indec.ptr.1 = getelementptr inbounds i8, !8* %inpr.025, i64 2
  %62 = load i8, !8* %indec.ptr, align 1, !bbaa 120
  %indec.ptr3.1 = getelementptr inbounds i8, !8* %outpr.026, i64 2
  store i8 %62, !8* %indec.ptr.1, !bbaa 120
  %indec.ptr.2 = getelementptr inbounds i8, !8* %inpr.025, i64 3
  %63 = load i8, !8* %indec.ptr.1, !bbaa 120
  %indec.ptr3.2 = getelementptr inbounds i8, !8* %outpr.026, i64 3
  store i8 %63, !8* %indec.ptr.2, !bbaa 120
  %indec.ptr.3 = getelementptr inbounds i8, !8* %inpr.025, i64 4
  %64 = load i8, !8* %indec.ptr.2, align 1, !bbaa 120
  %indec.ptr3.3 = getelementptr inbounds i8, !8* %outpr.026, i64 4
  store i8 %64, !8* %indec.ptr.3, align 1, !bbaa 120
  %indec.ptr.4 = getelementptr inbounds i8, !8* %inpr.025, i64 5
  %65 = load i8, !8* %indec.ptr.3, align 1, !bbaa 120
  %indec.ptr3.4 = getelementptr inbounds i8, !8* %outpr.026, i64 5
  store i8 %65, !8* %indec.ptr.4, !bbaa 120
  %indec.ptr.5 = getelementptr inbounds i8, !8* %inpr.025, i64 6
  %66 = load i8, !8* %indec.ptr.4, align 1, !bbaa 120
  %indec.ptr3.5 = getelementptr inbounds i8, !8* %outpr.026, i64 6
  store i8 %66, !8* %indec.ptr.5, align 1, !bbaa 120
  %indec.ptr.6 = getelementptr inbounds i8, !8* %inpr.025, i64 7
  %67 = load i8, !8* %indec.ptr.5, align 1, !bbaa 120
  %indec.ptr3.6 = getelementptr inbounds i8, !8* %outpr.026, i64 7
  store i8 %67, !8* %indec.ptr.6, align 1, !bbaa 120
  %indec.ptr.7 = getelementptr inbounds i8, !8* %inpr.025, i64 8
  %68 = load i8, !8* %indec.ptr.6, align 1, !bbaa 120
  %indec.ptr3.7 = getelementptr inbounds i8, !8* %outpr.026, i64 8
  store i8 %68, !8* %indec.ptr.7, align 1, !bbaa 120
  %dec.7 = add i32 %col.027, -8
  %cmp.7 = icmp eq i32 %dec.7, 0
  br i1 %cmp.7, label %for.end.loopexit.loopexit.unr.lessa, label %for.body,
  ... !llvm.loop 127
```

```
for.end.loopexit.loopexit.unr.lessa:
  br label %for.end.loopexit.loopexit
```

```
for.end.loopexit:
  %scevgep = getelementptr i8, !8* %8, i64 %14
  br label %for.end
```

```
for.end:
  %outpr.0.lessa = phi i8* [ %8, %entry ], [ %scevgep, %for.end.loopexit ]
  %pad_bytes = getelementptr inbounds %struct.djpeg_dest_struct,
  ... %struct.djpeg_dest_struct* %dinfo, i64 1, i32 3
  %69 = bitcast %struct_IO_FILE** %pad_bytes to i32*
  %70 = load i32, i32** %69, align 8, !bbaa 128
  %cmp521 = icmp sgt i32 %70, 0
  br i1 %cmp521, label %while.body.preheader, label %while.end
```

```
while.body.preheader:
  %71 = add i32 %70, -1
  %72 = zext i32 %71 to i64
  %73 = add nuw nsw i64 %72, 1
  call void @llvm.memset.pri8.i64@8* %outpr.0.lessa, i8 0, i64 %73, i32 1,
  ... !1 false
  br label %while.end
```

```
while.end:
  ret void
```