

```
vector.scevcheck:
%mul.i.i = shl i64 %6, 8
%mul.i = mul nsw i32 %4, %3
%sub.i = add nsw i32 %3, -1
%mul2.i = mul nsw i32 %sub.i, %4
%9 = mul i32 %4, %3
%10 = trunc i64 %6 to i32
%11 = shl i32 %10, 8
%12 = add i32 %9, %11
%13 = icmp sgt i32 %12, 2147483392
%14 = add i32 %3, -1
%15 = mul i32 %14, %4
%16 = add i32 %15, %11
%17 = icmp sgt i32 %16, 2147483392
%18 = or i1 %13, %17
br i1 %18, label %preregion_for_entry.entry.i.preheader, label %vector.memcheck
```

```
vector.memcheck:
%19 = mul i32 %4, %3
%20 = trunc i64 %6 to i32
%21 = shl i32 %20, 8
%22 = add i32 %19, %21
%23 = sext i32 %22 to i64
%scevgep = getelementptr float, float* %2, i64 %23
%24 = add nsw i64 %23, 256
%scevgep5 = getelementptr float, float* %2, i64 %24
%scevgep7 = getelementptr float, float* %1, i64 %23
%scevgep9 = getelementptr float, float* %1, i64 %24
%25 = add i32 %3, -1
%26 = mul i32 %25, %4
%27 = add i32 %26, %21
%28 = sext i32 %27 to i64
%scevgep11 = getelementptr float, float* %2, i64 %28
%29 = add nsw i64 %28, 256
%scevgep13 = getelementptr float, float* %2, i64 %29
%scevgep15 = getelementptr float, float* %0, i64 %23
%scevgep17 = getelementptr float, float* %0, i64 %24
%scevgep19 = getelementptr float, float* %1, i64 %28
%scevgep21 = getelementptr float, float* %1, i64 %29
%bound0 = icmp ult float* %scevgep, %scevgep9
%bound1 = icmp ult float* %scevgep7, %scevgep5
%found.conflict = and i1 %bound0, %bound1
%bound023 = icmp ult float* %scevgep, %scevgep13
%bound124 = icmp ult float* %scevgep11, %scevgep5
%found.conflict25 = and i1 %bound023, %bound124
%conflict.rdx = or i1 %found.conflict, %found.conflict25
%bound026 = icmp ult float* %scevgep, %scevgep17
%bound127 = icmp ult float* %scevgep15, %scevgep5
%found.conflict28 = and i1 %bound026, %bound127
%conflict.rdx29 = or i1 %conflict.rdx, %found.conflict28
%bound030 = icmp ult float* %scevgep, %scevgep21
%bound131 = icmp ult float* %scevgep19, %scevgep5
%found.conflict32 = and i1 %bound030, %bound131
%conflict.rdx33 = or i1 %conflict.rdx29, %found.conflict32
%bound034 = icmp ult float* %scevgep7, %scevgep13
%bound135 = icmp ult float* %scevgep11, %scevgep9
%found.conflict36 = and i1 %bound034, %bound135
%conflict.rdx37 = or i1 %conflict.rdx33, %found.conflict36
%bound038 = icmp ult float* %scevgep7, %scevgep17
%bound139 = icmp ult float* %scevgep15, %scevgep9
%found.conflict40 = and i1 %bound038, %bound139
%conflict.rdx41 = or i1 %conflict.rdx37, %found.conflict40
%bound042 = icmp ult float* %scevgep7, %scevgep21
%bound143 = icmp ult float* %scevgep19, %scevgep9
%found.conflict44 = and i1 %bound042, %bound143
%conflict.rdx45 = or i1 %conflict.rdx41, %found.conflict44
br i1 %conflict.rdx45, label %preregion_for_entry.entry.i.preheader, label ... %vector.ph
```

```
preregion_for_entry.entry.i.preheader:
br label %preregion_for_entry.entry.i
```

```
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.splatinsert46 = insertelement <8 x i32> undef, i32 %4, i32 0
%broadcast.splat47 = shufflevector <8 x i32> %broadcast.splatinsert46, <8 x
... i32> undef, <8 x i32> zeroinitializer
br label %vector.body
```

```
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 ]
%30 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
%31 = trunc <8 x i64> %30 to <8 x i32>
%32 = icmp sgt <8 x i32> %broadcast.splat47, %31
%33 = extractelement <8 x i32> %31, i32 0
%34 = add nsw i32 %mul.i, %33
%35 = sext i32 %34 to i64
%36 = getelementptr inbounds float, float* %2, i64 %35
%37 = bitcast float* %36 to <8 x float>*
%wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %37, i32 4, <8 x i1> %32, <8 x float> undef), !tbaa !12, !alias.scope
... !16, !noalias !19
%38 = add nsw i32 %mul2.i, %33
%39 = sext i32 %38 to i64
%40 = getelementptr inbounds float, float* %2, i64 %39
%41 = bitcast float* %40 to <8 x float>*
%wide.masked.load48 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %41, i32 4, <8 x i1> %32, <8 x float> undef), !tbaa !12, !alias.scope
... !24
%42 = getelementptr inbounds float, float* %0, i64 %35
%43 = bitcast float* %42 to <8 x float>*
%wide.masked.load49 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %43, i32 4, <8 x i1> %32, <8 x float> undef), !tbaa !12, !alias.scope
... !25
%44 = fmul <8 x float> %wide.masked.load48, %wide.masked.load49
%45 = getelementptr inbounds float, float* %1, i64 %39
%46 = bitcast float* %45 to <8 x float>*
%wide.masked.load50 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %46, i32 4, <8 x i1> %32, <8 x float> undef), !tbaa !12, !alias.scope
... !26
%47 = fdiv <8 x float> %44, %wide.masked.load50, !fpmath !27
%48 = fsub <8 x float> %wide.masked.load, %47
%49 = bitcast float* %36 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %48, <8 x float>*
... %49, i32 4, <8 x i1> %32), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !28
%50 = getelementptr inbounds float, float* %1, i64 %35
%51 = bitcast float* %50 to <8 x float>*
%wide.masked.load51 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %51, i32 4, <8 x i1> %32, <8 x float> undef), !tbaa !12, !alias.scope
... !30, !noalias !31
%52 = bitcast float* %42 to <8 x float>*
%wide.masked.load52 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %52, i32 4, <8 x i1> %32, <8 x float> undef), !tbaa !12, !alias.scope
... !25
%53 = fmul <8 x float> %wide.masked.load52, %wide.masked.load52
%54 = bitcast float* %45 to <8 x float>*
%wide.masked.load53 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %54, i32 4, <8 x i1> %32, <8 x float> undef), !tbaa !12, !alias.scope
... !26
%55 = fdiv <8 x float> %53, %wide.masked.load53, !fpmath !27
%56 = fsub <8 x float> %wide.masked.load51, %55
%57 = bitcast float* %50 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %56, <8 x float>*
... %57, i32 4, <8 x i1> %32), !tbaa !12, !alias.scope !30, !noalias !31,
... !llvm.access.group !28
%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>
%58 = icmp eq i64 %index.next, 256
br i1 %58, label %adi_kernel4.exit.loopexit55, label %vector.body,
... !llvm.loop !32
```

```
preregion_for_entry.entry.i:
% local_id_x.0 = phi i64 [ %66, %if.end_r_exit.i ], [ 0,
... %preregion_for_entry.entry.i.preheader ]
%add1.i.i = add nuw nsw i64 % local_id_x.0, %mul.i.i
%conv.i = trunc i64 %add1.i.i to i32
%cmp.i = icmp slt i32 %conv.i, %4
br i1 %cmp.i, label %if.then.i, label %if.end_r_exit.i
```

```
if.then.i:
%add.i = add nsw i32 %mul.i, %conv.i
%idxprom.i = sext i32 %add.i to i64
%arrayidx.i = getelementptr inbounds float, float* %2, i64 %idxprom.i
%59 = load float, float* %arrayidx.i, align 4, !tbaa !12
%add3.i = add nsw i32 %mul2.i, %conv.i
%idxprom4.i = sext i32 %add3.i to i64
%arrayidx5.i = getelementptr inbounds float, float* %2, i64 %idxprom4.i
%60 = load float, float* %arrayidx5.i, align 4, !tbaa !12
%arrayidx9.i = getelementptr inbounds float, float* %0, i64 %idxprom.i
%61 = load float, float* %arrayidx9.i, align 4, !tbaa !12
%mul10.i = fmul float %60, %61
%arrayidx15.i = getelementptr inbounds float, float* %1, i64 %idxprom4.i
%62 = load float, float* %arrayidx15.i, align 4, !tbaa !12
%div.i = fdiv float %mul10.i, %62, !fpmath !27
%sub16.i = fsub float %59, %div.i
store float %sub16.i, float* %arrayidx.i, align 4, !tbaa !12,
... !llvm.access.group !28
%arrayidx24.i = getelementptr inbounds float, float* %1, i64 %idxprom.i
%63 = load float, float* %arrayidx24.i, align 4, !tbaa !12
%64 = load float, float* %arrayidx9.i, align 4, !tbaa !12
%mul33.i = fmul float %64, %64
%65 = load float, float* %arrayidx15.i, align 4, !tbaa !12
%div39.i = fdiv float %mul33.i, %65, !fpmath !27
%sub40.i = fsub float %63, %div39.i
store float %sub40.i, float* %arrayidx24.i, align 4, !tbaa !12,
... !llvm.access.group !28
br label %if.end_r_exit.i
```

```
if.end_r_exit.i:
%66 = add nuw nsw i64 % local_id_x.0, 1
%exitcond.not = icmp eq i64 %66, 256
br i1 %exitcond.not, label %adi_kernel4.exit.loopexit, label
... %preregion_for_entry.entry.i, !llvm.loop !35
```

```
adi_kernel4.exit.loopexit:
br label %adi_kernel4.exit
```

```
adi_kernel4.exit.loopexit55:
br label %adi_kernel4.exit
```

```
adi_kernel4.exit:
ret void
```

CFG for ' _pool_kernel_adi_kernel4' function