

```
%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, %i2
%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 %i2, i32 0
%broadcast.splat14 = shufflevector <8 x i32> %broadcast.splatinsert13, <8 x
... i32> undef, <8 x i32> zeroinitializer
%i21 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%i22 = add <8 x i32> %i21, <i32 0, i32 1, i32 2, i32 3, i32 4, i32 5, i32 6,
... i32 7>
%i23 = icmp sgt <8 x i32> %broadcast.splat14, %i22
%i24 = extractelement <8 x i32> %i22, i32 0
%i25 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%i26 = add <8 x i32> %i25, <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32
... 14, i32 15>
%i27 = icmp sgt <8 x i32> %broadcast.splat14, %i26
%i28 = extractelement <8 x i32> %i26, i32 0
%i29 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%i30 = add <8 x i32> %i29, <i32 16, i32 17, i32 18, i32 19, i32 20, i32 21,
... i32 22, i32 23>
%i31 = icmp sgt <8 x i32> %broadcast.splat14, %i30
%i32 = extractelement <8 x i32> %i30, i32 0
%i33 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%i34 = add <8 x i32> %i33, <i32 24, i32 25, i32 26, i32 27, i32 28, i32 29,
... i32 30, i32 31>
%i35 = icmp sgt <8 x i32> %broadcast.splat14, %i34
%i36 = extractelement <8 x i32> %i34, i32 0
%i37 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%i38 = add <8 x i32> %i37, <i32 32, i32 33, i32 34, i32 35, i32 36, i32 37,
... i32 38, i32 39>
%i39 = icmp sgt <8 x i32> %broadcast.splat14, %i38
%i40 = extractelement <8 x i32> %i38, i32 0
%i41 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%i42 = add <8 x i32> %i41, <i32 40, i32 41, i32 42, i32 43, i32 44, i32 45,
... i32 46, i32 47>
%i43 = icmp sgt <8 x i32> %broadcast.splat14, %i42
%i44 = extractelement <8 x i32> %i42, i32 0
%i45 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%i46 = add <8 x i32> %i45, <i32 48, i32 49, i32 50, i32 51, i32 52, i32 53,
... i32 54, i32 55>
%i47 = icmp sgt <8 x i32> %broadcast.splat14, %i46
%i48 = extractelement <8 x i32> %i46, i32 0
%i49 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%i50 = add <8 x i32> %i49, <i32 56, i32 57, i32 58, i32 59, i32 60, i32 61,
... i32 62, i32 63>
%i51 = icmp sgt <8 x i32> %broadcast.splat14, %i50
%i52 = extractelement <8 x i32> %i50, 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 ], [ %i12, %region_for_end.i ]
%i53 = add nuw nsw i64 %local_id.y0, %i13
%conv2.i = trunc i64 %i53 to i32
%cmp4.i = icmp slt i32 %conv2.i, %i1
%reass.add.i = add i32 %mul6.i, %conv2.i
%reass.mul.i = mul i32 %reass.add.i, %i2
br i1 %cmp4.i, label %vector.scevcheck, label %region_for_end.i
```

```
vector.scevcheck:
%i54 = trunc i64 %local_id.y0 to i32
%i55 = mul i32 %i54, %i2
%i56 = add i32 %i20, %i55
%i57 = icmp sgt i32 %i56, 2147483584
%brmerge = or i1 %i57, %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:
%i58 = add i32 %reass.mul.i, %i24
%i59 = sext i32 %i58 to i64
%i60 = getelementptr inbounds float, float* %5, i64 %i59
%i61 = bitcast float* %i60 to <8 x i32>
%wide.masked.load = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %i61, i32 4, <8 x i1> %i23, <8 x i32> undef), !tbaa !i2, !alias.scope !i16
%i62 = getelementptr inbounds float, float* %3, i64 %i59
%i63 = bitcast float* %i62 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load, <8 x
... i32>* %i63, i32 4, <8 x i1> %i23), !tbaa !i2, !alias.scope !i19, !noalias !i16,
... !llvm.access.group !i21
%i64 = add i32 %reass.mul.i, %i28
%i65 = sext i32 %i64 to i64
%i66 = getelementptr inbounds float, float* %5, i64 %i65
%i67 = bitcast float* %i66 to <8 x i32>*
%wide.masked.load.1 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %i67, i32 4, <8 x i1> %i27, <8 x i32> undef), !tbaa !i2, !alias.scope !i16
%i68 = getelementptr inbounds float, float* %3, i64 %i65
%i69 = bitcast float* %i68 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.1, <8
... x i32>* %i69, i32 4, <8 x i1> %i27), !tbaa !i2, !alias.scope !i19, !noalias !i16,
... !llvm.access.group !i21
%i70 = add i32 %reass.mul.i, %i32
%i71 = sext i32 %i70 to i64
%i72 = getelementptr inbounds float, float* %5, i64 %i71
%i73 = bitcast float* %i72 to <8 x i32>*
%wide.masked.load.2 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %i73, i32 4, <8 x i1> %i31, <8 x i32> undef), !tbaa !i2, !alias.scope !i16
%i74 = getelementptr inbounds float, float* %3, i64 %i71
%i75 = bitcast float* %i74 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.2, <8
... x i32>* %i75, i32 4, <8 x i1> %i31), !tbaa !i2, !alias.scope !i19, !noalias !i16,
... !llvm.access.group !i21
%i76 = add i32 %reass.mul.i, %i36
%i77 = sext i32 %i76 to i64
%i78 = getelementptr inbounds float, float* %5, i64 %i77
%i79 = bitcast float* %i78 to <8 x i32>*
%wide.masked.load.3 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %i79, i32 4, <8 x i1> %i35, <8 x i32> undef), !tbaa !i2, !alias.scope !i16
%i80 = getelementptr inbounds float, float* %3, i64 %i77
%i81 = bitcast float* %i80 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.3, <8
... x i32>* %i81, i32 4, <8 x i1> %i35), !tbaa !i2, !alias.scope !i19, !noalias !i16,
... !llvm.access.group !i21
%i82 = add i32 %reass.mul.i, %i40
%i83 = sext i32 %i82 to i64
%i84 = getelementptr inbounds float, float* %5, i64 %i83
%i85 = bitcast float* %i84 to <8 x i32>*
%wide.masked.load.4 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %i85, i32 4, <8 x i1> %i39, <8 x i32> undef), !tbaa !i2, !alias.scope !i16
%i86 = getelementptr inbounds float, float* %3, i64 %i83
%i87 = bitcast float* %i86 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.4, <8
... x i32>* %i87, i32 4, <8 x i1> %i39), !tbaa !i2, !alias.scope !i19, !noalias !i16,
... !llvm.access.group !i21
%i88 = add i32 %reass.mul.i, %i44
%i89 = sext i32 %i88 to i64
%i90 = getelementptr inbounds float, float* %5, i64 %i89
%i91 = bitcast float* %i90 to <8 x i32>*
%wide.masked.load.5 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %i91, i32 4, <8 x i1> %i43, <8 x i32> undef), !tbaa !i2, !alias.scope !i16
%i92 = getelementptr inbounds float, float* %3, i64 %i89
%i93 = bitcast float* %i92 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.5, <8
... x i32>* %i93, i32 4, <8 x i1> %i43), !tbaa !i2, !alias.scope !i19, !noalias !i16,
... !llvm.access.group !i21
%i94 = add i32 %reass.mul.i, %i48
%i95 = sext i32 %i94 to i64
%i96 = getelementptr inbounds float, float* %5, i64 %i95
%i97 = bitcast float* %i96 to <8 x i32>*
%wide.masked.load.6 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %i97, i32 4, <8 x i1> %i47, <8 x i32> undef), !tbaa !i2, !alias.scope !i16
%i98 = getelementptr inbounds float, float* %3, i64 %i95
%i99 = bitcast float* %i98 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.6, <8
... x i32>* %i99, i32 4, <8 x i1> %i47), !tbaa !i2, !alias.scope !i19, !noalias !i16,
... !llvm.access.group !i21
%i100 = add i32 %reass.mul.i, %i52
%i101 = sext i32 %i100 to i64
%i102 = getelementptr inbounds float, float* %5, i64 %i101
%i103 = bitcast float* %i102 to <8 x i32>*
%wide.masked.load.7 = call <8 x i32> @llvm.masked.load.v8i32.p0v8i32(<8 x
... i32>* %i103, i32 4, <8 x i1> %i51, <8 x i32> undef), !tbaa !i2, !alias.scope !i16
%i104 = getelementptr inbounds float, float* %3, i64 %i101
%i105 = bitcast float* %i104 to <8 x i32>*
call void @llvm.masked.store.v8i32.p0v8i32(<8 x i32> %wide.masked.load.7, <8
... x i32>* %i105, i32 4, <8 x i1> %i51), !tbaa !i2, !alias.scope !i19, !noalias
... !i16, !llvm.access.group !i21
br label %region_for_end.i
```

```
pregion_for_entry.entry.i.us:
%local_id.x.0.us = phi i64 [ %i126, %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 %conv.i.us, %i2
%cmp.i.us = icmp slt i32 %conv.i.us, %i2
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 !i2
%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 !i2, !llvm.access.group !i21
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, %i2
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 !i2
%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 !i2, !llvm.access.group !i21
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, %i2
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 !i2
%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 !i2, !llvm.access.group !i21
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, %i2
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 !i2
%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 !i2, !llvm.access.group !i21
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 %pregion_for_end.i.loopexit, label
... %pregion_for_entry.entry.i.us, !llvm.loop !i26
```

```
pregion_for_end.i.loopexit:
br label %pregion_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
... %pregion_for_entry.pregion_for_init.i, !llvm.loop !i24
```

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