```
%8:
%9 = shl i64 %5, 8
%sub.i = add i32 %3, -2
%cmp258.i = icmp slt i32 %3, 3
%10 = 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
%11 = add nsw i64 %wide.trip.count.i, -1
%12 = trunc i64 %11 to i32
%13 = icmp ugt i64 %11, 4294967295
%mul6.i = tail call { i64, i1 } @llvm.umul.with.overflow.i64(i64 %11, i64 4)
... #2
%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
%14 = getelementptr inbounds float, float* %2, i64 -7
%15 = getelementptr inbounds float, float* %0, i64 -7
%16 = getelementptr inbounds float, float* %1, i64 -7
%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 ], [ %85, %if.end.i ] %17 = add nuw nsw i64 % local_id_x.0, %9
             %conv.i = trunc i64 %17 to i32
             %cmp.i = icmp sge i32 %conv.i, %3
             %brmerge = or i1 %cmp.i, %cmp258.i
             br i1 %brmerge, label %if.end.i, label %for.body.lr.ph.i
                            Τ
                                                                F
                            for.body.lr.ph.i:
                            %mul.i = mul nsw i32 %conv.i, %3
                             %sub21.i = add i32 %mul.i, %3
                             %sub22.i = add i32 %sub21.i, -3
                             %18 = \text{sext i} 32 \% \text{mul.i to i} 64
                             %sub9.i = add i32 %mul.i, -1
                            br i1 %min.iters.check.i, label %for.body.i.preheader, label
                             .. %vector.scevcheck.i
                                            T
                                                                                   F
                                               vector.scevcheck.i:
                                                %19 = add i32 %sub.i, %mul.i
                                                %20 = sub i32 %19, %12
                                                %21 = icmp sgt i32 %20, %19
                                                %22 = sub i32 %sub22.i, %12
                                                %23 = icmp sgt i32 %22, %sub22.i
                                                %24 = \text{ or i } 1 \% 13, \%23
                                                %25 = \text{ or i } 1 \%24, \%21
                                                %26 = \text{sext i} 32 \% 19 \text{ to i} 64
                                                %scevgep.i = getelementptr float, float* %2, i64 %26
                                                %scevgep5.i = ptrtoint float* %scevgep.i to i64
                                                %27 = icmp ugt i64 %mul.result7.i, %scevgep5.i
                                                %28 = or i1 %mul.overflow8.i, %27
                                                %29 = \text{ or i } 1 \%25, \%28
                                                %30 = add nsw i64 %18, %10
                                                %scevgep9.i = getelementptr float, float* %2, i64 %30
                                                %scevgep910.i = ptrtoint float* %scevgep9.i to i64
                                                %31 = icmp ugt i64 %mul.result7.i, %scevgep910.i
                                                %32 = \text{ or } i1 \%31, \%29
                                                %33 = \text{sext i} 32 \% \text{sub} 22.i \text{ to i} 64
                                                %scevgep14.i = getelementptr float, float* %2, i64 %33
                                                %scevgep1415.i = ptrtoint float* %scevgep14.i to i64
                                                %34 = icmp ugt i64 %mul.result7.i, %scevgep1415.i
                                                %35 = \text{ or i } 1 \%34, \%32
                                                %scevgep19.i = getelementptr float, float* %0, i64 %33
                                                %scevgep1920.i = ptrtoint float* %scevgep19.i to i64
                                                %36 = icmp ugt i64 %mul.result7.i, %scevgep1920.i
                                                %37 = or i1 %mul.overflow8.i, %36
                                                %38 = \text{ or i } 1 \%37, \%35
                                                %scevgep24.i = getelementptr float, float* %1, i64 %33
                                                %scevgep2425.i = ptrtoint float* %scevgep24.i to i64
                                                %39 = icmp ugt i64 %mul.result7.i, %scevgep2425.i
                                                %40 = \text{ or i } 1 \%39, \%38
                                                br i1 %40, label %for.body.i.preheader, label %vector.memcheck.i
                                                                      vector.memcheck.i:
                                                                       %41 = add nsw i64 %26, 1
                                                                       %42 = sub nsw i64 %41, %wide.trip.count.i
                                                                       %scevgep29.i = getelementptr float, float* %2, i64 %42
                                                                       %scevgep31.i = getelementptr float, float* %2, i64 %41
                                                                       %43 = add nsw i64 %30, 1
                                                                       %44 = sub nsw i64 %43, %wide.trip.count.i
                                                                       %scevgep33.i = getelementptr float, float* %2, i64 %44
                                                                       %scevgep35.i = getelementptr float, float* %2, i64 %43
                                                                       %45 = add nsw i64 %33, 1
                                                                      %46 = sub nsw i64 %45, %wide.trip.count.i
                                                                       %scevgep37.i = getelementptr float, float* %2, i64 %46
                                                                       %scevgep39.i = getelementptr float, float* %2, i64 %45
                                                                       %scevgep41.i = getelementptr float, float* %0, i64 %46
                                                                       %scevgep43.i = getelementptr float, float* %0, i64 %45
                                                                       %scevgep45.i = getelementptr float, float* %1, i64 %46
                                                                       %scevgep47.i = getelementptr float, float* %1, i64 %45
                                                                       %bound0.i = icmp ult float* %scevgep29.i, %scevgep35.i
                                                                       %bound1.i = icmp ult float* %scevgep33.i, %scevgep31.i
                                                                       %found.conflict.i = and i1 %bound0.i, %bound1.i
                                                                       %bound049.i = icmp ult float* %scevgep29.i, %scevgep39.i
                                                                       %bound150.i = icmp ult float* %scevgep37.i, %scevgep31.i
                                                                       %found.conflict51.i = and i1 %bound150.i, %bound049.i
                                                                       %conflict.rdx.i = or i1 %found.conflict.i, %found.conflict51.i
                                                                       %bound052.i = icmp ult float* %scevgep29.i, %scevgep43.i
                                                                       %bound153.i = icmp ult float* %scevgep41.i, %scevgep31.i
                                                                       %found.conflict54.i = and i1 %bound153.i, %bound052.i
                                                                       %conflict.rdx55.i = or i1 %found.conflict54.i, %conflict.rdx.i
                                                                       %bound056.i = icmp ult float* %scevgep29.i, %scevgep47.i
                                                                       %bound157.i = icmp ult float* %scevgep45.i, %scevgep31.i
                                                                       %found.conflict58.i = and i1 %bound157.i, %bound056.i
                                                                       %conflict.rdx59.i = or i1 %found.conflict58.i, %conflict.rdx55.i
                                                                       br i1 %conflict.rdx59.i, label %for.body.i.preheader, label
                                                                      ... %vector.bodv.i.preheader
                                                                                                                                 F
                                                                                                                   vector.body.i.preheader:
                                                                                                                   br label %vector.bodv.i
                                                                                vector.body.i:
                                                                                %index.next.i1 = phi i64 [ %index.next.i, %vector.body.i ], [ 0,
                                                                                ... %vector.body.i.preheader ]
                                                                                %47 = sub nsw i64 %10, %index.next.i1
                                                                                %48 = add nsw i64 %47, %18
                                                                                %49 = getelementptr inbounds float, float* %14, i64 %48
                                                                                \%50 = bitcast float* \%49 to <8 x float>*
                                                                                %wide.load.i = load <8 x float>, <8 x float>* \%50, align 4, !tbaa !12,
                                                                                ... !alias.scope !16
                                                                                %reverse.i = shufflevector <8 x float> %wide.load.i, <8 x float> undef, <8 x
                                                                                ... i32> <i32 7, i32 6, i32 5, i32 4, i32 3, i32 2, i32 1, i32 0>
                                                                                %51 = \text{trunc } i64 \%47 \text{ to } i32
                                                                                %52 = add i32 %sub9.i, %51
                                                                                %53 = \text{sext i} 32 \% 52 \text{ to i} 64
                                                                                %54 = getelementptr inbounds float, float* %14, i64 %53
                                                                                \%55 = bitcast float* \%54 to <8 x float>*
                                                                                %wide.load60.i = load <8 x float>, <8 x float>* %55, align 4, !tbaa !12,
                                                                                ... !alias.scope !19
                                                                                %reverse61.i = shufflevector <8 x float> %wide.load60.i, <8 x float> undef,
                                                                                ... <8 x i32> <i32 7, i32 6, i32 5, i32 4, i32 3, i32 2, i32 1, i32 0>
                                                                                %56 = trunc i64 %index.next.i1 to i32
                                                                                %57 = sub i32 %3, %56
                                                                                %58 = add i32 %57, %mul.i
                                                                                %59 = add i32 \%58, -3
                                                                                \%60 = \text{sext i} 32 \% 59 \text{ to i} 64
                                                                                %61 = getelementptr inbounds float, float* %15, i64 %60
                                                                                \%62 = bitcast float* \%61 to <8 x float>*
                                                                                %wide.load65.i = load <8 x float>, <8 x float>* %62, align 4, !tbaa !12,
                                                                                ... !alias.scope !21
                                                                                %reverse66.i = shufflevector <8 x float> %wide.load65.i, <8 x float> undef,
                                                                                ... <8 x i32> <i32 7, i32 6, i32 5, i32 4, i32 3, i32 2, i32 1, i32 0>
                                                                                \%63 = \text{fneg} < 8 \text{ x float} > \%\text{reverse} 61.i
                                                                                \%64 = tail call < 8 x float > @llvm.fmuladd.v8f32(< 8 x float > \%63, < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail call < 8 x float > \%64 = tail 
                                                                                ... %reverse66.i, <8 x float> %reverse.i) #2
                                                                                %65 = sub i32 %sub22.i, %56
                                                                                \%66 = \text{sext i} 32 \%65 \text{ to i} 64
                                                                                %67 = getelementptr inbounds float, float* %16, i64 %66
                                                                                \%68 = bitcast float* \%67 to <8 x float>*
                                                                                %wide.load67.i = load <8 x float>, <8 x float>* %68, align 4, !tbaa !12,
                                                                                ... !alias.scope !23
                                                                                %reverse68.i = shufflevector <8 x float> %wide.load67.i, <8 x float> undef,
                                                                                ... <8 x i32> <i32 7, i32 6, i32 5, i32 4, i32 3, i32 2, i32 1, i32 0>
                                                                                %69 = fdiv <8 x float> %64, %reverse68.i, !fpmath !25
                                                                                \%70 = add i32 \%58, -2
                                                                                \%71 = \text{sext i} 32 \% 70 \text{ to i} 64
                                                                                %reverse69.i = shufflevector <8 x float> %69, <8 x float> undef, <8 x i32>
                                                                                ... <i32 7, i32 6, i32 5, i32 4, i32 3, i32 2, i32 1, i32 0>
                                                                                %72 = getelementptr inbounds float, float* %14, i64 %71
                                                                                \%73 = bitcast float* \%72 to <8 x float>*
                                                                                store <8 x float> %reverse69.i, <8 x float>* %73, align 4, !tbaa !12,
                                                                                ... !alias.scope !26, !noalias !28, !llvm.access.group !29
                                                                                %index.next.i = add i64 %index.next.i1, 8
                                                                                %74 = icmp eq i64 %index.next.i, %n.vec.i
                                                                                br i1 %74, label %middle.block.i, label %vector.body.i, !llvm.loop !31
                                                                               middle.block.i:
                                                                               br i1 %cmp.n.i, label %if.end.i, label %for.body.i.preheader
                       for.body.i.preheader:
                       %indvars.iv.next.i3.ph = phi i64 [ 0, %for.body.lr.ph.i ], [ 0,
                       ... %vector.scevcheck.i ], [0, %vector.memcheck.i], [%n.vec.i, %middle.block.i
                       br label %for.body.i
                       for.body.i:
                        %indvars.iv.next.i3 = phi i64 [ %indvars.iv.next.i, %for.body.i ], [
                        ... %indvars.iv.next.i3.ph, %for.body.i.preheader ] %75 = sub nsw i64 %10, %indvars.iv.next.i3
                        %76 = add nsw i64 %75, %18
                        %arrayidx.i = getelementptr inbounds float, float* %2, i64 %76
                        %77 = load float, float* %arrayidx.i, align 4, !tbaa !12
                        %78 = trunc i64 %75 to i32
                        %add10.i = add i32 %sub9.i, %78
                        %idxprom11.i = sext i32 %add10.i to i64
                        %arrayidx12.i = getelementptr inbounds float, float* %2, i64 %idxprom11.i
                        %79 = load float, float* %arrayidx12.i, align 4, !tbaa !12
                        %80 = trunc i64 %indvars.iv.next.i3 to i32
                        %81 = sub i32 %3, %80
                        %sub15.i = add i32 %81, %mul.i
                        %add16.i = add i32 %sub15.i, -3
                        %idxprom17.i = sext i32 %add16.i to i64
                        %arrayidx18.i = getelementptr inbounds float, float* %0, i64 %idxprom17.i
                        %82 = load float, float* %arrayidx18.i, align 4, !tbaa !12
                        %neg.i = fneg float %79
                        %83 = tail call float @llvm.fmuladd.f32(float %neg.i, float %82, float %77)
                        ... #2
                        %add23.i = sub i32 %sub22.i, %80
                        %idxprom24.i = sext i32 %add23.i to i64
                        %arrayidx25.i = getelementptr inbounds float, float* %1, i64 %idxprom24.i
                        %84 = load float, float* %arrayidx25.i, align 4, !tbaa !12
                        %div.i = fdiv float %83, %84, !fpmath !25
                        %add29.i = add i32 %sub15.i, -2
                        %idxprom30.i = sext i32 %add29.i to i64
                        %arrayidx31.i = getelementptr inbounds float, float* %2, i64 %idxprom30.i
                        store float %div.i, float* %arrayidx31.i, align 4, !tbaa !12,
                        ...!llvm.access.group!29
                        %indvars.iv.next.i = add nuw nsw i64 %indvars.iv.next.i3, 1
                        %exitcond.i = icmp eq i64 %indvars.iv.next.i, %wide.trip.count.i
                        br i1 %exitcond.i, label %if.end.i.loopexit, label %for.body.i, !llvm.loop
                        ... !33
                                              Τ
                                                                                               F
                                                        if.end.i.loopexit:
                                                         br label %if.end.i
                                                              if.end.i:
                                                               \%85 = \text{add nuw nsw } i64 \% \text{ local id } x.0, 1
                                                              \%exitcond = icmp eq i64 \%85. 256
                                                              br i1 %exitcond, label %adi kernel3.exit, label %pregion for entry.entry.i,
                                                              ... !llvm.loop !34
                                                                                                                                   F
                                                                         adi kernel3.exit:
```