```
%11:
                                                                                                                                             %12 = \text{sext i} 32 \% 2 \text{ to i} 64
                                                                                                                                              %13 = icmp slt i64 %12, 32
                                                                                                                                             %14 = select i1 %13, i64 %12, i64 32
                                                                                                                                              %15 = \text{sext i} 32 \% 1 \text{ to i} 64
                                                                                                                                             %16 = icmp slt i64 %15, 8
                                                                                                                                             %17 = select i1 %16, i64 %15, i64 8
                                                                                                                                             %mul.i.i = shl i64 %8, 5
                                                                                                                                              %mul3.i.i = shl i64 %9, 3
                                                                                                                                             %mul6.i = mul i32 %6, %1, !llvm.access.group !12
                                                                                                                                             %cmp970.i = icmp sgt i32 %2, 0, !llvm.access.group !12
                                                                                                                                             %18 = zext i32 %2 to i64
                                                                                                                                             %19 = icmp ugt i64 %14, 1
                                                                                                                                             %umax = select i1 %19, i64 %14, i64 1
                                                                                                                                             %20 = icmp ugt i64 \%17, 1
                                                                                                                                             %umax3 = select i1 %20, i64 %17, 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 %mul6.i, 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 %2, 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 i64> undef, i64 %18, i32 0
                                                                                                                                                                                 %broadcast.splat19 = shufflevector <8 x i64> %broadcast.splatinsert18, <8 x
                                                                                                                                                                                ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                 %broadcast.splatinsert23 = insertelement <8 x i64> undef, i64 %umax, i32 0
                                                                                                                                                                                 %broadcast.splat24 = shufflevector <8 x i64> %broadcast.splatinsert23, <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.i25 ]
                                                                                                                                                                                                      %vec.ind = phi <8 x i64> [ <i64 0, i64 1, i64 2, i64 3, i64 4, i64 5, i64 6,
                                                                                                                                                                                                     ... i64 7>, %vector.ph ], [ %vec.ind.next, %pregion_for_end.i25 ] %21 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
                                                                                                                                                                                                     %22 = trunc < 8 \times i64 > %21 \text{ to } < 8 \times i32 >, !llvm.access.group !12
                                                                                                                                                                                                      %23 = add <8 x i32> %broadcast.splat6, %22, !llvm.access.group !12
                                                                                                                                                                                                      %24 = mul <8 x i32> %23, %broadcast.splat8, !llvm.access.group !12
                                                                                                                                                                                                      %25 = \text{sext} < 8 \times i32 > \%24 \text{ to} < 8 \times i64 >
                                                                                                                                                                                                     br label %pregion for entry.entry.i9
                                                                                                                                                                                                  pregion for entry.entry.i9:
                                                                                                                                                                                                   %vec.phi = phi <8 x i64> [ zeroinitializer, %vector.body ], [ %43,
                                                                                                                                                                                                   ... %if.end.r exit.i22 ]
                                                                                                                                                                                                   %26 = add <8 x i64 > %vec.phi, %broadcast.splat11, !llvm.access.group !12
                                                                                                                                                                                                   \%27 = trunc < 8 x i64 > \%26 to < 8 x i32 >, !llvm.access.group !12
                                                                                                                                                                                                   %28 = add nsw <8 x i32> %24, %27, !llvm.access.group !12
                                                                                                                                                                                                   %29 = \text{sext} < 8 \times \text{i}32 > \%28 \text{ to} < 8 \times \text{i}64 >, !llvm.access.group !12
                                                                                                                                                                                                   %30 = \text{getelementptr inbounds float, float* } \%5, <8 \times i64 > \%29,
                                                                                                                                                                                                   ...!llvm.access.group!12
                                                                                                                                                                                                   call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> zeroinitializer, <8
                                                                                                                                                                                                   ... x float*> %30, 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 %cmp970.i, label %for.body.preheader.i14, label %if.end.r exit.i22
                                                                                                                                                       for.body.preheader.i14:
                                                                                                                                                        %31 = shl <8 x i64> %26, <i64 32, i64 32, i64 32, i64 32, i64 32, i64 32, i64 32,
                                                                                                                                                        ... i64 32, i64 32>, !llvm.access.group !12
                                                                                                                                                        %32 = ashr exact <8 x i64> %31, <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> [ %40, %for.body.i15 ], [ zeroinitializer,
                                                                                                                                               ... %for.body.preheader.i14 ]
                                                                                                                                               %vec.phi17 = phi <8 x float> [ %39, %for.body.i15 ], [ zeroinitializer,
                                                                                                                                               ... %for.body.preheader.i14 ]
                                                                                                                                               %33 = add nsw <8 x i64> %vec.phi16, %25, !llvm.access.group !12 %34 = getelementptr inbounds float, float* %3, <8 x i64> %33,
                                                                                                                                               ...!llvm.access.group!12
                                                                                                                                               %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                                               ... x float*> %34, i32 4, <8 x i1> <i1 true, i1 
                                                                                                                                              ... i1 true, i1 true, i1 true>, <8 x float> undef), !tbaa !15, !llvm.access.group
                                                                                                                                               %35 = mul nuw nsw <8 x i64> %vec.phi16, %broadcast.splat19,
                                                                                                                                                ...!llvm.access.group!12
                                                                                                                                               %36 = add nsw <8 x i64> %35, %32, !llvm.access.group !12
                                                                                                                                               %37 = getelementptr inbounds float, float* %4, <8 x i64> %36,
                                                                                                                                               ...!llvm.access.group!12
                                                                                                                                              %wide.masked.gather20 = call <8 x float>
... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %37, i32 4, <8 x i1> <i1 true,
                                                                                                                                              ... il true, il true,
                                                                                                                                               ... undef), !tbaa !15, !llvm.access.group !12
                                                                                                                                               %38 = fmul <8 x float> %wide.masked.gather, %wide.masked.gather20,
                                                                                                                                              ...!llvm.access.group!12
%39 = fadd <8 x float> %vec.phi17, %38, !llvm.access.group!12
call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %39, <8 x float*>
                                                                                                                                               ... %30, 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
%40 = add nuw nsw <8 x i64> %vec.phi16, <i64 1, i64 1, i64 1, i64 1, i64 1,
                                                                                                                                               ... i64 1, i64 1, i64 1>, !llvm.access.group !12
                                                                                                                                               %41 = icmp eq <8 x i64> %40, %broadcast.splat19, !llvm.access.group !12
                                                                                                                                               %42 = \text{extractelement} < 8 \times i1 > %41, i32 0
                                                                                                                                                br i1 %42, label %if.end.r exit.i22.loopexit, label %for.body.i15
                                                                                                                                                                                                  if.end.r exit.i22.loopexit:
                                                                                                                                                                                                   br label %if.end.r exit.i22
                                                                                                                                                                                                   if.end.r exit.i22:
                                                                                                                                                                                                   %43 = add nuw <8 x i64> %vec.phi, <i64 1, i64 1, i64 1, i64 1, i64 1, i64 1,
                                                                                                                                                                                                   ... i64 1, i64 1>
                                                                                                                                                                                                   %44 = icmp eq <8 x i64> %43, %broadcast.splat24
                                                                                                                                                                                                   %45 = \text{extractelement} < 8 \times \text{i1} > \%44, \text{i32 0}
                                                                                                                                                                                                   br i1 %45, label %pregion for end.i25, label %pregion for entry.entry.i9
                                                                                                                                                                                 pregion for end.i25:
                                                                                                                                                                                  %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>
                                                                                                                                                                                 %46 = icmp eq i64 %index.next, %n.vec
                                                                                                                                                                                  br i1 %46, label %middle.block, label %vector.body, !llvm.loop !19
                                                                                                                                                                                                                                                                           F
                                                                                                                                                                      middle.block:
                                                                                                                                                                      %cmp.n = icmp eq i64 %umax3, %n.vec
br i1 %cmp.n, label %doitgen_kernel1.exit, label
                                                                                                                                                                      ... %pregion for entry.pregion for init.i.preheader
                                                                                  pregion_for_entry.pregion_for_init.i.preheader:
                                                                                   br label %pregion for entry.pregion_for_init.i
                                                                         pregion_for_entry.pregion_for_init.i:
                                                                        %_local_id_y.0 = phi i64 [ %58, %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
                                                                         %reass.add.i = add i32 %mul6.i, %conv2.i, !llvm.access.group !12
                                                                         %reass.mul.i = mul i32 %reass.add.i, %2, !llvm.access.group !12
                                                                         %47 = sext i32 % reass.mul.i to i64
                                                                         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], [
                                                       ... \( \bar{8} \) 57, \( \bar{8} \) if.end.r \( \text{exit.i} \) ]
                                                      %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
                                                      %add8.i = add nsw i32 %reass.mul.i, %conv.i, !llvm.access.group !12
                                                      %idxprom.i = sext i32 %add8.i to i64, !llvm.access.group !12
                                                      %arrayidx.i = getelementptr inbounds float, float* %5, 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 %cmp970.i, label %for.body.preheader.i, label %if.end.r_exit.i,
                                                      ...!llvm.access.group!12
                                                                                                                                           F
                            for.body.preheader.i:
                             %sext.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
                             %48 = 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.preheader.i ]
%49 = phi float [ %56, %for.body.i ], [ 0.000000e+00, %for.body.preheader.i ] %50 = add nsw i64 %indvars.iv.next.i2, %47, !llvm.access.group !12
%arrayidx24.i = getelementptr inbounds float, float* %3, i64 %50,
 ..!llvm.access.group!12
%51 = load float, float* %arrayidx24.i, align 4, !tbaa !15,
 ..!llvm.access.group!12
%52 = mul nuw nsw i64 %indvars.iv.next.i2, %18, !llvm.access.group !12
%53 = add nsw i64 %52, %48, !llvm.access.group !12
%arrayidx28.i = getelementptr inbounds float, float* %4, i64 %53,
 ..!llvm.access.group!12
%54 = load float, float* %arrayidx28.i, align 4, !tbaa !15,
...!llvm.access.group!12
%55 = fmul float %51, %54, !llvm.access.group!12
%56 = fadd float %49, %55, !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, %18, !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
                                                                                              F
                                                   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 i64 %57, %umax
                                                                               br i1 %exitcond.not, label %pregion_for_end.i, label
                                                                               ... %pregion for entry.entry.i, !llvm.loop \bar{1}24
                                                                                     pregion for end.i:
                                                                                     %58 = add nuw i64 %_local_id_y.0, 1
%exitcond4.not = icmp eq i64 %58, %umax3
br i1 %exitcond4.not, label %doitgen_kernel1.exit.loopexit, label
                                                                                      ... %pregion for entry.pregion for init.i, !llvm.loop !27
                                                                                                                 doitgen_kernel1.exit.loopexit:
br label %doitgen_kernel1.exit
                                                                                                                                        doitgen kernel1.exit:
```