

```
%i11:
%i12 = shl i64 %8, 6
%i13 = shl i64 %9, 5
%mul6.i = mul i32 %6, %1
%i14 = trunc i64 %9 to i32
%i15 = shl i32 %14, 5
%i16 = add i32 %mul6.i, %i15
%i17 = mul i32 %i16, %2
%i18 = trunc i64 %8 to i32
%i19 = shl i32 %i18, 6
%i20 = add i32 %i17, %i19
%scevgep5 = getelementptr float, float* %3, i64 64
%scevgep10 = getelementptr float, float* %5, i64 64
%bound0 = icmp ugt float* %scevgep10, %3
%bound1 = icmp ugt float* %scevgep5, %5
%found.conflict = and i1 %bound0, %bound1
%broadcast.splat = shufflevector <8 x i64> undef, i64 %i2, i32 0
%broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
... undef, <8 x i32> zeroinitializer
%broadcast.splatinsert13 = insertelement <8 x i32> undef, i32 %2, i32 0
%broadcast.splat14 = shufflevector <8 x i32> %broadcast.splatinsert13, <8 x
... i32> undef, <8 x i32> zeroinitializer
%21 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%22 = add <8 x i32> %21, <i32 0, i32 1, i32 2, i32 3, i32 4, i32 5, i32 6,
... i32 7>
%23 = icmp sgt <8 x i32> %broadcast.splat14, %22
%24 = extractelement <8 x i32> %22, i32 0
%25 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%26 = add <8 x i32> %25, <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32
... 14, i32 15>
%27 = icmp sgt <8 x i32> %broadcast.splat14, %26
%28 = extractelement <8 x i32> %26, i32 0
%29 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%30 = add <8 x i32> %29, <i32 16, i32 17, i32 18, i32 19, i32 20, i32 21,
... i32 22, i32 23>
%31 = icmp sgt <8 x i32> %broadcast.splat14, %30
%32 = extractelement <8 x i32> %30, i32 0
%33 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%34 = add <8 x i32> %33, <i32 24, i32 25, i32 26, i32 27, i32 28, i32 29,
... i32 30, i32 31>
%35 = icmp sgt <8 x i32> %broadcast.splat14, %34
%36 = extractelement <8 x i32> %34, i32 0
%37 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%38 = add <8 x i32> %37, <i32 32, i32 33, i32 34, i32 35, i32 36, i32 37,
... i32 38, i32 39>
%39 = icmp sgt <8 x i32> %broadcast.splat14, %38
%40 = extractelement <8 x i32> %38, i32 0
%41 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%42 = add <8 x i32> %41, <i32 40, i32 41, i32 42, i32 43, i32 44, i32 45,
... i32 46, i32 47>
%43 = icmp sgt <8 x i32> %broadcast.splat14, %42
%44 = extractelement <8 x i32> %42, i32 0
%45 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%46 = add <8 x i32> %45, <i32 48, i32 49, i32 50, i32 51, i32 52, i32 53,
... i32 54, i32 55>
%47 = icmp sgt <8 x i32> %broadcast.splat14, %46
%48 = extractelement <8 x i32> %46, i32 0
%49 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%50 = add <8 x i32> %49, <i32 56, i32 57, i32 58, i32 59, i32 60, i32 61,
... i32 62, i32 63>
%51 = icmp sgt <8 x i32> %broadcast.splat14, %50
%52 = extractelement <8 x i32> %50, i32 0
br label %region_for_entry.pregion_for_init.i
```

```
pregion_for_entry.pregion_for_init.i:
%local_id.y0 = phi i64 [ 0, %i11 ], [ %i112, %region_for_end.i ]
%53 = add nuw nsw i64 %local_id.y0, %i13
%conv2.i = trunc i64 %53 to i32
%cmp4.i = icmp slt i32 %conv2.i, %1
%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 %region_for_end.i
```

```
vector.scevcheck:
%54 = trunc i64 %local_id.y0 to i32
%55 = mul i32 %54, %2
%56 = add i32 %20, %55
%57 = icmp sgt i32 %56, 2147483584
%brmerge = or i1 %57, %found.conflict
br i1 %brmerge, label %region_for_entry.entry.i.us.preheader, label
... %vectorbody
```

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

```
vector.body:
%58 = add i32 %reass.mul.i, %24
%59 = sext i32 %58 to i64
%60 = getelementptr inbounds float, float* %5, i64 %59
%61 = bitcast float* %60 to <8 x i32>
%wide.masked.load = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %61, i32 4, <8 x i1> %23, <8 x i32> undef), !tbaa !12, !alias.scope !16
%62 = getelementptr inbounds float, float* %3, i64 %59
%63 = bitcast float* %62 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load, <8 x
... i32>* %63, i32 4, <8 x i1> %23), !tbaa !12, !alias.scope !19, !noalias !16,
... !llvm.access.group !21
%64 = add i32 %reass.mul.i, %28
%65 = sext i32 %64 to i64
%66 = getelementptr inbounds float, float* %5, i64 %65
%67 = bitcast float* %66 to <8 x i32>*
%wide.masked.load.1 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %67, i32 4, <8 x i1> %27, <8 x i32> undef), !tbaa !12, !alias.scope !16
%68 = getelementptr inbounds float, float* %3, i64 %65
%69 = bitcast float* %68 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.1, <8
... x i32>* %69, i32 4, <8 x i1> %27), !tbaa !12, !alias.scope !19, !noalias !16,
... !llvm.access.group !21
%70 = add i32 %reass.mul.i, %32
%71 = sext i32 %70 to i64
%72 = getelementptr inbounds float, float* %5, i64 %71
%73 = bitcast float* %72 to <8 x i32>*
%wide.masked.load.2 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %73, i32 4, <8 x i1> %31, <8 x i32> undef), !tbaa !12, !alias.scope !16
%74 = getelementptr inbounds float, float* %3, i64 %71
%75 = bitcast float* %74 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.2, <8
... x i32>* %75, i32 4, <8 x i1> %31), !tbaa !12, !alias.scope !19, !noalias !16,
... !llvm.access.group !21
%76 = add i32 %reass.mul.i, %36
%77 = sext i32 %76 to i64
%78 = getelementptr inbounds float, float* %5, i64 %77
%79 = bitcast float* %78 to <8 x i32>*
%wide.masked.load.3 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %79, i32 4, <8 x i1> %35, <8 x i32> undef), !tbaa !12, !alias.scope !16
%80 = getelementptr inbounds float, float* %3, i64 %77
%81 = bitcast float* %80 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.3, <8
... x i32>* %81, i32 4, <8 x i1> %35), !tbaa !12, !alias.scope !19, !noalias !16,
... !llvm.access.group !21
%82 = add i32 %reass.mul.i, %40
%83 = sext i32 %82 to i64
%84 = getelementptr inbounds float, float* %5, i64 %83
%85 = bitcast float* %84 to <8 x i32>*
%wide.masked.load.4 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %85, i32 4, <8 x i1> %39, <8 x i32> undef), !tbaa !12, !alias.scope !16
%86 = getelementptr inbounds float, float* %3, i64 %83
%87 = bitcast float* %86 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.4, <8
... x i32>* %87, i32 4, <8 x i1> %39), !tbaa !12, !alias.scope !19, !noalias !16,
... !llvm.access.group !21
%88 = add i32 %reass.mul.i, %44
%89 = sext i32 %88 to i64
%90 = getelementptr inbounds float, float* %5, i64 %89
%91 = bitcast float* %90 to <8 x i32>*
%wide.masked.load.5 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %91, i32 4, <8 x i1> %43, <8 x i32> undef), !tbaa !12, !alias.scope !16
%92 = getelementptr inbounds float, float* %3, i64 %89
%93 = bitcast float* %92 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.5, <8
... x i32>* %93, i32 4, <8 x i1> %43), !tbaa !12, !alias.scope !19, !noalias !16,
... !llvm.access.group !21
%94 = add i32 %reass.mul.i, %48
%95 = sext i32 %94 to i64
%96 = getelementptr inbounds float, float* %5, i64 %95
%97 = bitcast float* %96 to <8 x i32>*
%wide.masked.load.6 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %97, i32 4, <8 x i1> %47, <8 x i32> undef), !tbaa !12, !alias.scope !16
%98 = getelementptr inbounds float, float* %3, i64 %95
%99 = bitcast float* %98 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.6, <8
... x i32>* %99, i32 4, <8 x i1> %47), !tbaa !12, !alias.scope !19, !noalias !16,
... !llvm.access.group !21
%100 = add i32 %reass.mul.i, %52
%101 = sext i32 %100 to i64
%102 = getelementptr inbounds float, float* %5, i64 %101
%103 = bitcast float* %102 to <8 x i32>*
%wide.masked.load.7 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %103, i32 4, <8 x i1> %51, <8 x i32> undef), !tbaa !12, !alias.scope !16
%104 = getelementptr inbounds float, float* %3, i64 %101
%105 = bitcast float* %104 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.7, <8
... x i32>* %105, i32 4, <8 x i1> %51), !tbaa !12, !alias.scope !19, !noalias
... !16, !llvm.access.group !21
br label %region_for_end.i
```

```
pregion_for_entry.entry.i.us:
%local_id.x.0.us = phi i64 [ %i26, %if.end.r_exit.i.us.3 ], [ 0,
... %region_for_entry.entry.i.us.preheader ]
%i106 = add nuw nsw i64 %local_id.x.0.us, %i12
%conv.i.us = icmp slt i32 %i106 to i32
%cmp.i.us = icmp slt i32 %conv.i.us, %2
br i1 %cmp.i.us, label %if.then.i.us, label %if.end.r_exit.i.us
```

```
if.then.i.us:
%add8.i.us = add i32 %reass.mul.i, %conv.i.us
%idxprom.i.us = sext i32 %add8.i.us to i64
%arrayidx.i.us = getelementptr inbounds float, float* %5, i64 %idxprom.i.us
%i107 = bitcast float* %arrayidx.i.us to i32*
%i108 = load i32, i32* %i107, align 4, !tbaa !12
%arrayidx15.i.us = getelementptr inbounds float, float* %3, i64 %idxprom.i.us
%i109 = bitcast float* %arrayidx15.i.us to i32*
store i32 %i108, i32* %i109, align 4, !tbaa !12, !llvm.access.group !21
br label %if.end.r_exit.i.us
```

```
if.end.r_exit.i.us:
%i110 = or i64 %local_id.x.0.us, 1
%i111 = add nuw nsw i64 %i110, %i12
%conv.i.us.1 = trunc i64 %i111 to i32
%cmp.i.us.1 = icmp slt i32 %conv.i.us.1, %2
br i1 %cmp.i.us.1, label %if.then.i.us.1, label %if.end.r_exit.i.us.1
```

```
if.then.i.us.1:
%add8.i.us.1 = add i32 %reass.mul.i, %conv.i.us.1
%idxprom.i.us.1 = sext i32 %add8.i.us.1 to i64
%arrayidx.i.us.1 = getelementptr inbounds float, float* %5, i64
... %idxprom.i.us.1
%i113 = bitcast float* %arrayidx.i.us.1 to i32*
%i114 = load i32, i32* %i113, align 4, !tbaa !12
%arrayidx15.i.us.1 = getelementptr inbounds float, float* %3, i64
... %idxprom.i.us.1
%i115 = bitcast float* %arrayidx15.i.us.1 to i32*
store i32 %i114, i32* %i115, align 4, !tbaa !12, !llvm.access.group !21
br label %if.end.r_exit.i.us.1
```

```
if.end.r_exit.i.us.1:
%i116 = or i64 %local_id.x.0.us, 2
%i117 = add nuw nsw i64 %i116, %i12
%conv.i.us.2 = trunc i64 %i117 to i32
%cmp.i.us.2 = icmp slt i32 %conv.i.us.2, %2
br i1 %cmp.i.us.2, label %if.then.i.us.2, label %if.end.r_exit.i.us.2
```

```
if.then.i.us.2:
%add8.i.us.2 = add i32 %reass.mul.i, %conv.i.us.2
%idxprom.i.us.2 = sext i32 %add8.i.us.2 to i64
%arrayidx.i.us.2 = getelementptr inbounds float, float* %5, i64
... %idxprom.i.us.2
%i118 = bitcast float* %arrayidx.i.us.2 to i32*
%i119 = load i32, i32* %i118, align 4, !tbaa !12
%arrayidx15.i.us.2 = getelementptr inbounds float, float* %3, i64
... %idxprom.i.us.2
%i120 = bitcast float* %arrayidx15.i.us.2 to i32*
store i32 %i119, i32* %i120, align 4, !tbaa !12, !llvm.access.group !21
br label %if.end.r_exit.i.us.2
```

```
if.end.r_exit.i.us.2:
%i121 = or i64 %local_id.x.0.us, 3
%i122 = add nuw nsw i64 %i121, %i12
%conv.i.us.3 = trunc i64 %i122 to i32
%cmp.i.us.3 = icmp slt i32 %conv.i.us.3, %2
br i1 %cmp.i.us.3, label %if.then.i.us.3, label %if.end.r_exit.i.us.3
```

```
if.then.i.us.3:
%add8.i.us.3 = add i32 %reass.mul.i, %conv.i.us.3
%idxprom.i.us.3 = sext i32 %add8.i.us.3 to i64
%arrayidx.i.us.3 = getelementptr inbounds float, float* %5, i64
... %idxprom.i.us.3
%i123 = bitcast float* %arrayidx.i.us.3 to i32*
%i124 = load i32, i32* %i123, align 4, !tbaa !12
%arrayidx15.i.us.3 = getelementptr inbounds float, float* %3, i64
... %idxprom.i.us.3
%i125 = bitcast float* %arrayidx15.i.us.3 to i32*
store i32 %i124, i32* %i125, align 4, !tbaa !12, !llvm.access.group !21
br label %if.end.r_exit.i.us.3
```

```
if.end.r_exit.i.us.3:
%i126 = add nuw nsw i64 %local_id.x.0.us, 4
%exitcond.3 = icmp eq i64 %i126, 64
br i1 %exitcond.3, label %region_for_end.i.loopexit, label
... %region_for_entry.entry.i.us, !llvm.loop !26
```

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

```
pregion_for_end.i:
%i112 = add nuw nsw i64 %local_id.y0, 1
%exitcond3 = icmp eq i64 %i112, 32
br i1 %exitcond3, label %doitgen_kernel2.exit, label
... %region_for_entry.pregion_for_init.i, !llvm.loop !24
```

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