```
%mul.i.i = shl i64 %7, 5
                                                                                                                                                                             mul3.i.i = shl i64 \%8, 3
                                                                                                                                                                             %11 = trunc i64 %8 to i32
                                                                                                                                                                             %12 = mul i32 %11, %5
                                                                                                                                                                             %13 = \text{shl i} 32 \%12.3
                                                                                                                                                                             %14 = trunc i64 %7 to i32
                                                                                                                                                                             %15 = \text{shl i} 32 \%14, 5
                                                                                                                                                                             %16 = add i32 %13, %15
                                                                                                                                                                             %17 = trunc i64 %8 to i32
                                                                                                                                                                             %18 = shl i32 \%17, 3
                                                                                                                                                                             %19 = \text{sext i} 32 \% 18 \text{ to i} 64
                                                                                                                                                                             %20 = mul i32 %17, %5
                                                                                                                                                                             %21 = shl i32 %20, 3
                                                                                                                                                                             %22 = trunc i64 %7 to i32
                                                                                                                                                                             %23 = \text{shl i} 32 \%22, 5
                                                                                                                                                                             %24 = add i32 %21, %23
                                                                                                                                                                             %scevgep6 = getelementptr float, float* %0, i64 32
                                                                                                                                                                             %25 = \text{sext i} 32 \% 23 \text{ to i} 64
                                                                                                                                                                             %scevgep9 = getelementptr float, float* %1, i64 %25
                                                                                                                                                                             %26 = add nsw i64 %25, 32
                                                                                                                                                                             %scevgep11 = getelementptr float, float* %1, i64 %26
                                                                                                                                                                             %scevgep16 = getelementptr float, float* %2, i64 %25
                                                                                                                                                                             %scevgep18 = getelementptr float, float* %2, i64 %26
                                                                                                                                                                             br label %pregion for entry.pregion for init.i
                                                                                                                                                                  pregion for entry.pregion for init.i:
                                                                                                                                                                   \%_local_id_y.0 = phi i64 [0, \%10], [%63, %pregion for end.i]
                                                                                                                                                                   \%\bar{2}7 = \bar{a}d\bar{d}i64 \% local id y.0, \%19
                                                                                                                                                                   %scevgep = getelementptr float, float* %3, i64 %27
                                                                                                                                                                   %scevgep3 = bitcast float* %scevgep to i8*
                                                                                                                                                                   %uglygep = getelementptr i8, i8* %scevgep3, i64 1
                                                                                                                                                                   %28 = trunc i64 % local id v.0 to i32
                                                                                                                                                                   %29 = \text{mul i} 32 \% 28, \%5
                                                                                                                                                                   %30 = add i32 %29, %24
                                                                                                                                                                   %31 = \text{sext i} 32 \% 30 \text{ to i} 64
                                                                                                                                                                   %scevgep4 = getelementptr float, float* %0, i64 %31
                                                                                                                                                                   %scevgep45 = bitcast float* %scevgep4 to i8*
                                                                                                                                                                   %scevgep7 = getelementptr float, float* %scevgep6, i64 %31
                                                                                                                                                                   %scevgep13 = getelementptr float, float* %4, i64 %27
                                                                                                                                                                   %scevgep1314 = bitcast float* %scevgep13 to i8*
                                                                                                                                                                   %uglygep15 = getelementptr i8, i8* %scevgep1314, i64 1
                                                                                                                                                                   %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, %5
                                                                                                                                                                   %sext.i = shl i64 %add6.i.i, 32
                                                                                                                                                                   %idxprom.i = ashr exact i64 %sext.i, 32
                                                                                                                                                                   %arrayidx.i = getelementptr inbounds float, float* %3, i64 %idxprom.i
                                                                                                                                                                   %arrayidx9.i = getelementptr inbounds float, float* %4, i64 %idxprom.i
                                                                                                                                                                   %mul.i = mul nsw i32 %conv2.i, %5
                                                                                                                                                                   br i1 %cmp.i, label %vector.scevcheck, label %pregion for end.i
                                                                                    vector.scevcheck:
                                                                                    %32 = trunc i64 % local id y.0 to i32
                                                                                    %33 = \text{mul i} 32 \% 3\overline{2}, \%5
                                                                                     %34 = add i32 %33, %16
                                                                                    %35 = icmp sgt i32 %34, 2147483616
br i1 %35, label %pregion_for_entry.entry.i.us.preheader, label
                                                                                     ... %vector.memcheck
                                                                                                 vector.memcheck:
                                                                                                %bound0 = icmp ult float* %arrayidx.i, %scevgep7
%bound1 = icmp ugt i8* %uglygep, %scevgep45
                                                                                                 %found.conflict = and i1 %bound0, %bound1
                                                                                                %bound020 = icmp ult float* %scevgep9, %scevgep7
%bound121 = icmp ult float* %scevgep4, %scevgep11
%found.conflict22 = and i1 %bound020, %bound121
                                                                                                 %conflict.rdx = or i1 %found.conflict, %found.conflict22
                                                                                                %bound024 = icmp ult float* %arrayidx9.i, %scevgep7
%bound125 = icmp ugt i8* %uglygep15, %scevgep45
%found.conflict26 = and i1 %bound024, %bound125
%conflict.rdx27 = or i1 %conflict.rdx, %found.conflict26
                                                                                                %bound028 = icmp ult float* %scevgep16, %scevgep7
%bound129 = icmp ult float* %scevgep4, %scevgep18
%found.conflict30 = and i1 %bound028, %bound129
%conflict.rdx31 = or i1 %conflict.rdx27, %found.conflict30
                                                                                                br i1 %conflict.rdx31, label %pregion for entry.entry.i.us.preheader, label
                                                                                                 ... %vector.ph
                                                                                                                    Τ
                                                                                                                                                                F
                                                                                                                     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.splatinsert32 = insertelement <8 x i32> undef, i32 %5, i32 0 %broadcast.splat33 = shufflevector <8 x i32> %broadcast.splatinsert32, <8 x
                                                                                                                      ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                     %broadcast.splatinsert34 = insertelement <8 x float*> undef, float*
                                                                  pregion for entry.entry.i.us.preheader:
                                                                  br label %pregion for entry.entry.i.us
                                                                                                                     ... %arrayidx.i, i32 0
                                                                                                                     %broadcast.splat35 = shufflevector <8 x float*> %broadcast.splatinsert34, <8
                                                                                                                      ... x float*> undef, <8 x i32> zeroinitializer
                                                                                                                     %broadcast.splatinsert36 = insertelement <8 x float*> undef, float*
... %arrayidx9.i, i32 0
                                                                                                                     %broadcast.splat37 = shufflevector <8 x float*> %broadcast.splatinsert36, <8
                                                                                                                      ... x float*> undef, < 8 \times 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 ]
                                                                                                                    %36 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
                                                                                                                    %37 = \text{trunc} < 8 \times i64 > %36 \text{ to} < 8 \times i32 > 
                                                                                                                    %38 = icmp sgt <8 x i32> %broadcast.splat33, %37
                                                                                                                    %wide.masked.gather = call <8 x float>@llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                   ... x float*> %broadcast.splat35, i32 4, <8 x i1> %38, <8 x float> undef), !tbaa
                                                                                                                   ... !12, !alias.scope !16, !noalias !19
                                                                                                                    %39 = \text{extractelement} < 8 \times i64 > %36, i32 0
                                                                                                                    %40 = \text{shl } i64 \%39, 32
                                                                                                                    %41 = ashr exact i64 %40, 32
                                                                                                                    %42 = getelementptr inbounds float, float* %1, i64 %41
                                                                                                                    %43 = bitcast float* %42 to <8 x float>*
                                                                                                                    %wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                   ... float>* %43, i32 4, <8 x i1> %38, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                    ... !21, !noalias !19
                                                                                                                    %wide.masked.gather38 = call <8 x float>
                                                                                                                   ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %broadcast.splat37, i32 4, <8 ... x i1> %38, <8 x float> undef), !tbaa !12, !alias.scope !23, !noalias !19 %44 = getelementptr inbounds float, float* %2, i64 %41
                                       pregion for entry.entry.i.us:
                                       \%_{local}id_x.0.us = phi i64 [\%62, \%if.end.r_exit.i.us], [0, ]
                                       ... %pregion_for_entry.entry.i.us.preheader ]
%add1.i.i.us = add nuw nsw i64 %_local_id_x.0.us, %mul.i.i
                                                                                                                    %45 = bitcast float* %44 to <8 x float>*
                                                                                                                    %wide.masked.load39 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                    ... float>* %45, i32 4, <8 x i1> %38, <8 x float> undef), !tbaa !12, !alias.scope
                                       %conv.i.us = trunc i64 %add1.i.i.us to i32
                                       %cmp4.i.us = icmp slt i32 %conv.i.us, %5
                                                                                                                   ... !25, !noalias !19
                                                                                                                   %46 = fmul <8 x float> %wide.masked.gather38, %wide.masked.load39
%47 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %wide.masked.gather,
                                       br i1 %cmp4.i.us, label %if.then.i.us, label %if.end.r_exit.i.us
                                                                                                                    \dots <8 x float> %wide.masked.load, <8 x float> %46)
                                                                                                                    %48 = \text{extractelement} < 8 \times i32 > \%37, i32 0
                                                                                                                    %49 = add nsw i32 %mul.i, %48
                                                                                                                    %50 = \text{sext i} 32 \% 49 \text{ to i} 64
                                                                                                                    %51 = getelementptr inbounds float, float* %0, i64 %50 %52 = bitcast float* %51 to <8 x float>*
                                                                                                                    %wide.masked.load40 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                   ... float>* %52, i32 4, <8 x i1> %38, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                    %53 = fadd < 8 x float > %wide.masked.load40, %47
                                                                                                                    %54 = bitcast float* %51 to <8 x float>*
                                                                                                                    call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %53, <8 x float>*
                                                                                                                    ... %54, i32 4, <8 x i1> %38), !tbaa !12, !alias.scope !19, !llvm.access.group !27
                                                                                                                    %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>
                                                                                                                    %55 = icmp eq i64 %index.next, 32
                                                                                                                    br i1 %55, label %pregion_for_end.i.loopexit42, label %vector.body,
                                                                                                                   ...!llvm.loop!30
if.then.i.us:
%56 = load float, float* %arrayidx.i, align 4, !tbaa !12
%sext26.i.us = shl i64 %add1.i.i.us, 32
%idxprom6.i.us = ashr exact i64 %sext26.i.us, 32
%arrayidx7.i.us = getelementptr inbounds float, float* %1, i64 %idxprom6.i.us
%57 = load float, float* %arrayidx7.i.us, align 4, !tbaa !12
%58 = load float, float* %arrayidx9.i, align 4, !tbaa !12
%arrayidx11.i.us = getelementptr inbounds float, float* %2, i64
.. %idxprom6.i.us
%59 = load float, float* %arrayidx11.i.us, align 4, !tbaa !12
%mul12.i.us = fmul float %58, %59
%60 = tail call float @llvm.fmuladd.f32(float %56, float %57, float
.. %mul12.i.us) #6
%add.i.us = add nsw i32 %mul.i, %conv.i.us
%idxprom13.i.us = sext i32 %add.i.us to i64
%arrayidx14.i.us = getelementptr inbounds float, float* %0, i64
 .. %idxprom13.i.us
%61 = load float, float* %arrayidx14.i.us, align 4, !tbaa !12 %add15.i.us = fadd float %61, %60
store float %add15.i.us, float* %arrayidx14.i.us, align 4, !tbaa !12,
...!llvm.access.group!27
br label %if.end.r exit.i.us
                                                        if.end.r exit.i.us:
                                                        %62 = add nuw nsw i64 %_local_id_x.0.us, 1
                                                        %exitcond.not = icmp eq i64 %62, 32
br i1 %exitcond.not, label %pregion_for_end.i.loopexit, label
                                                                                                                                                           pregion_for_end.i.loopexit42:
                                                                                                                                                           br label %pregion_for_end.i
                                                         .. %pregion for entry.entry.i.us, !llvm.loop!33
                                                                                                      pregion for end.i.loopexit:
                                                                                                       br label %pregion for end.i
                                                                                                                                                                  pregion for end.i:
                                                                                                                                                                   \%63 = add nuw nsw i64 % local id y.0, 1
                                                                                                                                                                   %exitcond2.not = icmp eq i64 %63, 8
br i1 %exitcond2.not, label %gemver_kernel1.exit, label
                                                                                                                                                                   ... %pregion for entry.pregion for init.i, !llvm.loop !34
                                                                                                                                                                     gemver_kernel1.exit:
                                                                                                                                                                      ret void
```

%10: