```
%mul.i.i = shl i64 %5, 8
                                                                                                      %sub.i = add i32 %3, -2
                                                                                                      %cmp258.i = icmp sgt i32 %3, 2, !llvm.access.group !12
                                                                                                      %9 = sext i32 %sub.i to i64
                                                                                                      %wide.trip.count.i = zext i32 %sub.i to i64
                                                                                                      %min.iters.check.i = icmp ult i32 %sub.i, 8
                                                                                                      %10 = add nsw i64 %wide.trip.count.i, -1
                                                                                                       %11 = trunc i64 %10 to i32
                                                                                                      %12 = icmp ugt i64 %10, 4294967295
                                                                                                      %mul6.i = tail call { i64, i1 } @llvm.umul.with.overflow.i64(i64 %10, i64 4)
                                                                                                      ... #5
                                                                                                      %mul.result7.i = extractvalue { i64, i1 } %mul6.i, 0
                                                                                                      %mul.overflow8.i = extractvalue { i64, i1 } %mul6.i, 1
                                                                                                       %n.vec.i = and i64 %wide.trip.count.i, 4294967288
                                                                                                       %broadcast.splatinsert.i = insertelement <8 x i64> undef, i64 %9, i32 0
                                                                                                       %broadcast.splat.i = shufflevector <8 x i64> %broadcast.splatinsert.i, <8 x
                                                                                                      ... i64> undef, <8 x i32> zeroinitializer
                                                                                                      %broadcast.splatinsert36.i = insertelement <8 x i32> undef, i32 %3, i32 0
                                                                                                      %broadcast.splat37.i = shufflevector <8 x i32> %broadcast.splatinsert36.i,
                                                                                                      ... <8 x i32> undef, <8 x i32> zeroinitializer
                                                                                                      %cmp.n.i = icmp eq i64 %n.vec.i, %wide.trip.count.i
                                                                                                       br label %pregion for entry.entry.i
                                                                                                     pregion for entry.entry.i:
                                                                                                      .
__local_id_x.0 = phi i64 [ 0, %8 ], [ %68, %if.end.i ]
                                                                                                      %add1.i.i = add nuw nsw i64 % local id x.0, %mul.i.i, !llvm.access.group !12
                                                                                                      %conv.i = trunc i64 %add1.i.i to i32, !llvm.access.group !12
                                                                                                      %cmp.i = icmp slt i32 %conv.i, %3, !llvm.access.group !12
                                                                                                      %or.cond.i = and i1 %cmp258.i, %cmp.i, !llvm.access.group !12
                                                                                                      br i1 %or.cond.i, label %for.body.lr.ph.i, label %if.end.i,
                                                                                                      ...!llvm.access.group!12
                                                                                                                                                                                   F
                             for.body.lr.ph.i:
                             %mul.i = mul nsw i32 %conv.i, %3, !llvm.access.group !12
                             %sub21.i = add i32 %mul.i, %3, !llvm.access.group !12
                             %sub22.i = add i32 %sub21.i, -3, !llvm.access.group !12
                             %13 = sext i32 %mul.i to i64, !llvm.access.group !12
                             %sub9.i = add i32 %mul.i, -1, !llvm.access.group !12
                             br i1 %min.iters.check.i, label %for.body.i.preheader, label
                             ... %vector.scevcheck.i, !llvm.access.group !12
                                   vector.scevcheck.i:
                                    %14 = add i32 %sub.i, %mul.i, !llvm.access.group !12
                                    %15 = sub i32 %14, %11, !llvm.access.group !12
                                    %16 = icmp sgt i32 %15, %14, !llvm.access.group !12
                                    %17 = sub i32 %sub22.i, %11, !llvm.access.group !12
                                    %18 = icmp sqt i32 %17, %sub22.i, !llvm.access.group !12
                                    %19 = or i1 %12, %18, !llvm.access.group !12
                                    %20 = or i1 %19, %16, !llvm.access.group !12
                                    %21 = sext i32 %14 to i64, !llvm.access.group !12
                                    %scevgep.i = getelementptr float, float* %2, i64 %21, !llvm.access.group !12
                                    %scevgep5.i = ptrtoint float* %scevgep.i to i64, !llvm.access.group !12
                                    %22 = icmp ugt i64 %mul.result7.i, %scevgep5.i, !llvm.access.group !12
                                    %23 = or i1 %mul.overflow8.i, %22, !llvm.access.group !12
                                    %24 = or i1 %20, %23, !llvm.access.group !12
                                    %25 = add nsw i64 %13, %9, !llvm.access.group !12
                                    %scevgep9.i = getelementptr float, float* %2, i64 %25, !llvm.access.group !12
                                    %scevgep910.i = ptrtoint float* %scevgep9.i to i64, !llvm.access.group !12
                                    %26 = icmp ugt i64 %mul.result7.i, %scevgep910.i, !llvm.access.group !12
                                    %27 = or i1 %26, %24, !llvm.access.group !12
                                    %28 = sext i32 %sub22.i to i64, !llvm.access.group !12
                                    %scevgep14.i = getelementptr float, float* %2, i64 %28, !llvm.access.group
                                    %scevgep1415.i = ptrtoint float* %scevgep14.i to i64, !llvm.access.group !12
                                    %29 = icmp ugt i64 %mul.result7.i, %scevgep1415.i, !llvm.access.group !12
                                    %30 = or i1 %29, %27, !llvm.access.group !12
                                    %scevgep19.i = getelementptr float, float* %0, i64 %28, !llvm.access.group
                                    %scevgep1920.i = ptrtoint float* %scevgep19.i to i64, !llvm.access.group !12
                                    %31 = icmp ugt i64 %mul.result7.i, %scevgep1920.i, !llvm.access.group !12
                                    %32 = or i1 %mul.overflow8.i, %31, !llvm.access.group !12
                                    %33 = or i1 %32, %30, !llvm.access.group !12
                                    %scevgep24.i = getelementptr float, float* %1, i64 %28, !llvm.access.group
                                    ...!12
                                    %scevgep2425.i = ptrtoint float* %scevgep24.i to i64, !llvm.access.group !12
                                    %34 = icmp ugt i64 %mul.result7.i, %scevgep2425.i, !llvm.access.group !12
                                    %35 = or i1 %34, %33, !llvm.access.group !12
                                    br i1 %35, label %for.body.i.preheader, label %vector.ph.i,
                                    ... !llvm.access.group !12
                                                                                                                   F
                                                          vector.ph.i:
                                                           %broadcast.splatinsert29.i = insertelement <8 x i64> undef, i64 %13, i32 0,
                                                           ..!llvm.access.group!12
                                                          %broadcast.splat30.i = shufflevector <8 x i64> %broadcast.splatinsert29.i,
                                                           .. <8 x i64> undef, <8 x i32> zeroinitializer, !llvm.access.group !12
                                                          %broadcast.splatinsert31.i = insertelement <8 x i32> undef, i32 %sub9.i, i32
                                                           .. 0, !llvm.access.group !12
                                                           %broadcast.splat32.i = shufflevector <8 x i32> %broadcast.splatinsert31.i,
                                                           .. <8 x i32> undef, <8 x i32> zeroinitializer, !llvm.access.group !12
                                                           %broadcast.splatinsert38.i = insertelement <8 x i32> undef, i32 %mul.i, i32
                                                           .. 0, !llvm.access.group !12
                                                          %broadcast.splat39.i = shufflevector <8 x i32> %broadcast.splatinsert38.i,
                                                           .. <8 x i32> undef, <8 x i32> zeroinitializer, !llvm.access.group !12
                                                          %broadcast.splatinsert41.i = insertelement <8 x i32> undef, i32 %sub22.i,
                                                           .. i32 0, !llvm.access.group !12
                                                          %broadcast.splat42.i = shufflevector <8 x i32> %broadcast.splatinsert41.i,
                                                           .. <8 x i32> undef, <8 x i32> zeroinitializer, !llvm.access.group !12
                                                          br label %vector.body.i, !llvm.access.group !12
                                                   vector.body.i:
                                                    %vec.ind.next35.i5 = phi < 8 \times i32 > [%vec.ind.next35.i, %vector.body.i ], [
                                                      <i32 0, i32 1, i32 2, i32 3, i32 4, i32 5, i32 6, i32 7>, %vector.ph.i ]
                                                    %vec.ind.next.i3 = phi < 8 \times i64 > [%vec.ind.next.i, %vector.body.i ], [< i64
                                                    ... 0, i64 1, i64 2, i64 3, i64 4, i64 5, i64 6, i64 7>, %vector.ph.i ]
                                                    %index.next.i1 = phi i64 [ %index.next.i, %vector.body.i ], [ 0,
                                                    ... %vector.ph.i ]
                                                    %36 = sub nsw <8 x i64> %broadcast.splat.i, %vec.ind.next.i3,
                                                    ...!llvm.access.group!12
                                                    %37 = add nsw <8 x i64> %36, %broadcast.splat30.i, !llvm.access.group !12
                                                    %38 = \text{getelementptr inbounds float, float* } \%2, <8 \times i64 > \%37,
                                                    ..!llvm.access.group!12
                                                    %wide.masked.gather.i = tail call <8 x float>
                                                    ... @llvm.masked.gather.v8f32.v8p0f32(< 8 \times float* > %38, i32 4, < 8 \times i1 > < i1 true,
                                                    ... i1 true, i1 true, i1 true, i1 true, i1 true, i1 true>, <8 x float>
                                                    ... undef) #5, !tbaa !14, !llvm.access.group !12
                                                    %39 = trunc <8 x i64> %36 to <8 x i32>, !llvm.access.group !12
                                                    %40 = add <8 x i32> %broadcast.splat32.i, %39, !llvm.access.group !12
                                                    %41 = \text{sext} < 8 \times i32 > %40 \text{ to } < 8 \times i64 >, !llvm.access.group !12
                                                    \%42 = getelementptr inbounds float, float* \%2, <8 x i64> \%41,
                                                    ...!llvm.access.group!12
                                                    %wide.masked.gather33.i = tail call <8 x float>
                                                    ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %42, i32 4, <8 x i1> <i1 true,
                                                    ... i1 true, i1 true, i1 true, i1 true, i1 true, i1 true, i1 true>, <8 x float>
                                                    ... undef) #5, !tbaa !14, !llvm.access.group !12
                                                    \%43 = \text{sub} < 8 \times i32 > \%broadcast.splat37.i, \%vec.ind.next35.i5,
                                                    ...!llvm.access.group!12
                                                    %44 = add < 8 \times i32 > %43, %broadcast.splat39.i, !llvm.access.group !12
                                                    %45 = add < 8 \times i32 > %44, < i32 - 3, i32 - 3,
                                                    ... i32 -3, i32 -3>, !llvm.access.group !12
                                                    %46 = \text{sext} < 8 \times i32 > %45 \text{ to } < 8 \times i64 >, !llvm.access.group !12
                                                    %47 = \text{getelementptr inbounds float}, \text{ float* } %0, <8 \text{ x } i64 > \%46,
                                                    ...!llvm.access.group!12
                                                    %wide.masked.gather40.i = tail call <8 x float>
                                                    ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %47, i32 4, <8 x i1> <i1 true,
                                                    ... i1 true, i1 true, i1 true, i1 true, i1 true, i1 true, i1 true>, <8 x float>
                                                    ... undef) #5, !tbaa !14, !llvm.access.group !12
                                                    %48 = fneg <8 x float> %wide.masked.gather33.i, !llvm.access.group !12
                                                    \%49 = \text{tail call} < 8 \text{ x float} > \text{@llvm.fmuladd.v8f32} (< 8 \text{ x float} > \%48, < 8 \text{ x float} >
                                                    ... %wide.masked.gather40.i, <8 x float> %wide.masked.gather.i) #5,
                                                    ...!llvm.access.group!12
                                                    \%50 = \text{sub} < 8 \times \text{i}32 > \%broadcast.splat42.i, \%vec.ind.next35.i5,
                                                    ...!llvm.access.group!12
                                                    %51 = \text{sext} < 8 \times \text{i} 32 > \%50 \text{ to} < 8 \times \text{i} 64 >, !llvm.access.group !12
                                                    \%52 = getelementptr inbounds float, float* \%1, <8 \times i64 > \%51,
                                                    ...!llvm.access.group!12
                                                    %wide.masked.gather43.i = tail call <8 x float>
                                                    ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %52, i32 4, <8 x i1> <i1 true,
                                                    ... i1 true, i1 true, i1 true, i1 true, i1 true, i1 true, i1 true>, <8 x float>
                                                    ... undef) #5, !tbaa !14, !llvm.access.group !12
                                                    %53 = fdiv <8 x float> %49, %wide.masked.gather43.i, !fpmath !18,
                                                    ...!llvm.access.group!12
                                                    \%54 = \text{add} < 8 \times i32 > \%44, < i32 - 2, i32 - 2
                                                    ... i32 -2, i32 -2>, !llvm.access.group !12
                                                    \%55 = \text{sext} < 8 \times i32 > \%54 \text{ to } < 8 \times i64 >, !llvm.access.group !12
                                                    %56 = getelementptr inbounds float, float* %2, <8 x i64> %55,
                                                    ...!llvm.access.group!12
                                                    tail call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %53, <8 x
                                                    ... float*> %56, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1
                                                    ... true, i1 true, i1 true>) #5, !tbaa !14, !llvm.access.group !12
                                                    %index.next.i = add i64 %index.next.i1, 8, !llvm.access.group !12
                                                    %vec.ind.next.i = add <8 x i64> %vec.ind.next.i3, <i64 8, i64 8, i64 8, i64
                                                    ... 8, i64 8, i64 8, i64 8, i64 8>, !llvm.access.group !12
                                                    %vec.ind.next35.i = add <8 x i32> %vec.ind.next35.i5, <i32 8, i32 8,
                                                    ... i32 8, i32 8, i32 8, i32 8, i32 8>, !llvm.access.group !12
                                                    %57 = icmp eq i64 %index.next.i, %n.vec.i, !llvm.access.group !12
                                                    br i1 %57, label %middle.block.i, label %vector.body.i, !llvm.loop !19,
                                                    ...!llvm.access.group!12
                                                    middle.block.i:
                                                    br i1 %cmp.n.i, label %if.end.i, label %for.body.i.preheader,
                                                    ...!llvm.access.group!12
        for.body.i.preheader:
         %indvars.iv.next.i7.ph = phi i64 [ 0, %for.body.lr.ph.i ], [ 0,
         ... %vector.scevcheck.i ], [ %n.vec.i, %middle.block.i ]
         br label %for.body.i
for.body.i:
%indvars.iv.next.i7 = phi i64 [ %indvars.iv.next.i, %for.body.i ], [
... %indvars.iv.next.i7.ph, %for.body.i.preheader ]
%58 = sub nsw i64 %9, %indvars.iv.next.i7, !llvm.access.group !12
%59 = add nsw i64 %58, %13, !llvm.access.group !12
%arrayidx.i = getelementptr inbounds float, float* %2, i64 %59,
...!llvm.access.group!12
%60 = load float, float* %arrayidx.i, align 4, !tbaa !14, !llvm.access.group
%61 = trunc i64 %58 to i32, !llvm.access.group !12
%add10.i = add i32 %sub9.i, %61, !llvm.access.group !12
%idxprom11.i = sext i32 %add10.i to i64, !llvm.access.group !12
%arrayidx12.i = getelementptr inbounds float, float* %2, i64 %idxprom11.i,
...!llvm.access.group!12
%62 = load float, float* %arrayidx12.i, align 4, !tbaa !14,
...!llvm.access.group!12
%63 = trunc i64 %indvars.iv.next.i7 to i32, !llvm.access.group !12
%64 = sub i32 %3, %63, !llvm.access.group !12
%sub15.i = add i32 %64, %mul.i, !llvm.access.group !12
%add16.i = add i32 %sub15.i, -3, !llvm.access.group !12
%idxprom17.i = sext i32 %add16.i to i64, !llvm.access.group !12
%arrayidx18.i = getelementptr inbounds float, float* %0, i64 %idxprom17.i,
...!llvm.access.group!12
%65 = load float, float* %arrayidx18.i, align 4, !tbaa !14,
...!llvm.access.group!12
%neg.i = fneg float %62, !llvm.access.group !12
%66 = tail call float @llvm.fmuladd.f32(float %neg.i, float %65, float %60)
... #5, !llvm.access.group !12
%add23.i = sub i32 %sub22.i, %63, !llvm.access.group !12 %idxprom24.i = sext i32 %add23.i to i64, !llvm.access.group !12
%arrayidx25.i = getelementptr inbounds float, float* %1, i64 %idxprom24.i,
...!llvm.access.group!12
%67 = load float, float* %arrayidx25.i, align 4, !tbaa !14,
...!llvm.access.group!12
%div.i = fdiv float %66, %67, !fpmath !18, !llvm.access.group !12
%add29.i = add i32 %sub15.i, -2, !llvm.access.group !12
%idxprom30.i = sext i32 %add29.i to i64, !llvm.access.group !12
%arrayidx31.i = getelementptr inbounds float, float* %2, i64 %idxprom30.i,
...!llvm.access.group!12
store float %div.i, float* %arrayidx31.i, align 4, !tbaa !14,
...!llvm.access.group!12
%indvars.iv.next.i = add nuw nsw i64 %indvars.iv.next.i7, 1,
...!llvm.access.group!12
%exitcond.not.i = icmp eq i64 %indvars.iv.next.i, %wide.trip.count.i,
...!llvm.access.group!12
br i1 %exitcond.not.i, label %if.end.i.loopexit, label %for.body.i,
...!llvm.loop!22,!llvm.access.group!12
                                                            if.end.i.loopexit:
                                                             br label %if.end.i
                                                                                                             if.end.i:
                                                                                                             \%68 = \text{add nuw nsw } i64 \% \text{ local id } x.0, 1
                                                                                                             %exitcond.not = icmp eq \overline{164} %68, \overline{256}
                                                                                                             br i1 %exitcond.not, label %adi kernel3.exit, label
                                                                                                             ... %pregion for entry.entry.i, !llvm.loop !23
                                                                                                                  adi_kernel3.exit:
```

%8: