```
%mul.i.i = shl i64 %5, 5
                                                                                                                                                                     %mul3.i.i = shl i64 %6, 3
                                                                                                                                                                     %9 = trunc i64 %6 to i32
                                                                                                                                                                     %10 = mul i32 %9, %2
                                                                                                                                                                     %11 = shl i32 %10, 3
                                                                                                                                                                     %12 = trunc i64 %5 to i32
                                                                                                                                                                    %13 = \text{shl i} 32 \%12, 5
                                                                                                                                                                     %14 = add i32 %11, %13
                                                                                                                                                                    %15 = trunc i64 %5 to i32
                                                                                                                                                                     %16 = shl i32 %15.5
                                                                                                                                                                    %17 = \text{sext i} 32 \% 16 \text{ to i} 64
                                                                                                                                                                     %scevgep = getelementptr float, float* %0, i64 %17
                                                                                                                                                                    %18 = add nsw i64 %17, 32
                                                                                                                                                                    %scevgep4 = getelementptr float, float* %0, i64 %18
                                                                                                                                                                     %19 = trunc i64 %6 to i32
                                                                                                                                                                     %20 = mul i32 %19, %2
                                                                                                                                                                    %21 = \text{shl i} 32 \%20, 3
                                                                                                                                                                    %22 = add i32 %21, %16
                                                                                                                                                                     %scevgep8 = getelementptr float, float* %1, i64 32
                                                                                                                                                                    br label %pregion for entry.pregion for init.i
                                                                                                                                                              pregion for entry.pregion for init.i:
                                                                                                                                                               \%_local_id_y.0 = phi i64 [0, \%8], [ %50, %pregion_for_end.i ]
                                                                                                                                                               \%\overline{2}3 = \overline{\text{trunc}} \text{ i64 \% local id y.0 to i32}
                                                                                                                                                               %24 = \text{mul i} 32 \% 2\overline{3}, \%2
                                                                                                                                                               %25 = add i32 %24, %22
                                                                                                                                                               %26 = \text{sext i} 32 \% 25 \text{ to i} 64
                                                                                                                                                               %scevgep6 = getelementptr float, float* %1, i64 %26
                                                                                                                                                               %scevgep9 = getelementptr float, float* %scevgep8, i64 %26
                                                                                                                                                               %add6.i.i = add nuw nsw i64 % local id y.0, %mul3.i.i
                                                                                                                                                               %conv2.i = trunc i64 %add6.i.i to i32
                                                                                                                                                               %cmp.i = icmp slt i32 %conv2.i, %3
                                                                                                                                                               %mul.i = mul nsw i32 %conv2.i, %2
                                                                                                                                                               br i1 %cmp.i, label %vector.scevcheck, label %pregion for end.i
                                                                               vector.scevcheck:
                                                                               %27 = trunc i64 % local id v.0 to i32
                                                                               %28 = \text{mul i} 32 \% 27, \%2
                                                                               %29 = add i32 %28, %14
                                                                               %30 = icmp sqt i32 %29, 2147483616
                                                                               br i1 %30, label %pregion_for_entry.entry.i.us.preheader, label
                                                                               ... %vector.memcheck
                                                                                           vector.memcheck:
                                                                                           %bound0 = icmp ult float* %scevgep, %scevgep9
                                                                                           %bound1 = icmp ult float* %scevgep6, %scevgep4
                                                                                           %found.conflict = and i1 %bound0, %bound1
                                                                                           br i1 %found.conflict, 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>
                                                              pregion for entry.entry.i.us.preheader:
                                                                                                               .. undef, < 8 \times i32 > zeroinitializer
                                                              br label %pregion for entry entry i.us
                                                                                                              %broadcast.splatinsert11 = insertelement <8 x i32> undef, i32 %2, i32 0
                                                                                                              %broadcast.splat12 = shufflevector <8 x i32> %broadcast.splatinsert11, <8 x
                                                                                                               .. i32> 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 ]
                                                                                                             %31 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
                                                                                                             %32 = trunc < 8 \times i64 > %31 to < 8 \times i32 >
                                                                                                             %33 = icmp sgt <8 x i32> %broadcast.splat12, %32
                                                                                                             %34 = \text{extractelement} < 8 \times i64 > \%31, i32 0
                                                                                                             %35 = shl i64 %34, 32
                                                                                                             %36 = ashr exact i64 %35, 32
                                                                                                             %37 = getelementptr inbounds float, float* %0, i64 %36 %38 = bitcast float* %37 to <8 x float>*
                                                                                                             %wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                             ... float>* %38, i32 4, <8 x i1> %33, <8 x float> undef), !tbaa !12, !alias.scope
                                     pregion for entry.entry.i.us:
                                     %_local_id_x.0.us = phi i64 [ %49, %if.end.r_exit.i.us ], [ 0, ... %pregion_for_entry.i.us.preheader ]
                                                                                                             ... !16, !noalias !19
                                                                                                             %39 = \text{extractelement} < 8 \times i32 > \%32, i32 0
                                     %add1.i.i.us = add nuw nsw i64 %_local_id_x.0.us, %mul.i.i
                                                                                                             %40 = add nsw i32 %mul.i, %39
                                     %conv.i.us = trunc i64 %add1.i.i.us to i32
                                                                                                             %41 = \text{sext i} 32 \% 40 \text{ to i} 64
                                                                                                             %42 = getelementptr inbounds float, float* %1, i64 %41
                                     %cmp4.i.us = icmp slt i32 %conv.i.us, %2
                                     br i1 %cmp4.i.us, label %if.then.i.us, label %if.end.r_exit.i.us
                                                                                                             %43 = bitcast float* %42 to <8 x float>*
                                                                                                             %wide.masked.load13 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                             ... float>* %43, i32 4, <8 x i1> %33, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                             ... !19
                                                                                                             %44 = fsub <8 x float> %wide.masked.load13, %wide.masked.load
                                                                                                             %45 = bitcast float* %42 to <8 x float>*
                                                                                                             call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %44, <8 x float>* ... %45, i32 4, <8 x i1> %33), !tbaa !12, !alias.scope !19, !llvm.access.group !21
                                                                                                             %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, 32
                                                                                                             br i1 %46, label %pregion_for_end.i.loopexit15, label %vector.body,
                                                                                                             ...!llvm.loop!24
                                                                                                                                                                               F
if.then.i.us:
%sext.i.us = shl i64 %add1.i.i.us, 32
%idxprom.i.us = ashr exact i64 %sext.i.us, 32
%arrayidx.i.us = getelementptr inbounds float, float* %0, i64 %idxprom.i.us
%47 = load float, float* %arrayidx.i.us, align 4, !tbaa !12
%add.i.us = add nsw i32 %mul.i, %conv.i.us
%idxprom6.i.us = sext i32 %add.i.us to i64
%arrayidx7.i.us = getelementptr inbounds float, float* %1, i64 %idxprom6.i.us %48 = load float, float* %arrayidx7.i.us, align 4, !tbaa !12 %sub.i.us = fsub float %48, %47
store float %sub.i.us, float* %arrayidx7.i.us, align 4, !tbaa !12,
...!llvm.access.group!21
br label %if.end.r exit.i.us
                                                     if.end.r exit.i.us:
                                                     %49 = add nuw nsw i64 % local id x.0.us, 1
                                                     %exitcond.not = icmp eq i\overline{6}4 %\overline{49}, \overline{3}2
                                                                                                                                                 pregion for end.i.loopexit15:
                                                     br i1 %exitcond.not, label %pregion_for_end.i.loopexit, label
                                                                                                                                                  br label %pregion for end.i
                                                     ... %pregion for entry.entry.i.us, !llvm.loop !27
                                                                                                pregion for end.i.loopexit:
                                                                                                br label %pregion for end.i
                                                                                                                                                          pregion for end.i:
                                                                                                                                                          ^{1}\%50 = add nuw nsw i64 % local id y.0, 1
                                                                                                                                                          %exitcond2.not = icmp eq \overline{i}64 %50, 8
                                                                                                                                                          br i1 %exitcond2.not, label %reduce kernel.exit, label
                                                                                                                                                          ... %pregion for entry.pregion for init.i, !llvm.loop !28
                                                                                                                                                             reduce kernel.exit:
                                                                                                                                                              ret void
```

%8: