```
\%9 = \text{sext i} 32 \%2 \text{ to i} 64
                                                                                           %10 = icmp slt i64 \%9, 32
                                                                                           %11 = select i1 %10, i64 %9, i64 32
                                                                                           %12 = \text{sext i} 32 \% 3 \text{ to i} 64
                                                                                           %13 = icmp slt i64 \%12, 8
                                                                                           %14 = select i1 %13, i64 %12, i64 8
                                                                                           %mul.i.i = shl i64 %5, 5
                                                                                           %mul3.i.i = shl i64 %6, 3
                                                                                           %15 = icmp ugt i64 %11, 1
                                                                                           %umax = select i1 %15, i64 %11, i64 1
                                                                                           %16 = icmp ugt i64 %14, 1
                                                                                           %umax1 = select i1 %16, i64 %14, i64 1
                                                                                           %min.iters.check = icmp ult i64 %umax1, 8
                                                                                           br i1 %min.iters.check, label
                                                                                           ... %pregion for entry.pregion for init.i.preheader, label %vector.ph
                                                                                                                            vector.ph:
                                                                                                                             %n.vec = and i64 %umax1, -8
                                                                                                                             %broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul3.i.i, i32 0
                                                                                                                             %broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
                                                                                                                             ... undef, <8 x i32> zeroinitializer
                                                                                                                            %broadcast.splatinsert3 = insertelement <8 x i32> undef, i32 %2, i32 0
                                                                                                                             %broadcast.splat4 = shufflevector <8 x i32> %broadcast.splatinsert3, <8 x
                                                                                                                             ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                            %broadcast.splatinsert6 = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
                                                                                                                             %broadcast.splat7 = shufflevector <8 x i64> %broadcast.splatinsert6, <8 x
                                                                                                                             ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                            %broadcast.splatinsert9 = insertelement <8 x i64> undef, i64 %umax, i32 0
                                                                                                                             %broadcast.splat10 = shufflevector <8 x i64> %broadcast.splatinsert9, <8 x
                                                                                                                             .. i64> undef, <8 x i32> zeroinitializer
                                                                                                                             br label %vector.body
                                                                                                                                  vector.body:
                                                                                                                                  %index = phi i64 [ 0, %vector.ph ], [ %index.next, %pregion for end.i11 ]
                                                                                                                                  %vec.ind = phi < 8 \times 164 > [< 164 0, 164 1, 164 2, 164 3, 164 4, 164 5, 164 6, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1, 164 1
                                                                                                                                  ... i64 7>, %vector.ph ], [ %vec.ind.next, %pregion for end.i11 ]
                                                                                                                                  %17 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
                                                                                                                                   %18 = \text{trunc} < 8 \times i64 > \%17 \text{ to } < 8 \times i32 >, !llvm.access.group !12
                                                                                                                                   %19 = mul nsw <8 x i32> %broadcast.splat4, %18, !llvm.access.group !12
                                                                                                                                   br label %pregion for entry.entry.i5
                                                                                                                          pregion for entry.entry.i5:
                                                                                                                           %\sqrt{\text{vec.phi}} = \text{phi} < 8 \times \text{i} = 64 \times \text{[zeroinitializer, %vector.body]}, [\%29]
                                                                                                                           ... %pregion for entry.entry.i5 ]
                                                                                                                           \%20 = \text{add} < 8 \times \text{i} 64 > \%\text{vec.phi}, \%\text{broadcast.splat7}, !llvm.access.group !12
                                                                                                                            %21 = trunc <8 x i64> %20 to <8 x i32>, !llvm.access.group !12
                                                                                                                            \%22 = \text{shl} < 8 \times i64 > \%20, < i64 32, 
                                                                                                                           ... i64 32, i64 32>, !llvm.access.group !12
                                                                                                                           %23 = ashr exact < 8 \times i64 > %22, < i64 32, i64 32, i64 32, i64 32, i64 32, i64 32, i64
                                                                                                                           ... 32, i64 32, i64 32>, !llvm.access.group !12
                                                                                                                           \%24 = getelementptr inbounds float, float* \%0, <8 x i64> \%23,
                                                                                                                           ...!llvm.access.group!12
                                                                                                                           %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                           ... x float*> %24, 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 !15, !llvm.access.group
                                                                                                                          ...!12
                                                                                                                           %25 = add nsw <8 x i32> %19, %21, !llvm.access.group !12
                                                                                                                           \%26 = \text{sext} < 8 \times i32 > \%25 \text{ to } < 8 \times i64 >, !llvm.access.group !12
                                                                                                                           %27 = getelementptr inbounds float, float* %1, <8 x i64> %26,
                                                                                                                           ...!llvm.access.group!12
                                                                                                                           %wide.masked.gather8 = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                          ... x float*> %27, i32 4, <8 x i1> <i1 true, i1 
                                                                                                                          ... i1 true, i1 true, i1 true>, <8 x float> undef), !tbaa !15, !llvm.access.group
                                                                                                                          ...!12
                                                                                                                           %28 = fsub <8 x float> %wide.masked.gather8, %wide.masked.gather,
                                                                                                                           ...!llvm.access.group!12
                                                                                                                           call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %28, <8 x float*>
                                                                                                                           ... %27, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1 true, i1 true,
                                                                                                                          ... i1 true, i1 true>), !tbaa !15, !llvm.access.group !12
                                                                                                                           %29 = add nuw <8 x i64> %vec.phi, <i64 1, i64 1, i64 1, i64 1, i64 1, i64 1,
                                                                                                                           ... i64 1, i64 1>
                                                                                                                           %30 = icmp eq < 8 \times i64 > %29, %broadcast.splat10
                                                                                                                            %31 = \text{extractelement} < 8 \times i1 > %30, i32 0
                                                                                                                           br i1 %31, label %pregion for end.i11, label %pregion for entry.entry.i5
                                                                                                                                                                 pregion for end.i11:
                                                                                                                                                                   %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>
                                                                                                                                                                  %32 = icmp eq i64 %index.next, %n.vec
                                                                                                                                                                  br i1 %32, label %middle.block, label %vector.body, !llvm.loop !19
                                                                                                                                                                                                                                                                                  F
                                                                                                                                                   middle.block:
                                                                                                                                                    %cmp.n = icmp eq i64 %umax1, %n.vec
                                                                                                                                                    br i1 %cmp.n, label %reduce kernel.exit, label
                                                                                                                                                    ... %pregion for entry pregion for init.i.preheader
                                                pregion for entry.pregion for init.i.preheader:
                                                 % local id y.0.ph = phi i64 [\overline{0}, %8], [%n.vec, %middle.block]
                                                 br label %pregion for entry.pregion for init.i
                                  pregion_for_entry.pregion for init.i:
                                  \% local id y.0 = phi i64 [%36, %pregion for end.i], [% local id y.0.ph,
                                  ... %pregion_for_entry.pregion_for_init.i.preheader ]
                                  %add6.i.i = add i64 % local id y.0, %mul3.i.i, !llvm.access.group !12
                                   %conv2.i = trunc i64 %add6.i.i to i32, !llvm.access.group !12
                                   %mul.i = mul nsw i32 %conv2.i, %2, !llvm.access.group !12
                                   br label %pregion for entry.entry.i
pregion for entry.entry.i:
\%_{local\_id\_x.0} = phi i64 [0, \%pregion_for_entry.pregion_for_init.i], [
... %35, %pregion_for_entry.entry.i ]
%add1.i.i = add 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
%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* %0, i64 %idxprom.i,
 ..!llvm.access.group!12
 %33 = load float, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
...!12
%add.i = add nsw i32 %mul.i, %conv.i, !llvm.access.group !12
 %idxprom6.i = sext i32 %add.i to i64, !llvm.access.group !12
 %arrayidx7.i = getelementptr inbounds float, float* %1, i64 %idxprom6.i,
 ..!llvm.access.group!12
%34 = load float, float* %arrayidx7.i, align 4, !tbaa !15,
...!llvm.access.group!12
%sub.i = fsub float %34, %33, !llvm.access.group !12 store float %sub.i, float* %arrayidx7.i, align 4, !tbaa !15,
 ...!llvm.access.group!12
%35 = add nuw i64 \%_local_id_x.0, 1
%exitcond.not = icmp eq i6\overline{4} %35, %umax
br i1 %exitcond.not, label %pregion_for_end.i, label
... %pregion for entry.entry.i, !llvm.loop 122
                                                                                                               F
                                                     pregion for end.i:
                                                       \%36 = add nuw i64 % local id y.0, 1
                                                       %exitcond2.not = icm\overline{p} eq i\overline{6}4 \overline{\%}36, %umax1
                                                      br i1 %exitcond2.not, label %reduce kernel.exit.loopexit, label
                                                      ... %pregion for entry pregion for init.i, !llvm.loop!25
                                                                                        reduce kernel.exit.loopexit:
                                                                                         br label %reduce kernel.exit
                                                                                                                      reduce kernel.exit:
```

%8: