```
%14 = sext i32 %4 to i64
                                                                                                                                                                                             %15 = icmp slt i64 %14, 32
                                                                                                                                                                                             %16 = select i1 %15, i64 %14, i64 32
                                                                                                                                                                                             %17 = \text{sext i} 32 \% 3 \text{ to i} 64
                                                                                                                                                                                             %18 = icmp slt i64 %17, 8
                                                                                                                                                                                             %19 = select i1 %18, i64 %17, i64 8
                                                                                                                                                                                             %mul.i.i = shl i64 %10, 5
                                                                                                                                                                                             %mul3.i.i = shl i64 %11, 3
                                                                                                                                                                                             %cmp639.i = icmp sgt i32 %5, 0, !llvm.access.group !12
                                                                                                                                                                                             %wide.trip.count.i = zext i32 %5 to i64
                                                                                                                                                                                             %20 = icmp ugt i64 %16, 1
                                                                                                                                                                                             %umax = select i1 %20, i64 %16, i64 1
                                                                                                                                                                                             %21 = icmp ugt i64 %19, 1
                                                                                                                                                                                             %umax3 = select i1 %21, i64 %19, i64 1
                                                                                                                                                                                             %min.iters.check = icmp ult i64 %umax3, 8
                                                                                                                                                                                             br i1 %min.iters.check, label
                                                                                                                                                                                            ... %pregion for entry.pregion for init.i.preheader, label %vector.ph
                                                                                                                                                                                                                                              vector.ph:
                                                                                                                                                                                                                                               %n.vec = and i64 %umax3, -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.splatinsert5 = insertelement <8 x i32> undef, i32 %4, i32 0
                                                                                                                                                                                                                                               %broadcast.splat6 = shufflevector <8 x i32> %broadcast.splatinsert5, <8 x
                                                                                                                                                                                                                                               ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                               %broadcast.splatinsert7 = insertelement <8 x i32> undef, i32 %5, i32 0
                                                                                                                                                                                                                                               %broadcast.splat8 = shufflevector <8 x i32> %broadcast.splatinsert7, <8 x
                                                                                                                                                                                                                                               ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                               %broadcast.splatinsert10 = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
                                                                                                                                                                                                                                               %broadcast.splat11 = shufflevector <8 x i64> %broadcast.splatinsert10, <8 x
                                                                                                                                                                                                                                              ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                               %broadcast.splatinsert18 = insertelement <8 x float> undef, float %7, i32 0
                                                                                                                                                                                                                                               %broadcast.splat19 = shufflevector <8 x float> %broadcast.splatinsert18, <8
                                                                                                                                                                                                                                               ... x float> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                               %broadcast.splatinsert20 = insertelement <8 x i64> undef, i64 %14, i32 0
                                                                                                                                                                                                                                               %broadcast.splat21 = shufflevector <8 x i64> %broadcast.splatinsert20, <8 x
                                                                                                                                                                                                                                               ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                               %broadcast.splatinsert23 = insertelement <8 x i64> undef, i64
                                                                                                                                                                                                                                              ... %wide.trip.count.i, i32 0
                                                                                                                                                                                                                                               %broadcast.splat24 = shufflevector <8 x i64> %broadcast.splatinsert23, <8 x
                                                                                                                                                                                                                                               ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                               %broadcast.splatinsert27 = insertelement <8 x i64> undef, i64 %umax, i32 0
                                                                                                                                                                                                                                               %broadcast.splat28 = shufflevector <8 x i64> %broadcast.splatinsert27, <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.i29 ]
                                                                                                                                                                                                                                                                           %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.i29 ]
                                                                                                                                                                                                                                                                           %22 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
                                                                                                                                                                                                                                                                          %23 = trunc < 8 \times i64 > %22 to < 8 \times i32 >, !llvm.access.group !12
                                                                                                                                                                                                                                                                           %24 = mul nsw <8 x i32> %broadcast.splat6, %23, !llvm.access.group !12
                                                                                                                                                                                                                                                                           %25 = mul nsw <8 x i32> %broadcast.splat8, %23
                                                                                                                                                                                                                                                                           %26 = \text{sext} < 8 \times i32 > \%25 \text{ to} < 8 \times i64 > 3 \times i
                                                                                                                                                                                                                                                                           br label %pregion for entry.entry.i9
                                                                                                                                                                                                                                                                        pregion for entry.entry.i9:
                                                                                                                                                                                                                                                                        %vec.phi = phi <8 x i64> [ zeroinitializer, %vector.body ], [ %44,
                                                                                                                                                                                                                                                                        ... %if.end.r exit.i26 ]
                                                                                                                                                                                                                                                                         %27 = add <8 x i64> %vec.phi, %broadcast.splat11, !llvm.access.group !12
                                                                                                                                                                                                                                                                         %28 = trunc <8 x i64> %27 to <8 x i32>, !llvm.access.group !12
                                                                                                                                                                                                                                                                         %29 = add nsw <8 x i32> %24, %28, !llvm.access.group !12
                                                                                                                                                                                                                                                                         %30 = \text{sext} < 8 \times i32 > \%29 \text{ to } < 8 \times i64 > \text{, !llvm.access.group !12}
                                                                                                                                                                                                                                                                         %31 = getelementptr inbounds float, float* %0, <8 x i64 > %30,
                                                                                                                                                                                                                                                                        ...!llvm.access.group!12
call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> zeroinitializer, <8
                                                                                                                                                                                                                                                                        ... x float*> %31, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true,
                                                                                                                                                                                                                                                                        ... i1 true, i1 true, i1 true>), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                                                                                                         br i1 %cmp639.i, label %for.body.lr.ph.i14, label %if.end.r exit.i26
                                                                                                                                                                                                          for.body.lr.ph.i14:
                                                                                                                                                                                                           %32 = shl <8 x i64> %27, <i64 32, i64 32, i64 32, i64 32, i64 32, i64 32, i64 32,
                                                                                                                                                                                                           ... i64 32, i64 32>, !llvm.access.group !12
                                                                                                                                                                                                           %33 = ashr exact < 8 x i64 > %32, < i64 32, i64 32, i64 32, i64 32, i64 32, i64 32, i64
                                                                                                                                                                                                           ... 32, i64 32, i64 32>, !llvm.access.group !12
                                                                                                                                                                                                           br label %for.body.i15
                                                                                                                                                                                               for.body.i15:
                                                                                                                                                                                               %vec.phi16 = phi <8 x i64> [ %41, %for.body.i15 ], [ zeroinitializer,
                                                                                                                                                                                               ... %for.body.lr.ph.i14 ]
                                                                                                                                                                                              %vec.phi17 = phi <8 x float> [ %40, %for.body.i15 ], [ zeroinitializer, ... %for.body.lr.ph.i14 ]
                                                                                                                                                                                               %34 = add nsw <8 x i64> %vec.phi16, %26, !llvm.access.group !12
                                                                                                                                                                                               %35 = getelementptr inbounds float, float* %1, <8 x i64> %34,
                                                                                                                                                                                               ...!llvm.access.group!12
                                                                                                                                                                                               %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                                                                                              ... x float*> %35, 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
                                                                                                                                                                                               %36 = fmul <8 x float> %wide.masked.gather, %broadcast.splat19,
                                                                                                                                                                                               ...!llvm.access.group!12
                                                                                                                                                                                               %37 = mul nsw <8 x i64> %vec.phi16, %broadcast.splat21, !llvm.access.group
                                                                                                                                                                                               ... !12
                                                                                                                                                                                               %38 = add nsw < 8 \times i64 > %37, %33, !llvm.access.group !12
                                                                                                                                                                                               %39 = \text{getelementptr inbounds float, float* } \%2, <8 \text{ x } i64 > \%38,
                                                                                                                                                                                               ...!llvm.access.group!12
                                                                                                                                                                                              %wide.masked.gather22 = call <8 x float> ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %39, i32 4, <8 x i1> <i1 true, i1 tr
                                                                                                                                                                                               ... undef), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                               %40 = call < 8 \times float > @llvm.fmuladd.v8f32(< 8 \times float > %36, < 8 
                                                                                                                                                                                               ... %wide.masked.gather22, <8 x float> %vec.phi17), !llvm.access.group !12
                                                                                                                                                                                               call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %40, <8 x float*>
                                                                                                                                                                                              ... %31, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1 true,
                                                                                                                                                                                               ... i1 true, i1 true>), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                               %41 = add nuw nsw < 8 x i64 > %vec.phi16, < i64 1, i64 1
                                                                                                                                                                                               ... i64 1, i64 1, i64 1>, !llvm.access.group !12
                                                                                                                                                                                               %42 = icmp eq <8 x i64> %41, %broadcast.splat24, !llvm.access.group !12
                                                                                                                                                                                               %43 = \text{extractelement} < 8 \times i1 > %42, i32 0
                                                                                                                                                                                               br i1 %43, label %if.end.r exit.i26.loopexit, label %for.body.i15
                                                                                                                                                                                                                                                                        if.end.r_exit.i26.loopexit:
                                                                                                                                                                                                                                                                         br label %if.end.r exit.i26
                                                                                                                                                                                                                                                                         if.end.r exit.i26:
                                                                                                                                                                                                                                                                         %44 = add nuw <8 x i64> %vec.phi, <i64 1, i64 1, i64 1, i64 1, i64 1, i64 1,
                                                                                                                                                                                                                                                                          ... i64 1, i64 1>
                                                                                                                                                                                                                                                                         %45 = icmp eq <8 x i64> %44, %broadcast.splat28
                                                                                                                                                                                                                                                                          %46 = \text{extractelement} < 8 \times i1 > %45, i32 0
                                                                                                                                                                                                                                                                         br i1 %46, label %pregion for end.i29, label %pregion for entry.entry.i9
                                                                                                                                                                                                                                               pregion for end.i29:
                                                                                                                                                                                                                                                 %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>
                                                                                                                                                                                                                                                %47 = icmp eq i64 %index.next, %n.vec
                                                                                                                                                                                                                                               br i1 %47, label %middle.block, label %vector.body, !llvm.loop !19
                                                                                                                                                                                                                               middle.block:
                                                                                                                                                                                                                                %cmp.n = icmp eq i64 %umax3, %n.vec
                                                                                                                                                                                                                                br i1 %cmp.n, label %mm2 kernel1.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 [ 0, %13 ], [ %n.vec, %middle.block ]
                                                                                                          br label %pregion for entry.pregion_for_init.i
                                                                                           pregion for entry.pregion for init.i:
                                                                                            ... %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, %4, !llvm.access.group !12
                                                                                            %mul8.i = mul nsw i32 %conv2.i, %5
                                                                                            %48 = sext i32 %mul8.i to i64
                                                                                            br label %pregion for entry.entry.i
                                                                pregion for entry.entry.i:
                                                                  ... %57, %if.end.r exit.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
                                                                 %add.i = add nsw i32 %mul.i, %conv.i, !llvm.access.group !12 %idxprom.i = sext i32 %add.i to i64, !llvm.access.group !12
                                                                  %arrayidx.i = getelementptr inbounds float, float* %0, i64 %idxprom.i,
                                                                 ...!llvm.access.group!12
store float 0.000000e+00, float* %arrayidx.i, align 4,!tbaa!15,
                                                                  ...!llvm.access.group!12
                                                                  br i1 %cmp639.i, label %for.body.lr.ph.i, label %if.end.r_exit.i,
                                                                  ...!llvm.access.group!12
                                                                                                                                                                                           F
                                for.body.lr.ph.i:
                                %sext.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
                                %49 = ashr exact i64 %sext.i, 32, !llvm.access.group !12
                                 br label %for.body.i, !llvm.access.group !12
for.body.i:
 %indvars.iv.next.i2 = phi i64 [ %indvars.iv.next.i, %for.body.i ], [ 0,
  .. %for.body.lr.ph.i ]
 %50 = phi float [ %56, %for.body.i ], [ 0.000000e+00, %for.body.lr.ph.i ]
 %51 = add nsw i64 %indvars.iv.next.i2, %48, !llvm.access.group !12
  %arrayidx11.i = getelementptr inbounds float, float* %1, i64 %51,
 ... !llvm.access.group !12
%52 = load float, float* %arrayidx11.i, align 4, !tbaa !15,
  ..!llvm.access.group!12
 %mul12.i = fmul float %52, %7, !llvm.access.group !12
%53 = mul nsw i64 %indvars.iv.next.i2, %14, !llvm.access.group !12
 %54 = add nsw i64 %53, %49, !llvm.access.group !12
 %arrayidx16.i = getelementptr inbounds float, float* %2, i64 %54,
  ..!llvm.access.group!12
 \%55 = \text{load float}, float* \%arrayidx16.i, align 4, !tbaa !15,
  ..!llvm.access.group!12
 %56 = tail call float @llvm.fmuladd.f32(float %mul12.i, float %55, float
  .. %50) #5, !llvm.access.group !12
 store float %56, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
 ... !12
  %indvars.iv.next.i = add nuw nsw i64 %indvars.iv.next.i2, 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 %if.end.r exit.i.loopexit, label %for.body.i,
 ...!llvm.loop!22,!llvm.access.group!12
                                                                  if.end.r exit.i.loopexit:
                                                                  br label %if.end.r exit.i
                                                                                                    if.end.r exit.i:
                                                                                                     \%57 = add nuw i64 \% local id x.0, 1
                                                                                                     %exitcond.not = icmp eq i6\overline{4} %57, %umax
                                                                                                     br i1 %exitcond.not, label %pregion for end.i, label
                                                                                                     ... %pregion_for_entry.entry.i, !llvm.loop 124
                                                                                                                  pregion for end.i:
                                                                                                                   \frac{1}{8} = add nuw i64 % local id y.0, 1
                                                                                                                   %exitcond4.not = icm\bar{p} eq i\bar{6}4 %58, %umax3
                                                                                                                   br i1 %exitcond4.not, label %mm2 kernel1.exit.loopexit, label
                                                                                                                   ... %pregion for entry.pregion for init.i, !llvm.loop !27
                                                                                                                                                         mm2 kernel1.exit.loopexit:
                                                                                                                                                           br label %mm2 kernel1.exit
                                                                                                                                                                                           mm2 kernel1.exit:
```

%13: