```
%11 = shl i64 \%7, 5
                                                                                                                                                                                          %12 = shl i64 \%8, 3
                                                                                                                                                                                          %13 = trunc i64 %8 to i32
                                                                                                                                                                                          %14 = mul i32 %13, %5
                                                                                                                                                                                          %15 = \text{shl i} 32 \%14, 3
                                                                                                                                                                                          %16 = trunc i64 %7 to i32
                                                                                                                                                                                          %17 = \text{shl i} 32 \%16, 5
                                                                                                                                                                                          %18 = add i32 %15, %17
                                                                                                                                                                                          %19 = trunc i64 %8 to i32
                                                                                                                                                                                          %20 = \text{shl i} 32 \%19, 3
                                                                                                                                                                                          %21 = sext i32 %20 to i64
                                                                                                                                                                                          %22 = mul i32 %19. %5
                                                                                                                                                                                          %23 = shl i32 %22, 3
                                                                                                                                                                                          %24 = trunc i64 %7 to i32
                                                                                                                                                                                          %25 = \text{shl i} 32 \%24, 5
                                                                                                                                                                                          %26 = add i32 %23, %25
                                                                                                                                                                                          %scevgep6 = getelementptr float, float* %0, i64 32
                                                                                                                                                                                          %27 = sext i32 %25 to i64
                                                                                                                                                                                          %scevgep9 = getelementptr float, float* %1, i64 %27
                                                                                                                                                                                          %28 = add nsw i64 %27, 32
                                                                                                                                                                                          %scevgep11 = getelementptr float, float* %1, i64 %28
                                                                                                                                                                                          %scevgep16 = getelementptr float, float* %2, i64 %27
                                                                                                                                                                                          %scevgep18 = getelementptr float, float* %2, i64 %28
                                                                                                                                                                                          br label %pregion for entry.pregion for init.i
                                                                                                                                                                               pregion for entry.pregion for init.i:
                                                                                                                                                                                \c^{\infty}_local_id_y.0 = phi i64 [0, \c^{\infty}10], [%67, %pregion_for_end.i]
                                                                                                                                                                               \%29 = add i64 \% local_id_y.0, \%21
                                                                                                                                                                                %scevgep = getelementptr float, float* %3, i64 %29
                                                                                                                                                                                %scevgep3 = bitcast float* %scevgep to i8*
                                                                                                                                                                                %uglygep = getelementptr i8, i8* %scevgep3, i64 1
                                                                                                                                                                                %30 = trunc i64 % local id v.0 to i32
                                                                                                                                                                                %31 = \text{mul i} 32 \% 3\overline{0}, \%5
                                                                                                                                                                                %32 = add i32 %26, %31
                                                                                                                                                                                %33 = \text{sext i} 32 \% 32 \text{ to i} 64
                                                                                                                                                                                %scevgep4 = getelementptr float, float* %0, i64 %33
                                                                                                                                                                                %scevgep45 = bitcast float* %scevgep4 to i8*
                                                                                                                                                                                %scevgep7 = getelementptr float, float* %scevgep6, i64 %33
                                                                                                                                                                                %scevgep13 = getelementptr float, float* %4, i64 %29
                                                                                                                                                                                %scevgep1314 = bitcast float* %scevgep13 to i8*
                                                                                                                                                                                %uglygep15 = getelementptr i8, i8* %scevgep1314, i64 1
                                                                                                                                                                                %34 = add nuw nsw i64 % local id y.0, %12
                                                                                                                                                                                %conv2.i = trunc i64 %34 to i32
                                                                                                                                                                                %cmp.i = icmp slt i32 %conv2.i, %5
                                                                                                                                                                                %sext.i = shl i64 %34, 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:
                                                                                           %35 = trunc i64 %_local_id_y.0 to i32
%36 = mul i32 %35, %5
                                                                                           %37 = add i32 %18, %36
                                                                                           %38 = icmp sgt i32 %37, 2147483616
br i1 %38, 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
                                                                                                                             Т
                                                                                                                                                                             F
                                                                                                                               %broadcast.splatinsert = insertelement <8 x i64> undef, i64 %11, 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 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]
%39 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
                                                                                                                             %40 = trunc < 8 \times i64 > %39 \text{ to } < 8 \times i32 >
                                                                                                                             %41 = icmp sgt <8 x i32> %broadcast.splat33, %40
                                                                                                                             %wide.masked.gather = call <8 x float>@llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                            ... x float*> %broadcast.splat35, i32 4, <8 x i1> %41, <8 x float> undef), !tbaa
                                                                                                                            ... !12, !alias.scope !16, !noalias !19
                                                                                                                             %42 = \text{extractelement} < 8 \times i64 > %39, i32 0
                                                                                                                             %43 = shl i64 %42, 32
                                                                                                                             %44 = ashr exact i64 %43, 32
                                                                                                                             %45 = getelementptr inbounds float, float* %1, i64 %44
                                                                                                                             %46 = bitcast float* %45 to <8 x float>*
                                                                                                                             %wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                            ... float>* %46, i32 4, <8 x i1> %41, <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> %41, <8 x float> undef), !tbaa !12, !alias.scope !23, !noalias !19 %47 = getelementptr inbounds float, float* %2, i64 %44 %48 = bitcast float* %47 to <8 x float>*
                                          pregion for entry.entry.i.us:
                                         %_local_id_x.0.us = phi i64 [ %66, %if.end.r_exit.i.us ], [ 0, ... %pregion_for_entry.entry.i.us.preheader ] %59 = add nuw nsw i64 %_local_id_x.0.us, %11 %conv.i.us = trunc i64 %59 to i32
                                                                                                                            %wide.masked.load39 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x ... float>* %48, i32 4, <8 x i1> %41, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                            ... !25, !noalias !19
                                          %cmp4.i.us = icmp slt i32 %conv.i.us, %5
                                                                                                                            %49 = fmul <8 x float> %wide.masked.gather38, %wide.masked.load39
%50 = 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> %49)
                                                                                                                             %51 = \text{extractelement} < 8 \text{ x i} 32 > \%40, i 32 0
                                                                                                                             %52 = add nsw i32 %mul.i, %51
                                                                                                                             %53 = \text{sext i} 32 \% 52 \text{ to i} 64
                                                                                                                            %54 = getelementptr inbounds float, float* %0, i64 %53 %55 = bitcast float* %54 to <8 x float>*
                                                                                                                             %wide.masked.load40 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                            ... float>* %55, i32 4, <8 x i1> %41, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                             %56 = fadd < 8 x float > %wide.masked.load40, %50
                                                                                                                             \%57 = bitcast float* \%54 to <8 x float>*
                                                                                                                            call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %56, <8 x float>* ... %57, i32 4, <8 x i1> %41), !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>
                                                                                                                            %58 = icmp eq i64 %index.next, 32
br i1 %58, label %pregion_for_end.i.loopexit42, label %vector.body,
                                                                                                                             ...!llvm.loop!30
if.then.i.us:
%60 = load float, float* %arrayidx.i, align 4, !tbaa !12
%sext26.i.us = shl i64 %59, 32
%idxprom6.i.us = ashr exact i64 %sext26.i.us, 32
%arrayidx7.i.us = getelementptr inbounds float, float* %1, i64 %idxprom6.i.us %61 = load float, float* %arrayidx7.i.us, align 4, !tbaa !12
%62 = load float, float* %arrayidx9.i, align 4, !tbaa !12
%arrayidx11.i.us = getelementptr inbounds float, float* %2, i64
... %idxprom6.i.us
%63 = load float, float* %arrayidx11.i.us, align 4, !tbaa !12 %mul12.i.us = fmul float %62, %63
%64 = tail call float @llvm.fmuladd.f32(float %60, float %61, 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
%65 = load float, float* %arrayidx14.i.us, align 4, !tbaa !12 %add15.i.us = fadd float %65, %64
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:
                                                               %66 = add nuw nsw i64 %_local_id_x.0.us, 1
%exitcond = icmp eq i64 %66, 32
br i1 %exitcond, 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:
                                                                                                                                                                                 %67 = add nuw nsw i64 % local_id_y.0, 1
%exitcond2 = icmp eq i64 %67, 8
br i1 %exitcond2, label %gemver_kernel1.exit, label
                                                                                                                                                                                 ... %pregion for entry.pregion for init.i, !llvm.loop !34
                                                                                                                                                                                   gemver_kernel1.exit:
                                                                                                                                                                                    ret void
```

%10: