

```
%8:
%9 = sext i32 %3 to i64
%10 = icmp slt i64 %9, 256
%11 = select i1 %10, i64 %9, i64 256
%mul.i.i = shl i64 %5, 8
%sub.i = add nsw i32 %3, -1
%12 = icmp ugt i64 %11, 1
%umax = select i1 %12, i64 %11, i64 1
%min.iters.check = icmp ult i64 %umax, 8
br i1 %min.iters.check, label %pregion_for_entry.entry.i.preheader, label
... %vector.scevcheck
```

T

F

```
vector.scevcheck:
%ident.check = icmp ne i32 %3, 1
%13 = add nsw i64 %umax, -1
%14 = trunc i64 %5 to i32
%15 = shl i32 %14, 8
%16 = trunc i64 %13 to i32
%17 = add i32 %15, %16
%18 = icmp slt i32 %17, %15
%19 = icmp ugt i64 %13, 4294967295
%20 = or i1 %18, %19
%21 = or i1 %ident.check, %20
br i1 %21, label %pregion_for_entry.entry.i.preheader, label %vector.memcheck
```

T

F

```
vector.memcheck:
%22 = trunc i64 %5 to i32
%23 = shl i32 %22, 8
%24 = sext i32 %23 to i64
%scevgep = getelementptr float, float* %2, i64 %24
%25 = add nsw i64 %umax, %24
%scevgep2 = getelementptr float, float* %2, i64 %25
%scevgep4 = getelementptr float, float* %1, i64 %24
%scevgep6 = getelementptr float, float* %1, i64 %25
%bound0 = icmp ult float* %scevgep, %scevgep6
%bound1 = icmp ult float* %scevgep4, %scevgep2
%found.conflict = and i1 %bound0, %bound1
br i1 %found.conflict, label %pregion_for_entry.entry.i.preheader, label
... %vector.ph
```

T

F

```
vector.ph:
%n.vec = and i64 %umax, -8
br label %vector.body
```

```
vector.body:
%index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
%26 = add i64 %index, %mul.i.i
%27 = trunc i64 %26 to i32
%28 = mul nsw i32 %27, %3
%29 = add nsw i32 %sub.i, %28
%30 = sext i32 %29 to i64
%31 = getelementptr inbounds float, float* %2, i64 %30
%32 = bitcast float* %31 to <8 x float>*
%wide.load = load <8 x float>, <8 x float>* %32, align 4, !tbaa !12,
... !alias.scope !16, !noalias !19
%33 = getelementptr inbounds float, float* %1, i64 %30
%34 = bitcast float* %33 to <8 x float>*
%wide.load8 = load <8 x float>, <8 x float>* %34, align 4, !tbaa !12,
... !alias.scope !19
%35 = fdiv <8 x float> %wide.load, %wide.load8, !fpmath !21
%36 = bitcast float* %31 to <8 x float>*
store <8 x float> %35, <8 x float>* %36, align 4, !tbaa !12, !alias.scope
... !16, !noalias !19, !llvm.access.group !22
%index.next = add i64 %index, 8
%37 = icmp eq i64 %index.next, %n.vec
br i1 %37, label %middle.block, label %vector.body, !llvm.loop !24
```

T

F

```
middle.block:
%cmp.n = icmp eq i64 %umax, %n.vec
br i1 %cmp.n, label %adi_kernel2.exit, label
... %pregion_for_entry.entry.i.preheader
```

T

F

```
pregion_for_entry.entry.i.preheader:
%_local_id_x.0.ph = phi i64 [ 0, %vector.memcheck ], [ 0, %vector.scevcheck
... ], [ 0, %8 ], [ %n.vec, %middle.block ]
br label %pregion_for_entry.entry.i
```

```
pregion_for_entry.entry.i:
%_local_id_x.0 = phi i64 [ %40, %pregion_for_entry.entry.i ], [
... %_local_id_x.0.ph, %pregion_for_entry.entry.i.preheader ]
%add1.i.i = add i64 %_local_id_x.0, %mul.i.i
%conv.i = trunc i64 %add1.i.i to i32
%mul.i = mul nsw i32 %conv.i, %3
%add.i = add nsw i32 %sub.i, %mul.i
%idxprom.i = sext i32 %add.i to i64
%arrayidx.i = getelementptr inbounds float, float* %2, i64 %idxprom.i
%38 = load float, float* %arrayidx.i, align 4, !tbaa !12
%arrayidx6.i = getelementptr inbounds float, float* %1, i64 %idxprom.i
%39 = load float, float* %arrayidx6.i, align 4, !tbaa !12
%div.i = fdiv float %38, %39, !fpmath !21
store float %div.i, float* %arrayidx.i, align 4, !tbaa !12,
... !llvm.access.group !22
%40 = add nuw i64 %_local_id_x.0, 1
%exitcond.not = icmp eq i64 %40, %umax
br i1 %exitcond.not, label %adi_kernel2.exit.loopexit, label
... %pregion_for_entry.entry.i, !llvm.loop !27
```

T

F

```
adi_kernel2.exit.loopexit:
br label %adi_kernel2.exit
```

```
adi_kernel2.exit:
ret void
```

CFG for '_pocl_kernel_adi_kernel2' function