```
%mul.i.i = shl i64 %4, 5
                                                                                                                                                                                                                             %add.i = add nsw i32 %1, 1, !llvm.access.group !12
                                                                                                                                                                                                                             %mul3.i.i = shl i64 %5, 3
                                                                                                                                                                                                                             %mul16.i = mul nsw i32 %2, %1
                                                                                                                                                                                                                             \%8 = \text{trunc } i64 \%5 \text{ to } i32
                                                                                                                                                                                                                             \%9 = \text{shl i} 32 \%8, 3
                                                                                                                                                                                                                             %10 = add i32 %9, %1
                                                                                                                                                                                                                              %11 = add i32 %10, 1
                                                                                                                                                                                                                             %12 = mul i32 %11, %2
                                                                                                                                                                                                                              %13 = add i32 %12, %1
                                                                                                                                                                                                                              %14 = trunc i64 %4 to i32
                                                                                                                                                                                                                              %15 = \text{shl i} 32 \%14, 5
                                                                                                                                                                                                                             %16 = add i32 %13, %15
                                                                                                                                                                                                                              %17 = add i32 %16, 1
                                                                                                                                                                                                                             %18 = \text{shl i} 32 \%1, 1
                                                                                                                                                                                                                              %19 = add i32 %15, %18
                                                                                                                                                                                                                              %20 = \text{ or } i32 \%19, 1
                                                                                                                                                                                                                             br label %pregion for entry.pregion for init.i
                                                                                                                                                                                                         pregion_for_entry.pregion_for_init.i:
%_local_id_y.0 = phi i64 [ 0, %7 ], [ %52, %pregion_for_end.i ]
%add6.i.i = add nuw nsw i64 %_local_id_y.0, %mul3.i.i, !llvm.access.group !12
                                                                                                                                                                                                          %21 = \text{trunc i} 64 \% \text{add} 6.i.i \text{ to i} 3\overline{2}, !llvm.access.group !12
                                                                                                                                                                                                          %conv7.i = add i32 %add.i, %21, !llvm.access.group !12
                                                                                                                                                                                                          %cmp.i = icmp slt i32 %conv7.i, %2, !llvm.access.group !12 %mul.i = mul nsw i32 %conv7.i, %2
                                                                                                                                                                                                          %add13.i = add nsw i32 %mul.i, %1
                                                                                                                                                                                                          %idxprom14.i = sext i32 %add13.i to i64
                                                                                                                                                                                                          %arrayidx15.i = getelementptr inbounds float, float* %0, i64 %idxprom14.i
                                                                                                                                                                                                          br i1 %cmp.i, label %vector.scevcheck, label %pregion for end.i
                                                                                                                               vector.scevcheck:
                                                                                                                               %22 = trunc i64 % local id y.0 to i32
                                                                                                                               %23 = \text{mul i} 32 \% 2\overline{2}, \%2
                                                                                                                               %24 = add i32 %23, %17
                                                                                                                                %25 = icmp sgt i32 %24, 2147483616
                                                                                                                               %ident.check = icmp ne i32 %2, 1
%26 = or i1 %25, %ident.check
                                                                                                                               %27 = icmp sgt i32 %20, 2147483616
%28 = or i1 %26, %27
                                                                                                                                br i1 %28, label %pregion for entry.entry.i.us.preheader, label %vector.ph
                                                                                                                                                       vector.ph:
                                                                                                                                                        %broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
                                                                                                                                                        %broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
                                                                                                                                                        ... undef, <8 x i32> zeroinitializer
                                                                                                                                                        %broadcast.splatinsert6 = insertelement <8 x i32> undef, i32 %add.i, i32 0
                                                                           pregion_for_entry.entry.i.us.preheader:
                                                                                                                                                        %broadcast.splat7 = shufflevector <8 x i32> %broadcast.splatinsert6, <8 x
                                                                                                                                                        ... i32> undef, <8 x i32> zeroinitializer
                                                                            br label %pregion for entry.entry.i.us
                                                                                                                                                        %broadcast.splatinsert8 = insertelement <8 x float*> undef, float*
                                                                                                                                                        ... %arrayidx15.i, i32 0
                                                                                                                                                        %broadcast.splat9 = shufflevector <8 x float*> %broadcast.splatinsert8, <8 x
                                                                                                                                                        ... float*> undef, <8 x i32> zeroinitializer
                                                                                                                                                        br label %vector.body
                                                                                                                                                      vector.body:
                                                                                                                                                      %index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ] %vec.ind = phi <8 x i64> [ <i64 0, i64 1, i64 2, i64 3, i64 4, i64 5, i64 6,
                                                                                                                                                      ... i64 7>, %vector.ph ], [ %vec.ind.next, %vector.body ]
                                                                                                                                                      %29 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group
                                                                                                                                                      %30 = trunc < 8 \times i64 > %29 to < 8 \times i32 >, !llvm.access.group !12
                                                                                                                                                      %31 = add <8 x i32> %broadcast.splat7, %30, !llvm.access.group !12
                                                                                                                                                      \%32 = \text{icmp slt} < 8 \times i32 > \%31, < i32 1, i32 1
                                                                                                                                                       .. 1, i32 1>, !llvm.access.group !12
                                                                                                                                                      %33 = \text{extractelement} < 8 \times i32 > %31, i32 0
                                                                                                                                                      %34 = add nsw i32 %33, %mul.i, !llvm.access.group !12
                                                                                                                                                      %35 = sext i32 %34 to i64, !llvm.access.group !12
                                                                                                                                                       %36 = getelementptr inbounds float, float* %0, i64 %35, !llvm.access.group
                                                                                                                                                       .. !12
                                                                                                                                                       %37 = bitcast float* %36 to <8 x float>*
                                           pregion for entry.entry.i.us:
                                                                                                                                                       %wide.load = load <8 x float>, <8 x float>* \%37, align 4, !tbaa !15,
                                            \%_{local\_id\_x.0.us} = phi i64 [\%51, \%if.end.r_exit.i.us], [0,
                                                                                                                                                       ..!llvm.access.group!12
                                           ... %pregion_for_entry.entry.i.us.preheader ]
%add1.i.i.us = add nuw nsw i64 %_local_id_x.0.us, %mul.i.i,
                                                                                                                                                      %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                                                      ... x float*> %broadcast.splat9, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1
                                           ...!llvm.access.group!12
%46 = trunc i64 %add1.i.i.us to i32,!llvm.access.group!12
                                                                                                                                                      ... true, i1 true, i1 true, i1 true, i1 true>, <8 x float> undef), !tbaa !15,
                                                                                                                                                     ...!llvm.access.group!12
%38 = add nsw i32 %33, %mul16.i,!llvm.access.group!12
                                            %conv2.i.us = add i32 %add.i, %46, !llvm.access.group !12
                                            %cmp9.i.us = icmp slt i32 %conv2.i.us, %2, !llvm.access.group !12
                                                                                                                                                       %39 = sext i32 %38 to i64, !llvm.access.group !12
                                            br i1 %cmp9.i.us, label %if.then.i.us, label %if.end.r_exit.i.us,
                                                                                                                                                      %40 = getelementptr inbounds float, float* %0, i64 %39, !llvm.access.group
                                            ...!llvm.access.group!12
                                                                                                                                                      ... !12
                                                                                                                                                      %41 = bitcast float* %40 to <8 x float>*
                                                                                                                                                       \%wide.load10 = load <8 x float>, <8 x float>* \%41, align 4, !tbaa !15,
                                                                                                                                                       .. !llvm.access.group !12
                                                                                                                                                      %42 = fneg <8 x float> %wide.masked.gather, !llvm.access.group !12
                                                                                                                                                      %43 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %42, <8 x float>
                                                                                                                                                      ... %wide.load10, <8 x float> %wide.load), !llvm.access.group !12
                                                                                                                                                      %44 = bitcast float* %36 to <8 x float>*
                                                                                                                                                      call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %43, <8 x float>*
                                                                                                                                                      ... %44, i32 4, <8 x i1> %32), !tbaa !15, !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>
                                                                                                                                                      %45 = icmp eq i64 %index.next, 32
br i1 %45, label %pregion_for_end.i.loopexit12, label %vector.body,
                                                                                                                                                      ... !llvm.loop !19
if.then.i.us:
%add11.i.us = add nsw i32 %conv2.i.us, %mul.i, !llvm.access.group !12
%idxprom.i.us = sext i32 %add11.i.us to i64, !llvm.access.group !12
%arrayidx.i.us = getelementptr inbounds float, float* %0, i64 %idxprom.i.us,
 ..!llvm.access.group!12
%47 = load float, float* %arrayidx.i.us, align 4, !tbaa !15,
...!llvm.access.group!12
%48 = load float, float* %arrayidx15.i, align 4, !tbaa !15,
...!llvm.access.group!12
%add17.i.us = add nsw i32 %conv2.i.us, %mul16.i, !llvm.access.group !12 %idxprom18.i.us = sext i32 %add17.i.us to i64, !llvm.access.group !12 %arrayidx19.i.us = getelementptr inbounds float, float* %0, i64
... %idxprom18.i.us, !llvm.access.group !12
%49 = load float, float* %arrayidx19.i.us, align 4, !tbaa !15,
...!llvm.access.group!12
%neg.i.us = fneg float %48, !llvm.access.group !12
%50 = tail call float @llvm.fmuladd.f32(float %neg.i.us, float %49, float
... %47) #5, !llvm.access.group !12
store float %50, float* %arrayidx.i.us, align 4, !tbaa !15,
...!llvm.access.group!12
br label %if.end.r exit.i.us, !llvm.access.group !12
                                                                     if.end.r exit.i.us:
                                                                     \%51 = add nuw nsw i64 % local id x.0.us, 1
                                                                     %exitcond.not = icmp eq i64 %51, 32
br i1 %exitcond.not, label %pregion_for_end.i.loopexit, label
                                                                                                                                                                                                   pregion_for_end.i.loopexit12:
                                                                                                                                                                                                    br label %pregion for end.i
                                                                     ... %pregion for entry.entry.i.us, !llvm.loop !22
                                                                                                                               pregion for end.i.loopexit:
                                                                                                                                br label %pregion for end.i
                                                                                                                                                                                                                  pregion for end.i:
                                                                                                                                                                                                                  \frac{1}{852} = add nuw nsw i64 %_local_id_y.0, 1
                                                                                                                                                                                                                  \%exitcond2.not = icmp eq \overline{164} \%52, \overline{8}
                                                                                                                                                                                                                  br i1 %exitcond2.not, label %lu kernel2.exit, label
                                                                                                                                                                                                                   .. %pregion for entry.pregion for init.i, !llvm.loop!23
                                                                                                                                                                                                                                                                             F
                                                                                                                                                                                                                        lu kernel2.exit:
                                                                                                                                                                                                                         ret void
```

%7: