```
%mul.i.i = shl i64 %7, 5
                                                                                                                                                                                  %mul3.i.i = shl i64 %8. 3
                                                                                                                                                                                  %11 = tail call float @llvm.sqrt.f32(float %3) #5
                                                                                                                                                                                  %12 = trunc i64 %8 to i32
                                                                                                                                                                                  %13 = mul i32 %12, %4
                                                                                                                                                                                  %14 = shl i32 %13, 3
                                                                                                                                                                                  %15 = trunc i64 %7 to i32
                                                                                                                                                                                  %16 = shl i32 %15, 5
                                                                                                                                                                                  %17 = add i32 %14, %16
                                                                                                                                                                                  %18 = trunc i64 %7 to i32
                                                                                                                                                                                  %19 = \text{shl i} 32 \%18, 5
                                                                                                                                                                                  %20 = \text{sext i} 32 \% 19 \text{ to i} 64
                                                                                                                                                                                  %scevgep = getelementptr float, float* %0, i64 %20
                                                                                                                                                                                  %21 = add nsw i64 %20, 32
                                                                                                                                                                                  %scevgep4 = getelementptr float, float* %0, i64 %21
                                                                                                                                                                                  %22 = trunc i64 %8 to i32
                                                                                                                                                                                  %23 = mul i32 %22, %4
                                                                                                                                                                                  %24 = shl i32 %23, 3
                                                                                                                                                                                  %25 = add i32 %24, %19
                                                                                                                                                                                  %scevgep8 = getelementptr float, float* %2, i64 32
                                                                                                                                                                                  %scevgep11 = getelementptr float, float* %1, i64 %20
                                                                                                                                                                                  %scevgep13 = getelementptr float, float* %1, i64 %21
                                                                                                                                                                                  br label %pregion for entry.pregion for init.i
                                                                                                                                                                            pregion for entry pregion for init.i:
                                                                                                                                                                            % local id y.0 = phi i64 [0, \sqrt{8}10], [%59, %pregion for end.i]
                                                                                                                                                                            \%\overline{2}6 = \overline{\text{trunc}} \text{ i}64 \% \text{ local id y.0 to i}32
                                                                                                                                                                            \%27 = \text{mul i} 32 \%26. \%4
                                                                                                                                                                            %28 = add i32 %27, %25
                                                                                                                                                                            %29 = \text{sext i} 32 \% 28 \text{ to i} 64
                                                                                                                                                                            %scevgep6 = getelementptr float, float* %2, i64 %29
                                                                                                                                                                            %scevgep9 = getelementptr float, float* %scevgep8, i64 %29
                                                                                                                                                                            %add6.i.i = add nuw nsw i64 % local id y.0, %mul3.i.i
                                                                                                                                                                             %conv2.i = trunc i64 %add6.i.i to i32
                                                                                                                                                                            %cmp.i = icmp slt i32 %conv2.i, %5
                                                                                                                                                                            %mul.i = mul nsw i32 %conv2.i, %4
                                                                                                                                                                            br i1 %cmp.i, label %vector.scevcheck, label %pregion for end.i
                                                                                     vector.scevcheck:
                                                                                     %30 = trunc i64 % local id v.0 to i32
                                                                                      %31 = \text{mul i} 32 \% 30, \% 4
                                                                                      %32 = add i32 %31, %17
                                                                                      %33 = icmp sgt i32 %32, 2147483616
                                                                                      br i1 %33, label %pregion for entry.entry.i.us.preheader, label
                                                                                     ... %vector.memcheck
                                                                                                                                             F
                                                                                                  vector.memcheck:
                                                                                                   %bound0 = icmp ult float* %scevgep, %scevgep9
%bound1 = icmp ult float* %scevgep6, %scevgep4
                                                                                                    %found.conflict = and i1 %bound0, %bound1
                                                                                                   %bound015 = icmp ult float* %scevgep11, %scevgep9
%bound116 = icmp ult float* %scevgep6, %scevgep13
%found.conflict17 = and i1 %bound015, %bound116
                                                                                                    %conflict.rdx = or i1 %found.conflict, %found.conflict17
                                                                                                   br i1 %conflict.rdx, label %pregion for entry.entry.i.us.preheader, label
                                                                                                   ... %vector.ph
                                                                                                                       Τ
                                                                                                                       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.splatinsert18 = insertelement <8 x i32> undef, i32 %4, i32 0 %broadcast.splat19 = shufflevector <8 x i32> %broadcast.splatinsert18, <8 x
                                                                   pregion for entry.entry.i.us.preheader:
                                                                    br label %pregion for entry.entry.i.us
                                                                                                                        ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                       %broadcast.splatinsert22 = insertelement <8 x float> undef, float %11, i32 0 %broadcast.splat23 = shufflevector <8 x float> %broadcast.splatinsert22, <8
                                                                                                                       ... x float> 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 ] %34 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
                                                                                                                      %35 = trunc < 8 \times i64 > %34 to < 8 \times i32 >
                                                                                                                      %36 = icmp sgt <8 x i32> %broadcast.splat19, %35
                                                                                                                      %37 = \text{extractelement} < 8 \times i64 > \%34, i32 0
                                                                                                                      %38 = shl i64 %37, 32
                                                                                                                      %39 = ashr exact i64 %38, 32
                                                                                                                      %40 = getelementptr inbounds float, float* %0, i64 %39
                                                                                                                      %41 = bitcast float* %40 to <8 x float>*
                                                                                                                      %wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                      ... float>* %41, i32 4, <8 x i1> %36, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                      ... !16, !noalias !19
                                                                                                                      %42 = \text{extractelement} < 8 \times i32 > \%35, i32 0
                                                                                                                      %43 = add nsw i32 %mul.i, %42
                                                                                                                      %44 = \text{sext i} 32 \% 43 \text{ to i} 64
                                                                                                                      %45 = getelementptr inbounds float, float* %2, i64 %44 %46 = bitcast float* %45 to <8 x float>*
                                        pregion for entry.entry.i.us:
                                        %_local_id_x.0.us = phi i64 [ %58, %if.end.r_exit.i.us ], [ 0, ... %pregion_for_entry.entry.i.us.preheader ] %add1.i.i.us = add nuw nsw i64 %_local_id_x.0.us, %mul.i.i
                                                                                                                      %wide.masked.load20 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                      ... float>* %46, i32 4, <8 x i1> %36, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                      ... !19
                                        %conv.i.us = trunc i64 %add1.i.i.us to i3\overline{2}
                                        %cmp4.i.us = icmp slt i32 %conv.i.us, %4
                                                                                                                       %47 = fsub <8 x float> %wide.masked.load20, %wide.masked.load
                                        br il %cmp4.i.us, label %if.then.i.us, label %if.end.r_exit.i.us
                                                                                                                      %48 = bitcast float* %45 to <8 x float>*
                                                                                                                      call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %47, <8 x float>*
... %48, i32 4, <8 x i1> %36), !tbaa !12, !alias.scope !19, !llvm.access.group !21
%49 = getelementptr inbounds float, float* %1, i64 %39
                                                                                                                      %50 = bitcast float* %49 to <8 x float>*
                                                                                                                      %wide.masked.load21 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                       ... float>* %50, i32 4, <8 x i1> %36, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                      ... !24, !noalias !19
                                                                                                                      %51 = fmul <8 x float> %broadcast.splat23, %wide.masked.load21
                                                                                                                       %52 = fdiv <8 x float> %47, %51, !fpmath !26
                                                                                                                      %53 = bitcast float* %45 to <8 x float>*
                                                                                                                      call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %52, <8 x float>* ... %53, i32 4, <8 x i1> %36), !tbaa !12, !alias.scope !19, !llvm.access.group !21
                                                                                                                       %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>
                                                                                                                      %54 = icmp eq i64 %index.next, 32
                                                                                                                      br i1 %54, label %pregion_for_end.i.loopexit25, label %vector.body,
                                                                                                                       ...!llvm.loop!27
                                                                                                                                            Τ
                                                                                                                                                                                              F
if.then.i.us:
%sext.i.us = shl i64 %add1.i.i.us, 32
%idxprom.i.us = ashr exact i64 %sext.i.us, 32
%arrayidx.i.us = getelementptr inbounds float, float* %0, i64 %idxprom.i.us %55 = load float, float* %arrayidx.i.us, align 4, !tbaa !12 %add.i.us = add nsw i32 %mul.i, %conv.i.us
%idxprom6.i.us = sext i32 %add.i.us to i64
%arrayidx7.i.us = getelementptr inbounds float, float* %2, i64 %idxprom6.i.us %56 = load float, float* %arrayidx7.i.us, align 4, !tbaa !12
%sub.i.us = fsub float %56. %55
store float %sub.i.us, float* %arrayidx7.i.us, align 4, !tbaa !12,
 .. !llvm.access.group !21
%arrayidx10.i.us = getelementptr inbounds float, float* %1, i64 %idxprom.i.us %57 = load float, float* %arrayidx10.i.us, align 4, !tbaa !12
%mul11.i.us = fmul float %11, %57
%div.i.us = fdiv float %sub.i.us, %mul11.i.us, !fpmath !26
store float %div.i.us, float* %arrayidx7.i.us, align 4, !tbaa !12,
...!llvm.access.group!21
br label %if.end.r exit.i.us
                                                         if.end.r exit.i.us:
                                                          \%58 = add nuw nsw i64 % local id x.0.us, 1
                                                          %exitcond.not = icmp eq i64 %58, 32
br i1 %exitcond.not, label %pregion for end.i.loopexit, label
                                                                                                                                                              pregion for end.i.loopexit25:
                                                                                                                                                              br label %pregion for end.i
                                                          ... %pregion for entry.entry.i.us, !llvm.loop !30
                                                                                                        pregion for end.i.loopexit:
                                                                                                         br label %pregion for end.i
                                                                                                                                                                       pregion for end.i:
                                                                                                                                                                       \%59 = add nuw nsw i64 % local id y.0, 1
                                                                                                                                                                       \%exitcond2.not = icmp eq \overline{164} \%59, \overline{8}
                                                                                                                                                                       br i1 %exitcond2.not, label %reduce kernel.exit, label
                                                                                                                                                                       ... %pregion for entry.pregion for init.i, !llvm.loop !31
                                                                                                                                                                           reduce kernel.exit:
                                                                                                                                                                           ret void
```

%10: