

```
%i1:
%mul.i.i = shl i64 %8, 6
%mul3.i.i = shl i64 %9, 5
%mul6.i = mul i32 %6, %1
%12 = trunc i64 %9 to i32
%13 = shl i32 %12, 5
%14 = add i32 %mul6.i, %13
%15 = mul i32 %14, %2
%16 = trunc i64 %8 to i32
%17 = shl i32 %16, 6
%18 = add i32 %15, %17
%broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
%broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
  undef, <8 x i32> zeroinitializer
%broadcast.splatinsert4 = insertelement <8 x i32> undef, i32 %2, i32 0
%broadcast.splat5 = shufflevector <8 x i32> %broadcast.splatinsert4, <8 x
  i32> undef, <8 x i32> zeroinitializer
%19 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%20 = or <8 x i32> %19, <i32 0, i32 2, i32 3, i32 4, i32 5, i32 6,
  ... i32 7>
%21 = icmp sgt <8 x i32> %broadcast.splat5, %20
%22 = extractelement <8 x i32> %20, i32 0
%23 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%24 = or <8 x i32> %23, <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32
  ... 14, i32 15>
%25 = icmp sgt <8 x i32> %broadcast.splat5, %24
%26 = extractelement <8 x i32> %20, i32 0
%27 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%28 = or <8 x i32> %27, <i32 16, i32 17, i32 18, i32 19, i32 20, i32 21, i32
  ... 22, i32 23>
%29 = icmp sgt <8 x i32> %broadcast.splat5, %28
%30 = extractelement <8 x i32> %28, i32 0
%31 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%32 = or <8 x i32> %31, <i32 24, i32 25, i32 26, i32 27, i32 28, i32 29, i32
  ... 30, i32 31>
%33 = icmp sgt <8 x i32> %broadcast.splat5, %32
%34 = extractelement <8 x i32> %32, i32 0
%35 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%36 = or <8 x i32> %35, <i32 32, i32 33, i32 34, i32 35, i32 36, i32 37, i32
  ... 38, i32 39>
%37 = icmp sgt <8 x i32> %broadcast.splat5, %36
%38 = extractelement <8 x i32> %36, i32 0
%39 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%40 = or <8 x i32> %39, <i32 40, i32 41, i32 42, i32 43, i32 44, i32 45, i32
  ... 46, i32 47>
%41 = icmp sgt <8 x i32> %broadcast.splat5, %40
%42 = extractelement <8 x i32> %40, i32 0
%43 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%44 = or <8 x i32> %43, <i32 48, i32 49, i32 50, i32 51, i32 52, i32 53, i32
  ... 54, i32 55>
%45 = icmp sgt <8 x i32> %broadcast.splat5, %44
%46 = extractelement <8 x i32> %44, i32 0
%47 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%48 = or <8 x i32> %47, <i32 56, i32 57, i32 58, i32 59, i32 60, i32 61, i32
  ... 62, i32 63>
%49 = icmp sgt <8 x i32> %broadcast.splat5, %48
%50 = extractelement <8 x i32> %48, i32 0
br label %pregion_for_entry.pregion_for_init.i
```

```
pregion_for_entry.pregion_for_init.i:
% local_id.y.0 = phi i64 [ 0, %i1 ], [ %i07, %pregion_for_end.i ]
%add6.i.i = add nuw nsw i64 % local_id.y.0, %mul3.i.i, !llvm.access.group !12
%conv2.i = trunc i64 %add6.i.i to i32, !llvm.access.group !12
%cmp4.i = icmp slt i32 %conv2.i, %1, !llvm.access.group !12
%reass.add.i = add i32 %mul6.i, %conv2.i
%reass.mul.i = mul i32 %reass.add.i, %2
br i1 %cmp4.i, label %vector.scevcheck, label %pregion_for_end.i
```

```
vector.scevcheck:
%51 = trunc i64 % local_id.y.0 to i32
%52 = mul i32 %51, %2
%53 = add i32 %52, %18
%54 = icmp sgt i32 %53, 2147483584
br i1 %54, label %pregion_for_entry.entry.i.us.preheader, label %vector.body
```

```
pregion_for_entry.entry.i.us.preheader:
br label %pregion_for_entry.entry.i.us
```

```
vector.body:
%55 = add i32 %reass.mul.i, %22, !llvm.access.group !12
%56 = sext i32 %55 to i64, !llvm.access.group !12
%57 = getelementptr inbounds float, float* %5, i64 %56, !llvm.access.group
  ... !12
%58 = bitcast float* %57 to <8 x i32>*
%wide.load = load <8 x i32>, <8 x i32>* %58, align 4, !tbaa !15,
  ... !llvm.access.group !12
%59 = getelementptr inbounds float, float* %3, i64 %56, !llvm.access.group
  ... !12
%60 = bitcast float* %59 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.load, <8 x i32>*
  ... %60, i32 4, <8 x i1> %21), !tbaa !15, !llvm.access.group !12
%61 = add i32 %reass.mul.i, %26, !llvm.access.group !12
%62 = sext i32 %61 to i64, !llvm.access.group !12
%63 = getelementptr inbounds float, float* %5, i64 %62, !llvm.access.group
  ... !12
%64 = bitcast float* %63 to <8 x i32>*
%wide.load.1 = load <8 x i32>, <8 x i32>* %64, align 4, !tbaa !15,
  ... !llvm.access.group !12
%65 = getelementptr inbounds float, float* %3, i64 %62, !llvm.access.group
  ... !12
%66 = bitcast float* %65 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.load.1, <8 x
  ... i32>* %66, i32 4, <8 x i1> %25), !tbaa !15, !llvm.access.group !12
%67 = add i32 %reass.mul.i, %30, !llvm.access.group !12
%68 = sext i32 %67 to i64, !llvm.access.group !12
%69 = getelementptr inbounds float, float* %5, i64 %68, !llvm.access.group
  ... !12
%70 = bitcast float* %69 to <8 x i32>*
%wide.load.2 = load <8 x i32>, <8 x i32>* %70, align 4, !tbaa !15,
  ... !llvm.access.group !12
%71 = getelementptr inbounds float, float* %3, i64 %68, !llvm.access.group
  ... !12
%72 = bitcast float* %71 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.load.2, <8 x
  ... i32>* %72, i32 4, <8 x i1> %29), !tbaa !15, !llvm.access.group !12
%73 = add i32 %reass.mul.i, %34, !llvm.access.group !12
%74 = sext i32 %73 to i64, !llvm.access.group !12
%75 = getelementptr inbounds float, float* %5, i64 %74, !llvm.access.group
  ... !12
%76 = bitcast float* %75 to <8 x i32>*
%wide.load.3 = load <8 x i32>, <8 x i32>* %76, align 4, !tbaa !15,
  ... !llvm.access.group !12
%77 = getelementptr inbounds float, float* %3, i64 %74, !llvm.access.group
  ... !12
%78 = bitcast float* %77 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.load.3, <8 x
  ... i32>* %78, i32 4, <8 x i1> %33), !tbaa !15, !llvm.access.group !12
%79 = add i32 %reass.mul.i, %38, !llvm.access.group !12
%80 = sext i32 %79 to i64, !llvm.access.group !12
%81 = getelementptr inbounds float, float* %5, i64 %80, !llvm.access.group
  ... !12
%82 = bitcast float* %81 to <8 x i32>*
%wide.load.4 = load <8 x i32>, <8 x i32>* %82, align 4, !tbaa !15,
  ... !llvm.access.group !12
%83 = getelementptr inbounds float, float* %3, i64 %80, !llvm.access.group
  ... !12
%84 = bitcast float* %83 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.load.4, <8 x
  ... i32>* %84, i32 4, <8 x i1> %37), !tbaa !15, !llvm.access.group !12
%85 = add i32 %reass.mul.i, %42, !llvm.access.group !12
%86 = sext i32 %85 to i64, !llvm.access.group !12
%87 = getelementptr inbounds float, float* %5, i64 %86, !llvm.access.group
  ... !12
%88 = bitcast float* %87 to <8 x i32>*
%wide.load.5 = load <8 x i32>, <8 x i32>* %88, align 4, !tbaa !15,
  ... !llvm.access.group !12
%89 = getelementptr inbounds float, float* %3, i64 %86, !llvm.access.group
  ... !12
%90 = bitcast float* %89 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.load.5, <8 x
  ... i32>* %90, i32 4, <8 x i1> %41), !tbaa !15, !llvm.access.group !12
%91 = add i32 %reass.mul.i, %46, !llvm.access.group !12
%92 = sext i32 %91 to i64, !llvm.access.group !12
%93 = getelementptr inbounds float, float* %5, i64 %92, !llvm.access.group
  ... !12
%94 = bitcast float* %93 to <8 x i32>*
%wide.load.6 = load <8 x i32>, <8 x i32>* %94, align 4, !tbaa !15,
  ... !llvm.access.group !12
%95 = getelementptr inbounds float, float* %3, i64 %92, !llvm.access.group
  ... !12
%96 = bitcast float* %95 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.load.6, <8 x
  ... i32>* %96, i32 4, <8 x i1> %45), !tbaa !15, !llvm.access.group !12
%97 = add i32 %reass.mul.i, %50, !llvm.access.group !12
%98 = sext i32 %97 to i64, !llvm.access.group !12
%99 = getelementptr inbounds float, float* %5, i64 %98, !llvm.access.group
  ... !12
%100 = bitcast float* %99 to <8 x i32>*
%wide.load.7 = load <8 x i32>, <8 x i32>* %100, align 4, !tbaa !15,
  ... !llvm.access.group !12
%101 = getelementptr inbounds float, float* %3, i64 %98, !llvm.access.group
  ... !12
%102 = bitcast float* %101 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.load.7, <8 x
  ... i32>* %102, i32 4, <8 x i1> %49), !tbaa !15, !llvm.access.group !12
br label %pregion_for_end.i
```

```
pregion_for_entry.entry.i.us:
% local_id.x.0.us = phi i64 [ %i19, %if.end.r_exit.i.us.3 ], [ 0,
  ... %pregion_for_entry.entry.i.us.preheader ]
%add1.i.i.us = add nuw nsw i64 % local_id.x.0.us, %mul.i.i,
  ... !llvm.access.group !12
%conv1.us = trunc i64 %add1.i.i.us to i32, !llvm.access.group !12
%cmp.i.us = icmp slt i32 %conv1.us, %2, !llvm.access.group !12
br i1 %cmp.i.us, label %if.then.i.us, label %if.end.r_exit.i.us,
  ... !llvm.access.group !12
```

```
if.then.i.us:
%add8.i.us = add i32 %reass.mul.i, %conv1.us, !llvm.access.group !12
%idxprom.i.us = sext i32 %add8.i.us to i64, !llvm.access.group !12
%arrayidx.i.us = getelementptr inbounds float, float* %5, i64 %idxprom.i.us,
  ... !llvm.access.group !12
%103 = bitcast float* %arrayidx.i.us to i32*, !llvm.access.group !12
%104 = load i32, i32* %103, align 4, !tbaa !15, !llvm.access.group !12
%arrayidx15.i.us = getelementptr inbounds float, float* %3, i64
  ... %idxprom.i.us, !llvm.access.group !12
%105 = bitcast float* %arrayidx15.i.us to i32*, !llvm.access.group !12
store i32 %104, i32* %105, align 4, !tbaa !15, !llvm.access.group !12
br label %if.end.r_exit.i.us, !llvm.access.group !12
```

```
if.end.r_exit.i.us:
%106 = or i64 % local_id.x.0.us, 1
%add1.i.i.us.1 = add nuw nsw i64 %106, %mul.i.i, !llvm.access.group !12
%conv1.us.1 = trunc i64 %add1.i.i.us.1 to i32, !llvm.access.group !12
%cmp.i.us.1 = icmp slt i32 %conv1.us.1, %2, !llvm.access.group !12
br i1 %cmp.i.us.1, label %if.then.i.us.1, label %if.end.r_exit.i.us.1,
  ... !llvm.access.group !12
```

```
if.then.i.us.1:
%add8.i.us.1 = add i32 %reass.mul.i, %conv1.us.1, !llvm.access.group !12
%idxprom.i.us.1 = sext i32 %add8.i.us.1 to i64, !llvm.access.group !12
%arrayidx.i.us.1 = getelementptr inbounds float, float* %5, i64
  ... %idxprom.i.us.1, !llvm.access.group !12
%108 = bitcast float* %arrayidx.i.us.1 to i32*, !llvm.access.group !12
%109 = load i32, i32* %108, align 4, !tbaa !15, !llvm.access.group !12
%arrayidx15.i.us.1 = getelementptr inbounds float, float* %3, i64
  ... %idxprom.i.us.1, !llvm.access.group !12
%110 = bitcast float* %arrayidx15.i.us.1 to i32*, !llvm.access.group !12
store i32 %109, i32* %110, align 4, !tbaa !15, !llvm.access.group !12
br label %if.end.r_exit.i.us.1, !llvm.access.group !12
```

```
if.end.r_exit.i.us.1:
%111 = or i64 % local_id.x.0.us, 2
%add1.i.i.us.2 = add nuw nsw i64 %111, %mul.i.i, !llvm.access.group !12
%conv1.us.2 = trunc i32 %conv1.us.2, %2, !llvm.access.group !12
br i1 %cmp.i.us.2, label %if.then.i.us.2, label %if.end.r_exit.i.us.2,
  ... !llvm.access.group !12
```

```
if.then.i.us.2:
%add8.i.us.2 = add i32 %reass.mul.i, %conv1.us.2, !llvm.access.group !12
%idxprom.i.us.2 = sext i32 %add8.i.us.2 to i64, !llvm.access.group !12
%arrayidx.i.us.2 = getelementptr inbounds float, float* %5, i64
  ... %idxprom.i.us.2, !llvm.access.group !12
%112 = bitcast float* %arrayidx.i.us.2 to i32*, !llvm.access.group !12
%113 = load i32, i32* %112, align 4, !tbaa !15, !llvm.access.group !12
%arrayidx15.i.us.2 = getelementptr inbounds float, float* %3, i64
  ... %idxprom.i.us.2, !llvm.access.group !12
%114 = bitcast float* %arrayidx15.i.us.2 to i32*, !llvm.access.group !12
store i32 %113, i32* %114, align 4, !tbaa !15, !llvm.access.group !12
br label %if.end.r_exit.i.us.2, !llvm.access.group !12
```

```
if.end.r_exit.i.us.2:
%115 = or i64 % local_id.x.0.us, 3
%add1.i.i.us.3 = add nuw nsw i64 %115, %mul.i.i, !llvm.access.group !12
%conv1.us.3 = trunc i64 %add1.i.i.us.3 to i32, !llvm.access.group !12
%cmp.i.us.3 = icmp slt i32 %conv1.us.3, %2, !llvm.access.group !12
br i1 %cmp.i.us.3, label %if.then.i.us.3, label %if.end.r_exit.i.us.3,
  ... !llvm.access.group !12
```

```
if.then.i.us.3:
%add8.i.us.3 = add i32 %reass.mul.i, %conv1.us.3, !llvm.access.group !12
%idxprom.i.us.3 = sext i32 %add8.i.us.3 to i64, !llvm.access.group !12
%arrayidx.i.us.3 = getelementptr inbounds float, float* %5, i64
  ... %idxprom.i.us.3, !llvm.access.group !12
%116 = bitcast float* %arrayidx.i.us.3 to i32*, !llvm.access.group !12
%117 = load i32, i32* %116, align 4, !tbaa !15, !llvm.access.group !12
%arrayidx15.i.us.3 = getelementptr inbounds float, float* %3, i64
  ... %idxprom.i.us.3, !llvm.access.group !12
%118 = bitcast float* %arrayidx15.i.us.3 to i32*, !llvm.access.group !12
store i32 %117, i32* %118, align 4, !tbaa !15, !llvm.access.group !12
br label %if.end.r_exit.i.us.3, !llvm.access.group !12
```

```
if.end.r_exit.i.us.3:
%119 = add nuw nsw i64 % local_id.x.0.us, 4
%exitcond.not.3 = icmp eq i64 %119, 64
br i1 %exitcond.not.3, label %pregion_for_end.i.loopexit, label
  ... %pregion_for_entry.entry.i.us, !llvm.loop !21
```

```
pregion_for_end.i.loopexit:
br label %pregion_for_end.i
```

```
pregion_for_end.i:
%107 = add nuw nsw i64 % local_id.y.0, 1
%exitcond3.not = icmp eq i64 %107, 32
br i1 %exitcond3.not, label %doitgen_kernel2.exit, label
  ... %pregion_for_entry.pregion_for_init.i, !llvm.loop !19
```

```
doitgen_kernel2.exit:
ret void
```