%mul.i.i = shl i64 %5, 8 %cmp288.i = icmp sgt i32 %3, 1 %wide.trip.count.i = zext i32 %3 to i64 %9 = add nsw i32 %3, -2br i1 %cmp288.i, label %pregion for entry.entry.i.us.preheader, label ... %adi kernel1.exit pregion for entry.entry.i.us.preheader: %scevgep13 = getelementptr float, float* %2, i64 %wide.trip.count.i %scevgep18 = getelementptr float, float* %1, i64 %wide.trip.count.i %10 = add nsw i64 %wide.trip.count.i, -2 %11 = trunc i64 %5 to i32 %12 = mul i32 %11, %3 %13 = shl i32 %12, 8 %14 = trunc i64 %5 to i32 %15 = mul i32 %14, %3 %16 = shl i32 %15, 8br label %pregion for entry.entry.i.us pregion for entry.entry.i.us: % local id x.0.us = phi i64 [%52, %if.end.i.us], [0, ... %pregion_for_entry.entry.i.us.preheader] $\%17 = \text{trunc} \text{ } \text{i}64 \% \text{_local_id_x.0.us to i}32$ %18 = mul i 32 %17, %3%19 = add i32 %18, %16 %20 = sext i 32 % 19 to i 64%scevgep22 = getelementptr float, float* %2, i64 %20 %scevgep23 = getelementptr float, float* %1, i64 %20 %21 = trunc i64 % local id x.0.us to i32 $%22 = \text{mul i} 32 \% 2\overline{1}, \% 3$ %23 = add i32 %22, %13 %add1.i.i.us = add nuw nsw i64 % local id x.0.us, %mul.i.i $%conv.i.us = trunc i64 %add1.i.i.us to i3\overline{2}$ %cmp.i.us = icmp slt i32 %conv.i.us, %3 br i1 %cmp.i.us, label %for.body.lver.check.i.us, label %if.end.i.us for.body.lver.check.i.us: %mul.i.us = mul i32 %conv.i.us, %3 %24 = sext i32 %mul.i.us to i64 %scevgep.i.us = getelementptr float, float* %2, i64 %24 %25 = add nsw i64 %24, %wide.trip.count.i %scevgep92.i.us = getelementptr float, float* %2, i64 %25 %scevgep94.i.us = getelementptr float, float* %1, i64 %24 %scevgep96.i.us = getelementptr float, float* %1, i64 %25 %bound0.i.us = icmp ult float* %scevgep.i.us, %scevgep96.i.us %bound1.i.us = icmp ult float* %scevgep94.i.us, %scevgep92.i.us %found.conflict.i.us = and i1 %bound0.i.us, %bound1.i.us %26 = add i32 %9, %mul.i.us %27 = icmp slt i32 %26, %mul.i.us %lver.safe.i.us = or i1 %27, %found.conflict.i.us br i1 %lver.safe.i.us, label %for.body.lver.orig.lver.orig.i.us.lver.check, ... label %for.body.ph.i.us for.body.lver.orig.lver.orig.i.us.lver.check: %sub.lver.orig.i.us = add i32 %mul.i.us, -1 %bound0 = icmp ugt float* %scevgep18, %2 %bound1 = icmp ugt float* %scevgep13, %1 %found.conflict = and i1 %bound0, %bound1 %33 = trunc i64 %10 to i32 for.body.ph.i.us: %34 = add i32 %23, %33 %load initial.i.us = load float, float* %scevgep.i.us, align 4 %35 = icmp slt i32 %34, %23 %36 = icmp ugt i64 %10, 4294967295 %37 = or i1 %35, %36 %load initial102.i.us = load float, float* %scevgep94.i.us, align 4 br label %for.body.i.us %lver.safe = or i1 %found.conflict, %37 br i1 %lver.safe, label ... %for.body.lver.orig.lver.orig.i.us.lver.orig.preheader, label ... %for.body.lver.orig.lver.orig.i.us.ph Τ F for.body.i.us: %indvars.iv.next.i6.us = phi i64 [%indvars.iv.next.i.us, %for.body.i.us], ... [1, %for.body.ph.i.us] %sub18.i4.us = phi float [%sub18.i.us, %for.body.i.us], [... %load_initial.i.us, %for.body.ph.i.us] %sub42.i2.us = phi float [%sub42.i.us, %for.body.i.us], [... %load_initial102.i.us, %for.body.ph.i.us] %28 = add nsw i64 %indvars.iv.next.i6.us, %24 %arrayidx.i.us = getelementptr inbounds float, float* %2, i64 %28 %29 = load float, float* %arrayidx.i.us, align 4, !tbaa !12 %arrayidx11.i.us = getelementptr inbounds float, float* %0, i64 %28 %30 = load float, float* %arrayidx11.i.us, align 4, !tbaa !12 %mul12.i.us = fmul float %sub18.i4.us, %30 %div.i.us = fdiv float %mul12.i.us, %sub42.i2.us, !fpmath !16 for.body.lver.orig.lver.orig.i.us.ph: %sub18.i.us = fsub float %29, %div.i.us %load initial = load float, float* %scevgep22, align 4 for.body.lver.orig.lver.orig.i.us.lver.orig.preheader: br label %for.body.lver.orig.lver.orig.i.us.lver.orig store float %sub18.i.us, float* %arrayidx.i.us, align 4, !tbaa !12, %load initial24 = load float, float* %scevgep23, align 4 ..!llvm.access.group!17 br label %for.body.lver.orig.lver.orig.i.us %arrayidx26.i.us = getelementptr inbounds float, float* %1, i64 %28 %31 = load float, float* %arrayidx26.i.us, align 4, !tbaa !12 %32 = load float, float* %arrayidx11.i.us, align 4, !tbaa !12 %mul35.i.us = fmul float %32, %32 %div41.i.us = fdiv float %mul35.i.us, %sub42.i2.us, !fpmath !16 %sub42.i.us = fsub float %31, %div41.i.us store float %sub42.i.us, float* %arrayidx26.i.us, align 4, !tbaa !12, ...!llvm.access.group!17 %indvars.iv.next.i.us = add nuw nsw i64 %indvars.iv.next.i6.us, 1 %exitcond.not.i.us = icmp eq i64 %indvars.iv.next.i.us, %wide.trip.count.i br i1 %exitcond.not.i.us, label %if.end.i.us.loopexit31, label .. %for.body.i.us, !llvm.loop !19 for.body.lver.orig.lver.orig.i.us.lver.orig: %indvars.iv.next.lver.orig.lver.orig.i8.us.lver.orig = phi i64 [... %indvars.iv.next.lver.orig.lver.orig.i.us.lver.orig, ... %for.body.lver.orig.lver.orig.i.us.lver.orig], [1, ... %for.body.lver.orig.lver.orig.i.us.lver.orig.preheader] %38 = add nsw i64 %indvars.iv.next.lver.orig.lver.orig.i8.us.lver.orig, %24 for.body.lver.orig.lver.orig.i.us: %store forwarded25 = phi float [%load initial24, %arrayidx.lver.orig.lver.orig.i.us.lver.orig = getelementptr inbounds float, .. float* %2, i64 %38 .. %for.body.lver.orig.lver.orig.i.us.ph], [%sub42.lver.orig.lver.orig.i.us, %39 = load float, float* %arrayidx.lver.orig.lver.orig.i.us.lver.orig, align .. %for.body.lver.orig.lver.orig.i.us] %store forwarded = phi float [%load initial, ... 4, !tbaa !12 ... %for.body.lver.orig.lver.orig.i.us.ph J, [%sub18.lver.orig.lver.orig.i.us, %40 = trunc i64 %indvars.iv.next.lver.orig.lver.orig.i8.us.lver.orig to i32 ... %for.body.lver.orig.lver.orig.i.us] %add5.lver.orig.lver.orig.i.us.lver.orig = add i32 %sub.lver.orig.i.us, %40 %indvars.iv.next.lver.orig.lver.orig.i8.us = phi i64 [1, %idxprom6.lver.orig.lver.orig.i.us.lver.orig = sext i32 ... %for.body.lver.orig.lver.orig.i.us.ph], [... %add5.lver.orig.lver.orig.i.us.lver.orig to i64 ... %indvars.iv.next.lver.orig.lver.orig.i.us, %for.body.lver.orig.lver.orig.i.us %arrayidx7.lver.orig.lver.orig.i.us.lver.orig = getelementptr inbounds ... float, float* %2, i64 %idxprom6.lver.orig.lver.orig.i.us.lver.orig %41 = load float, float* %arrayidx7.lver.orig.lver.orig.i.us.lver.orig, %47 = add nsw i64 %indvars.iv.next.lver.orig.lver.orig.i8.us, %24 ... align 4, !tbaa !12 %arrayidx.lver.orig.lver.orig.i.us = getelementptr inbounds float, float* %arrayidx11.lver.orig.lver.orig.i.us.lver.orig = getelementptr inbounds .. %2, i64 %47 ... float, float* %0, i64 %38 %48 = load float, float* %arrayidx.lver.orig.lver.orig.i.us, align 4, !tbaa %42 = load float, float* %arrayidx11.lver.orig.lver.orig.i.us.lver.orig, ... align 4, !tbaa !12 %arrayidx11.lver.orig.lver.orig.i.us = getelementptr inbounds float, float* .. %0, i64 %47 %mul12.lver.orig.lver.orig.i.us.lver.orig = fmul float %41, %42 %arrayidx17.lver.orig.lver.orig.i.us.lver.orig = getelementptr inbounds %49 = load float, float* %arrayidx11.lver.orig.lver.orig.i.us, align 4, ... float, float* %1, i64 %idxprom6.lver.orig.lver.orig.i.us.lver.orig ... !tbaa !12 %43 = load float, float* %arrayidx17.lver.orig.lver.orig.i.us.lver.orig, %mul12.lver.orig.lver.orig.i.us = fmul float %store forwarded, %49 %div.lver.orig.lver.orig.i.us = fdiv float %mul12.lver.orig.lver.orig.i.us, ... align 4, !tbaa !12 %div.lver.orig.lver.orig.i.us.lver.orig = fdiv float .. %store forwarded25, !fpmath !16 %sub18.lver.orig.lver.orig.i.us = fsub float %48, ... %mul12.lver.orig.lver.orig.i.us.lver.orig, %43, !fpmath !16 ... %div.lver.orig.lver.orig.i.us %sub18.lver.orig.lver.orig.i.us.lver.orig = fsub float %39, store float %sub18.lver.orig.lver.orig.i.us, float* ... %div.lver.orig.lver.orig.i.us.lver.orig store float %sub18.lver.orig.lver.orig.i.us.lver.orig, float* ... %arrayidx.lver.orig.lver.orig.i.us, align 4, !tbaa !12, !llvm.access.group !17 ... %arrayidx.lver.orig.lver.orig.i.us.lver.orig, align 4, !tbaa !12, %arrayidx26.lver.orig.lver.orig.i.us = getelementptr inbounds float, float* ...!llvm.access.group!17 .. %1, i64 %47 %arrayidx26.lver.orig.lver.orig.i.us.lver.orig = getelementptr inbounds %50 = load float, float* %arrayidx26.lver.orig.lver.orig.i.us, align 4, ... float, float* %1, i64 %38 ..!tbaa!12 %44 = load float, float* %arrayidx26.lver.orig.lver.orig.i.us.lver.orig, %51 = load float, float* %arrayidx11.lver.orig.lver.orig.i.us, align 4, ... align 4, !tbaa !12 ... !tbaa !12 %45 = load float, float* %arrayidx11.lver.orig.lver.orig.i.us.lver.orig, %mul35.lver.orig.lver.orig.i.us = fmul float %51, %51 ... align 4, !tbaa !12 %div41.lver.orig.lver.orig.i.us = fdiv float %mul35.lver.orig.lver.orig.i.us.lver.orig = fmul float %45, %45 .. %mul35.lver.orig.lver.orig.i.us, %store forwarded25, !fpmath !16 %46 = load float, float* %arrayidx17.lver.orig.lver.orig.i.us.lver.orig, %sub42.lver.orig.lver.orig.i.us = fsub float %50, ... %div41.lver.orig.lver.orig.i.us ... align 4, !tbaa !12 store float %sub42.lver.orig.lver.orig.i.us, float* %div41.lver.orig.lver.orig.i.us.lver.orig = fdiv float ... %arrayidx26.lver.orig.lver.orig.i.us, align 4, !tbaa !12, !llvm.access.group ... %mul35.lver.orig.lver.orig.i.us.lver.orig, %46, !fpmath !16 %sub42.lver.orig.lver.orig.i.us.lver.orig = fsub float %44, ... %div41.lver.orig.lver.orig.i.us.lver.orig %indvars.iv.next.lver.orig.lver.orig.i.us = add nuw nsw i64 store float %sub42.lver.orig.lver.orig.i.us.lver.orig, float* ... %indvars.iv.next.lver.orig.lver.orig.i8.us, 1 ... %arrayidx26.lver.orig.lver.orig.i.us.lver.orig, align 4, !tbaa !12, %exitcond.not.lver.orig.lver.orig.i.us = icmp eq i64 ... %indvars.iv.next.lver.orig.lver.orig.i.us, %wide.trip.count.i ...!llvm.access.group!17 %indvars.iv.next.lver.orig.lver.orig.i.us.lver.orig = add nuw nsw i64 br i1 %exitcond.not.lver.orig.lver.orig.i.us, label %if.end.i.us.loopexit30, ... %indvars.iv.next.lver.orig.lver.orig.i8.us.lver.orig, 1 .. label %for.body.lver.orig.lver.orig.i.us, !llvm.loop !19 %exitcond.not.lver.orig.lver.orig.i.us.lver.orig = icmp eq i64 ... %indvars.iv.next.lver.orig.lver.orig.i.us.lver.orig, %wide.trip.count.i br i1 %exitcond.not.lver.orig.lver.orig.i.us.lver.orig, label ... %if.end.i.us.loopexit, label %for.body.lver.orig.lver.orig.i.us.lver.orig, ...!llvm.loop!19 if.end.i.us.loopexit: if.end.i.us.loopexit30: if.end.i.us.loopexit31: br label %if.end.i.us br label %if.end.i.us br label %if.end.i.us if.end.i.us: %52 = add nuw nsw i64 % local id x.0.us, 1%exitcond.not = icmp eq $i\overline{6}4$ %5 $\overline{2}$, $\overline{2}56$ br i1 %exitcond.not, label %adi kernel1.exit.loopexit, label ... %pregion for entry.entry.i.us, !llvm.loop!21 adi kernel1.exit.loopexit: br label %adi kernel1.exit adi kernel1.exit: ret void