```
%12 = \text{sext i} 32 \% 5 \text{ to i} 64
                                                                                                                                                                    %13 = icmp slt i64 %12, 256
                                                                                                                                                                   %14 = select i1 %13, i64 %12, i64 256
                                                                                                                                                                    %mul.i.i = shl i64 %8, 8
                                                                                                                                                                   %cmp259.i = icmp sgt i32 %6, 0, !llvm.access.group !12
                                                                                                                                                                    %wide.trip.count.i = zext i32 %6 to i64
                                                                                                                                                                    %15 = icmp ugt i64 \%14, 1
                                                                                                                                                                   %umax = select i1 %15, i64 %14, i64 1
                                                                                                                                                                   %min.iters.check = icmp ult i64 %umax, 8
                                                                                                                                                                    br i1 %min.iters.check, label %pregion for entry.entry.i.preheader, label
                                                                                                                                                                   ... %vector.ph
                                                                                                                                                                                                              Τ
                                                                                                                                                                                                                                                                                                                F
                                                                                                                                                                                                                                            vector.ph:
                                                                                                                                                                                                                                            %n.vec = and i64 %umax, -8
                                                                                                                                                                                                                                            %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.splatinsert9 = insertelement <8 x i64> undef, i64 %12, i32 0
                                                                                                                                                                                                                                            %broadcast.splat10 = shufflevector <8 x i64> %broadcast.splatinsert9, <8 x
                                                                                                                                                                                                                                             ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                            %broadcast.splatinsert12 = insertelement <8 x i64> undef, i64
                                                                                                                                                                                                                                             ... %wide.trip.count.i, i32 0
                                                                                                                                                                                                                                            %broadcast.splat13 = shufflevector <8 x i64> %broadcast.splatinsert12, <8 x
                                                                                                                                                                                                                                             ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                            %broadcast.splatinsert18 = insertelement <8 x float> undef, float %3, i32 0
                                                                                                                                                                                                                                            %broadcast.splat19 = shufflevector <8 x float> %broadcast.splatinsert18, <8
                                                                                                                                                                                                                                             ... x float> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                            %broadcast.splatinsert20 = insertelement <8 x float> undef, float %4, i32 0
                                                                                                                                                                                                                                            %broadcast.splat21 = shufflevector <8 x float> %broadcast.splatinsert20, <8
                                                                                                                                                                                                                                             ... x float> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                            br label %vector.body
                                                                                                                                                                                                                                                                                vector.body:
                                                                                                                                                                                                                                                                               %index = phi i64 [ 0, %vector.ph ], [ %index.next, %for.end.i16 ]
                                                                                                                                                                                                                                                                                % vec.ind = phi < 8 \times i64 > [ < i64 0, i64 1, i64 2, i64 3, i64 4, i64 5, i64 6, i64 6]
                                                                                                                                                                                                                                                                                 .. i64 7>, %vector.ph ], [ %vec.ind.next, %for.end.i16 ]
                                                                                                                                                                                                                                                                                %16 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
                                                                                                                                                                                                                                                                                %17 = shl < 8 \times i64 > %16, < i64 32, i64
                                                                                                                                                                                                                                                                                 .. i64 32, i64 32>, !llvm.access.group !12
                                                                                                                                                                                                                                                                                \%18 = ashr exact < 8 \times i64 > \%17, < i64 32, i64 32
                                                                                                                                                                                                                                                                                ... 32, i64 32, i64 32>, !llvm.access.group !12
                                                                                                                                                                                                                                                                               %19 = getelementptr inbounds float, float* %1, <8 x i64> %18,
                                                                                                                                                                                                                                                                                ...!llvm.access.group!12
                                                                                                                                                                                                                                                                                call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> zeroinitializer, <8
                                                                                                                                                                                                                                                                                ... x float*> %19, i32 4, <8 x i1> <i1 true, i1 
                                                                                                                                                                                                                                                                                ... i1 true, i1 true, i1 true>), !tbaa !14, !llvm.access.group !12
                                                                                                                                                                                                                                                                                br i1 %cmp259.i, label %for.body.lr.ph.i6, label %for.end.i16
                                                                                                                                                                                                                                                                                                                                                                                                                                      F
                                                                                                                                                                                                                          for.body.lr.ph.i6:
                                                                                                                                                                                                                          %20 = getelementptr inbounds float, float* %0, <8 x i64> %18,
                                                                                                                                                                                                                            ..!llvm.access.group!12
                                                                                                                                                                                                                          br label %for.body.i7
                                                                                                                                                                                        for.body.i7:
                                                                                                                                                                                         %vec.phi = phi <8 x i64> [ %26, %for.body.i7 ], [ zeroinitializer,
                                                                                                                                                                                          .. %for.body.lr.ph.i6 ]
                                                                                                                                                                                         %vec.phi8 = phi <8 x float> [ %25, %for.body.i7 ], [ zeroinitializer,
                                                                                                                                                                                         ... %for.body.lr.ph.i6 ]
                                                                                                                                                                                        %21 = mul nsw <8 x i64> %vec.phi, %broadcast.splat10, !llvm.access.group !12 %22 = add nsw <8 x i64> %21, %18, !llvm.access.group !12
                                                                                                                                                                                         %23 = getelementptr inbounds float, float* %2, <8 x i64> %22,
                                                                                                                                                                                         ...!llvm.access.group!12
                                                                                                                                                                                         %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                                                                                        ... x float*> %23, 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), !tbaa !14, !llvm.access.group
                                                                                                                                                                                         %wide.masked.gather11 = call <8 x float>
                                                                                                                                                                                        ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %20, i32 4, <8 x i1> <i1 true, i1 true, i1
                                                                                                                                                                                         ... undef), !tbaa !14, !llvm.access.group !12
                                                                                                                                                                                         %24 = fsub <8 x float> %wide.masked.gather, %wide.masked.gather11,
                                                                                                                                                                                         ...!llvm.access.group!12
                                                                                                                                                                                        %25 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %24, <8 x float> %24, ... <8 x float> %vec.phi8), !llvm.access.group !12 call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %25, <8 x float*> ... %19, i32 4, <8 x i1> <i1 true, i1 true,
                                                                                                                                                                                         ... i1 true, i1 true>), !tbaa !14, !llvm.access.group !12
                                                                                                                                                                                         \%26 = \text{add nuw nsw} < 8 \times i64 > \%\text{vec.phi}, < i64 1, i
                                                                                                                                                                                         ... i64 1, i64 1, i64 1>, !llvm.access.group !12
                                                                                                                                                                                         \%27 = \text{icmp eq} < 8 \times \text{i}64 > \%26, %broadcast.splat13, !llvm.access.group !12
                                                                                                                                                                                         %28 = \text{extractelement} < 8 \times \text{i1} > \%27, \text{i32 } 0
                                                                                                                                                                                         br i1 %28, label %for.end.i16.loopexit, label %for.body.i7
                                                                                                                                                                                                                                                                                                                                                         F
                                                                                                                                                                                                                                       for.end.i16.loopexit:
                                                                                                                                                                                                                                       \%.lcssa25 = phi < 8 x float > [\%25, \%for.body.i7]
                                                                                                                                                                                                                                        br label %for.end.i16
                                                                                                                                                                                                                                                                for.end.i16:
                                                                                                                                                                                                                                                                  %vec.phi17 = phi <8 x float> [ zeroinitializer, %vector.body ], [ %.lcssa25,
                                                                                                                                                                                                                                                                  ... %for.end.i16.loopexit ]
                                                                                                                                                                                                                                                                 %29 = fdiv <8 x float> %vec.phi17, %broadcast.splat19, !fpmath !18,
                                                                                                                                                                                                                                                                  ...!llvm.access.group!12
                                                                                                                                                                                                                                                                 \%30 = \text{call} < 8 \text{ x float} > \text{@llvm.sqrt.v8f32}(< 8 \text{ x float} > \%29), !llvm.access.group
                                                                                                                                                                                                                                                                  ...!12
                                                                                                                                                                                                                                                                 %31 = fcmp ugt <8 x float> %30, %broadcast.splat21, !llvm.access.group !12
                                                                                                                                                                                                                                                                 \%32 = \text{select} < 8 \times i1 > \%31, < 8 \times float > \%30, < 8 \times float > < float 1.000000e + 00,
                                                                                                                                                                                                                                                                 ... float 1.000000e+00, float 1.000000e+00, float 1.000000e+00, float
                                                                                                                                                                                                                                                                \dots 1.000000e+00, float 1.000000e+00, float 1.000000e+00, float 1.000000e+00,
                                                                                                                                                                                                                                                                ...!llvm.access.group!12
                                                                                                                                                                                                                                                                 call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %32, <8 x float*>
                                                                                                                                                                                                                                                                 ... %19, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1 true,
                                                                                                                                                                                                                                                                 ... i1 true, i1 true>), !tbaa !14, !llvm.access.group !12 %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>
                                                                                                                                                                                                                                                                 %33 = icmp eq i64 %index.next, %n.vec
br i1 %33, label %middle.block, label %vector.body, !llvm.loop !19
                                                                                                                                                                                                                                                            middle.block:
                                                                                                                                                                                                                                                            %cmp.n = icmp eq i64 %umax, %n.vec
                                                                                                                                                                                                                                                            br i1 %cmp.n, label %std kernel.exit, label
                                                                                                                                                                                                                                                            ... %pregion for entry.entry.i.preheader
                                                                                        pregion for entry.entry.i.preheader:
                                                                                        % Ĭocal id x.0.ph = phi i64 [ 0, %11 ], [ %n.vec, %middle.block ]
                                                                                        br label % pregion for entry.entry.i
                                                                               pregion for entry.entry.i:
                                                                                % local id x.0 = phi i64 [ %42, %for.end.i ], [ %_local_id_x.0.ph,
                                                                               ... %pregion_for_entry.entry.i.preheader ]
%add1.i.i = add i64 %_local_id_x.0, %mul.i.i, !llvm.access.group !12
%sext.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
                                                                               %idxprom.i = ashr exact i64 %sext.i, 32, !llvm.access.group !12
                                                                               %arrayidx.i = getelementptr inbounds float, float* %1, i64 %idxprom.i,
                                                                               ...!llvm.access.group!12
                                                                               store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !14,
                                                                                ...!llvm.access.group!12
                                                                               br i1 %cmp259.i, label %for.body.lr.ph.i, label %for.end.i,
                                                                               ...!llvm.access.group!12
   for.body.lr.ph.i:
    %arrayidx7.i = getelementptr inbounds float, float* %0, i64 %idxprom.i,
     ...!llvm.access.group!12
    br label %for.body.i, !llvm.access.group !12
for.bodv.i:
%indvars.iv.next.i3 = phi i64 [ %indvars.iv.next.i, %for.body.i ], [ 0,
... %for.body.lr.ph.i ]
%34 = phi float [ %39, %for.body.i ], [ 0.000000e+00, %for.body.lr.ph.i ] %35 = mul nsw i64 %indvars.iv.next.i3, %12, !llvm.access.group !12
%36 = add nsw i64 %35, %idxprom.i, !llvm.access.group !12
%arrayidx5.i = getelementptr inbounds float, float* %2, i64 %36,
...!llvm.access.group!12
%37 = load float, float* %arrayidx5.i, align 4, !tbaa !14,
  ..!llvm.access.group!12
%38 = load float, float* %arrayidx7.i, align 4, !tbaa !14,
...!llvm.access.group!12
%sub.i = fsub float %37, %38, !llvm.access.group !12
%39 = tail call float @llvm.fmuladd.f32(float %sub.i, float %sub.i, float
... %34) #5, !llvm.access.group !12
store float %39, float* %arrayidx.i, align 4, !tbaa !14, !llvm.access.group
...!12
%indvars.iv.next.i = add nuw nsw i64 %indvars.iv.next.i3, 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 %for.end.i.loopexit, label %for.body.i,
... !llvm.loop !22, !llvm.access.group !12
                                                                                                                                             F
                                            for.end.i.loopexit:
                                             %.lcssa = phi float [ %39, %for.body.i ]
                                              br label %for.end.i
                                                               for.end.i:
                                                                %40 = \text{phi float } [0.000000e+00, \%pregion for entry.entry.i], [%.lcssa,]
                                                                ... %for.end.i.loopexit ]
                                                               %div.i = fdiv float %40, %3, !fpmath !18, !llvm.access.group !12 %41 = tail call float @llvm.sqrt.f32(float %div.i) #5, !llvm.access.group !12
                                                               %cmp27.i = fcmp ugt float %41, %4, !llvm.access.group !12 %storemerge.i = select i1 %cmp27.i, float %41, float 1.000000e+00,
                                                                ...!llvm.access.group!12
                                                               store float %storemerge.i, float* %arrayidx.i, align 4, !tbaa !14,
                                                               ...!llvm.access.group!12
%42 = add nuw i64 %_local_id_x.0, 1
                                                               %exitcond.not = icmp eq i64 %42, %umax
br i1 %exitcond.not, label %std_kernel.exit.loopexit, label
                                                                ... %pregion for entry.entry.i, !llvm.loop!24
                                                                                                                                                                                                                  F
                                                                                                                                          std kernel.exit.loopexit:
                                                                                                                                           br label %std kernel.exit
```

%11:

std kernel.exit: