

```
%i1:
%mul3.i = shl i64 %8, 5
%mul3.i.i = shl i64 %9, 3
%idxprom.i = sext i32 %4 to i64
%arrayidx.i = getelementptr inbounds float, float* %0, i64 %idxprom.i
%12 = trunc i64 %9 to i32
%13 = mul i32 %12, %6
%14 = shl i32 %13, 3
%15 = trunc i64 %8 to i32
%16 = shl i32 %15, 5
%17 = add i32 %14, %16
%scevgep = getelementptr float, float* %0, i64 %idxprom.i
%scevgep7 = bitcast float* %scevgep to i8*
%uglygep = getelementptr i8, i8* %scevgep7, i64 1
%18 = trunc i64 %9 to i32
%19 = mul i32 %18, %6
%20 = shl i32 %19, 3
%21 = trunc i64 %8 to i32
%22 = shl i32 %21, 5
%23 = add i32 %20, %22
%scevgep10 = getelementptr float, float* %2, i64 32
%24 = trunc i64 %9 to i32
%25 = mul i32 %24, %6
%26 = shl i32 %25, 3
%27 = trunc i64 %8 to i32
%28 = shl i32 %27, 5
%29 = add i32 %26, %28
%30 = zext i32 %6 to i64
%31 = shl i32 %24, 3
%32 = add i32 %31, -1
%33 = mul i32 %32, %6
%34 = add i32 %33, %28
%35 = trunc i64 %9 to i32
%36 = mul i32 %35, %6
%37 = shl i32 %36, 3
%38 = trunc i64 %8 to i32
%39 = shl i32 %38, 5
%40 = add i32 %37, %39
%41 = zext i32 %6 to i64
%scevgep33 = getelementptr float, float* %2, i64 32
%42 = shl i32 %35, 3
%43 = add i32 %42, -1
%44 = mul i32 %43, %6
%45 = add i32 %44, %39
%scevgep38 = getelementptr float, float* %3, i64 32
%scevgep43 = getelementptr float, float* %3, i64 32
br label %region_for_entry.region_for_init.i
```

```
region_for_entry.region_for_init.i:
%local_id.y.0 = phi i64 [ 0, %i1 ], [ %i06, %region_for_end.i ]
%46 = mul i64 %local_id.y.0, %41
%47 = trunc i64 %46 to i32
%48 = add i32 %40, %47
%49 = sext i32 %48 to i64
%scevgep31 = getelementptr float, float* %2, i64 %49
%scevgep34 = getelementptr float, float* %scevgep33, i64 %49
%50 = trunc i64 %46 to i32
%51 = add i32 %45, %50
%52 = sext i32 %51 to i64
%scevgep36 = getelementptr float, float* %3, i64 %52
%scevgep39 = getelementptr float, float* %scevgep38, i64 %52
%53 = mul i64 %local_id.y.0, %30
%54 = trunc i64 %53 to i32
%55 = add i32 %29, %54
%56 = trunc i64 %53 to i32
%57 = add i32 %34, %56
%58 = trunc i64 %local_id.y.0 to i32
%59 = mul i32 %58, %6
%60 = add i32 %59, %23
%61 = sext i32 %60 to i64
%scevgep8 = getelementptr float, float* %2, i64 %61
%scevgep89 = bitcast float* %scevgep8 to i8*
%scevgep11 = getelementptr float, float* %scevgep10, i64 %61
%62 = trunc i64 %local_id.y.0 to i32
%63 = mul i32 %62, %6
%64 = add i32 %63, %17
%add6.i.i = add nuw nsw i64 %local_id.y.0, %mul3.i.i
%conv2.i = trunc i64 %add6.i.i to i32
%cmp.i = icmp slt i32 %conv2.i, %5
%mul.i = mul nsw i32 %conv2.i, %6
%sub.i = add nsw i32 %conv2.i, -1
%mul22.i = mul nsw i32 %sub.i, %6
br label %cmp.i, label %region_for_entry.region_for_init.i.split.us, label ... %region_for_end.i
```

T F

```
region_for_entry.region_for_init.i.split.us:
%cmp6.i = icmp eq i32 %conv2.i, 0
br i1 %cmp6.i, label %vector.scevcheck, label %vector.scevcheck27
```

T F

```
vector.scevcheck27:
%65 = icmp sgt i32 %55, 2147483616
%66 = icmp sgt i32 %57, 2147483616
%67 = or i1 %65, %66
br i1 %67, label %region_for_entry.entry.i.us.preheader, label ... %vector.memcheck29
```

T F

```
vector.memcheck29:
%bound046 = icmp ult float* %scevgep31, %scevgep39
%bound147 = icmp ult float* %scevgep36, %scevgep34
%found.conflict48 = and i1 %bound046, %bound147
%bound049 = icmp ugt float* %scevgep43, %2
%bound150 = icmp ugt float* %scevgep33, %3
%found.conflict51 = and i1 %bound049, %bound150
%conflict.rdx = or i1 %found.conflict48, %found.conflict51
br i1 %conflict.rdx, label %region_for_entry.entry.i.us.preheader, label ... %vector.ph30
```

T F

```
vector.ph30:
%broadcast.splatinsert59 = 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.splatinsert13 = insertelement <8 x i32> undef, i32 %6, i32 0
%broadcast.splat14 = shufflevector <8 x i32> %broadcast.splatinsert13, <8 x
... i32> undef, <8 x i32> zeroinitializer
%broadcast.splatinsert15 = insertelement <8 x float> undef, float*
... %arrayidx.i, i32 0
%broadcast.splat16 = shufflevector <8 x float*> %broadcast.splatinsert15, <8
... x float*> undef, <8 x i32> zeroinitializer
br label %vector.body19
```

T F

```
vector.body19:
%index53 = phi i64 [ 0, %vector.ph30 ], [ %index.next54, %vector.body19 ]
%vec.ind57 = phi <8 x i64> [ <i64 0, i64 1, i64 2, i64 3, i64 4, i64 5, i64
... 6, i64 7>, %vector.ph30 ], [ %vec.ind.next58, %vector.body19 ]
%68 = add nuw nsw <8 x i64> %vec.ind57, %broadcast.splat60
%69 = trunc <8 x i64> %68 to <8 x i32>
%70 = icmp sgt <8 x i32> %broadcast.splat62, %69
%71 = extractelement <8 x i32> %69, i32 0
%72 = add nsw i32 %mul.i, %71
%73 = sext i32 %72 to i64
%74 = getelementptr inbounds float, float* %2, i64 %73
%75 = bitcast float* %74 to <8 x float*>
%wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %75, i32 4, <8 x i1> %70, <8 x float> undef), !tbaa !12, !alias.scope
... !16, !noalias !19
%76 = fpxext <8 x float> %wide.masked.load to <8 x double>
%77 = getelementptr inbounds float, float* %3, i64 %73
%78 = bitcast float* %77 to <8 x float*>
%wide.masked.load63 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %78, i32 4, <8 x i1> %70, <8 x float> undef), !tbaa !12, !alias.scope
... !23
%83 = fsub <8 x float> %wide.masked.load63, %wide.masked.load64
%84 = fpxext <8 x float> %83 to <8 x double>
%85 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %84, <8 x double>
... <double -5.000000e-01, double -5.000000e-01, double -5.000000e-01, double
... -5.000000e-01, double -5.000000e-01, double -5.000000e-01, double
... -5.000000e-01, double -5.000000e-01>, <8 x double> %76)
%86 = fptrunc <8 x double> %85 to <8 x float>
%87 = bitcast float* %74 to <8 x float*>
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %86, <8 x float>*
... %87, i32 4, <8 x i1> %70), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !24
%index.next54 = add i64 %index53, 8
%vec.ind.next58 = add <8 x i64> %vec.ind57, <i64 8, i64 8, i64 8, i64 8, i64
... 8, i64 8, i64 8, i64 8>
%88 = icmp eq i64 %index.next54, 32
br i1 %88, label %region_for_end.i.loopexit70, label %vector.body19,
... !llvm.loop !27
```

T F

```
region_for_entry.entry.i.us:
%local_id.x.0.us = phi i64 [ %i05, %if.end34.i.us ], [ 0,
... %region_for_entry.entry.i.us.preheader ]
%add1.i.i.us = add nuw nsw i64 %local_id.x.0.us, %mul.i.i
%conv1.us = trunc i64 %add1.i.i.us to i32
%cmp4.i.us = icmp slt i32 %conv1.us, %6
br i1 %cmp4.i.us, label %if.then3.i.us, label %if.end34.i.us
```

T F

```
if.else.i.us:
%add14.i.us = add nsw i32 %mul.i, %conv1.us
%idxprom15.i.us = sext i32 %add14.i.us to i64
%arrayidx16.i.us = getelementptr inbounds float, float* %2, i64
... %idxprom15.i.us
%101 = load float, float* %arrayidx16.i.us, align 4, !tbaa !12
%conv17.i.us = fpxext float %101 to double
%arrayidx21.i.us = getelementptr inbounds float, float* %3, i64
... %idxprom15.i.us
%102 = load float, float* %arrayidx21.i.us, align 4, !tbaa !12
%add23.i.us = add nsw i32 %mul22.i, %conv1.us
%idxprom24.i.us = sext i32 %add23.i.us to i64
%arrayidx25.i.us = getelementptr inbounds float, float* %3, i64
... %idxprom24.i.us
%103 = load float, float* %arrayidx25.i.us, align 4, !tbaa !12
%sub26.i.us = fsub float %102, %103
%conv27.i.us = fpxext float %sub26.i.us to double
%104 = tail call double @llvm.fmuladd.f64(double %conv27.i.us, double
... -5.000000e-01, double %conv17.i.us) #6
%conv29.i.us = fptrunc double %104 to float
store float %conv29.i.us, float* %arrayidx16.i.us, align 4, !tbaa !12,
... !llvm.access.group !24
br label %if.end34.i.us
```

T F

```
if.end34.i.us:
%105 = add nuw nsw i64 %local_id.x.0.us, 1
%exitcond4.not = icmp eq i64 %105, 32
br i1 %exitcond4.not, label %region_for_end.i.loopexit69, label
... %region_for_entry.entry.i.us, !llvm.loop !37
```

T F

region_for_end.i.loopexit69:
br label %region_for_end.i

```
region_for_end.i:
%106 = add nuw nsw i64 %local_id.y.0, 1
%exitcond4.not = icmp eq i64 %106, 8
br i1 %exitcond4.not, label %f.tdtd kernel1.exit, label
... %region_for_entry.region_for_init.i, !llvm.loop !38
```

T F

f.tdtd kernel1.exit:
ret void

```
vector.scevcheck:
%89 = icmp sgt i32 %64, 2147483616
br i1 %89, label %region_for_entry.entry.i.us.preheader, label
... %vector.memcheck
```

T F

```
vector.memcheck:
%bound0 = icmp ult float* %arrayidx.i, %scevgep11
%bound1 = icmp ugt i8* %uglygep, %scevgep89
%found.conflict = and i1 %bound0, %bound1
br i1 %found.conflict, label %region_for_entry.entry.i.us.preheader,
... label %vector.ph
```

T F

```
vector.ph:
%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.splatinsert13 = insertelement <8 x i32> undef, i32 %6, i32 0
%broadcast.splat14 = shufflevector <8 x i32> %broadcast.splatinsert13, <8 x
... i32> undef, <8 x i32> zeroinitializer
%broadcast.splatinsert15 = insertelement <8 x float> undef, float*
... %arrayidx.i, i32 0
%broadcast.splat16 = shufflevector <8 x float*> %broadcast.splatinsert15, <8
... x float*> undef, <8 x i32> zeroinitializer
br label %vector.body
```

T F

```
vector.body:
%index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
%vec.ind = phi <8 x i64> [ <i64 0, i64 1, i64 2, i64 3, i64 4, i64 5, i64 6,
... i64 7>, %vector.ph ], [ %vec.ind.next, %vector.body ]
%90 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
%91 = trunc <8 x i64> %90 to <8 x i32>
%92 = icmp sgt <8 x i32> %broadcast.splat14, %91
%wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
... x float> %broadcast.splat16, i32 4, <8 x i1> %92, <8 x float> undef), !tbaa
... !12, !alias.scope !30, !noalias !33
%93 = extractelement <8 x i32> %91, i32 0
%94 = add nuw nsw i32 %mul.i, %93
%95 = sext i32 %94 to i64
%96 = getelementptr inbounds float, float* %2, i64 %95
%97 = bitcast float* %96 to <8 x float*>
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %wide.masked.gather,
... <8 x float>* %97, i32 4, <8 x i1> %92), !tbaa !12, !alias.scope !33,
... !llvm.access.group !24
%index.next = add i64 %index, 8
%vec.ind.next = add <8 x i64> %vec.ind, <i64 8, i64 8, i64 8, i64 8, i64 8,
... i64 8, i64 8, i64 8>
%98 = icmp eq i64 %index.next, 32
br i1 %98, label %region_for_end.i.loopexit68, label %vector.body,
... !llvm.loop !35
```

T F

region_for_end.i.loopexit68:
br label %region_for_end.i

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

```
region_for_entry.entry.i.us.us:
%local_id.x.0.us.us = phi i64 [ %i00, %if.end34.i.us.us ], [ 0,
... %region_for_entry.entry.i.us.us.preheader ]
%add1.i.i.us.us = add nuw nsw i64 %local_id.x.0.us.us, %mul.i.i
%conv1.us.us = trunc i64 %add1.i.i.us.us to i32
%cmp4.i.us.us = icmp slt i32 %conv1.us.us, %6
br i1 %cmp4.i.us.us, label %if.then3.i.us.us, label %if.end34.i.us.us
```

T F

```
if.then8.i.us.us:
%99 = load float, float* %arrayidx.i, align 4, !tbaa !12
%add10.i.us.us = add nuw nsw i32 %mul.i, %conv1.us.us
%idxprom11.i.us.us = sext i32 %add10.i.us.us to i64
%arrayidx12.i.us.us = getelementptr inbounds float, float* %2, i64
... %idxprom11.i.us.us
store float %99, float* %arrayidx12.i.us.us, align 4, !tbaa !12,
... !llvm.access.group !24
br label %if.end34.i.us.us
```

T F

```
if.end34.i.us.us:
%100 = add nuw nsw i64 %local_id.x.0.us.us, 1
%exitcond3.not = icmp eq i64 %100, 32
br i1 %exitcond3.not, label %region_for_end.i.loopexit, label
... %region_for_entry.entry.i.us.us, !llvm.loop !36
```

T F

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