```
%10:
                                    %11 = \text{sext i} 32 \% 5 \text{ to i} 64
                                    %12 = icmp slt i64 %11, 32
                                    %13 = select i1 %12, i64 %11, i64 32
                                    %14 = icmp slt i64 %11, 8
                                    %15 = select i1 %14, i64 %11, i64 8
                                    %mul.i.i = shl i64 %7, 5
                                    %mul3.i.i = shl i64 %8, 3
                                    %16 = icmp ugt i64 \%13, 1
                                    %umax = select i1 %16, i64 %13, i64 1
                                    %17 = icmp ugt i64 \%15, 1
                                    %umax1 = select i1 %17, i64 %15, i64 1
                                    %18 = add nsw i64 %umax, -1
                                    %19 = \text{trunc } i64 \%7 \text{ to } i32
                                    %20 = \text{shl i} 32 \%19.5
                                    %21 = trunc i64 %8 to i32
                                    %22 = mul i32 %21, %5
                                    %23 = \text{shl i} 32 \%22, 3
                                    %24 = \text{zext i} 32 \% 23 \text{ to i} 64
                                    %25 = \text{zext i} 32 \% 20 \text{ to i} 64
                                    %26 = add nuw nsw i64 %24, %25
                                    %27 = \text{zext i} 32 \% 5 \text{ to i} 64
                                    br label %pregion for entry.pregion for init.i
                  pregion for entry.pregion for init.i:
                  \%\bar{2}8 = \bar{m}u\bar{l} i64 \% local id y.0, \%27
                  \%29 = \text{add } i64 \% \overline{2}6, \% \overline{2}8
                   %30 = \text{trunc } i64 \%29 \text{ to } i32
                  %add6.i.i = add i64 %_local_id_y.0, %mul3.i.i, !llvm.access.group !12
                   %conv2.i = trunc i64 %add6.i.i to i32, !llvm.access.group !12
                   %sext.i = shl i64 %add6.i.i, 32, !llvm.access.group !12
                   %idxprom.i = ashr exact i64 %sext.i, 32, !llvm.access.group !12
                   %arrayidx.i = getelementptr inbounds float, float* %3, i64 %idxprom.i,
                   ..!llvm.access.group!12
                  %arrayidx9.i = getelementptr inbounds float, float* %4, i64 %idxprom.i,
                  ...!llvm.access.group!12
                  %mul.i = mul nsw i32 %conv2.i, %5, !llvm.access.group !12
                  %min.iters.check = icmp ult i64 %umax, 16
                   br i1 %min.iters.check, label %pregion for entry.entry.i.preheader, label
                  ... %vector.scevcheck
                                       Τ
                                                                                       F
                                           vector.scevcheck:
                                           %31 = \text{trunc } i64 \%18 \text{ to } i32
                                           %32 = add i32 %20, %31
                                           %33 = icmp slt i32 %32, %20
                                           %34 = icmp ugt i64 %18, 4294967295
                                           %35 = \text{ or i } 1 \%33, \%34
                                           %36 = \text{trunc } i64 \%18 \text{ to } i32
                                            %37 = add i32 %30, %36
                                           %38 = icmp slt i32 %37, %30
                                           %39 = \text{icmp ugt } i64 \%18, 4294967295
                                            %40 = \text{ or i } 1 \%38, \%39
                                           %41 = \text{ or i } 1 \%35, \%40
                                           br i1 %41, label %pregion for entry.entry.i.preheader, label %vector.ph
                                                                                           vector.ph:
                                                                                            %n.vec = and i64 %umax, -16
                                                                                            br label %vector.body
                                                         vector.body:
                                                          %index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
                                                          %42 = add i64 %index, %mul.i.i, !llvm.access.group !12
                                                          %43 = trunc i64 %42 to i32, !llvm.access.group !12
                                                          %44 = load float, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
                                                          ... !12
                                                          %broadcast.splatinsert11 = insertelement <8 x float> undef, float %44, i32 0
                                                          %broadcast.splat12 = shufflevector <8 x float> %broadcast.splatinsert11, <8
                                                          .. x float> undef, <8 x i32> zeroinitializer
                                                          %broadcast.splatinsert13 = insertelement <8 x float> undef, float %44, i32 0
                                                          %broadcast.splat14 = shufflevector <8 x float> %broadcast.splatinsert13, <8
                                                          ... x float> undef, <8 x i32> zeroinitializer
                                                          %45 = shl i64 %42, 32, !llvm.access.group !12
                                                          %46 = ashr exact i64 %45, 32, !llvm.access.group !12
                                                          %47 = getelementptr inbounds float, float* %1, i64 %46, !llvm.access.group
                                                          ... !12
                                                          %48 = bitcast float* %47 to <8 x float>*
                                                          %wide.load = load <8 x float>, <8 x float>* \%48, align 4, !tbaa !15,
                                                          ...!llvm.access.group!12
                                                          %49 = getelementptr inbounds float, float* %47, i64 8
                                                          %50 = bitcast float* %49 to <8 x float>*
                                                          %wide.load6 = load <8 x float>, <8 x float>* %50, align 4, !tbaa !15,
                                                          ...!llvm.access.group!12
                                                          %51 = load float, float* %arrayidx9.i, align 4, !tbaa !15,
                                                          ...!llvm.access.group!12
                                                          %broadcast.splatinsert = insertelement <8 x float> undef, float %51, i32 0
                                                          %broadcast.splat = shufflevector <8 x float> %broadcast.splatinsert, <8 x
                                                          .. float> undef, <8 x i32> zeroinitializer
                                                          %broadcast.splatinsert9 = insertelement <8 x float> undef, float %51, i32 0
                                                          %broadcast.splat10 = shufflevector <8 x float> %broadcast.splatinsert9, <8 x
                                                          .. float> undef, <8 x i32> zeroinitializer
                                                          %52 = getelementptr inbounds float, float* %2, i64 %46, !llvm.access.group
                                                          ...!12
                                                          %53 = bitcast float* %52 to <8 x float>*
                                                          %wide.load7 = load < 8 \times float >, < 8 \times float > %53, align 4, !tbaa !15,
                                                          ...!llvm.access.group!12
                                                          %54 = getelementptr inbounds float, float* %52, i64 8
                                                          %55 = bitcast float* %54 to <8 x float>*
                                                          %wide.load8 = load <8 x float>, <8 x float>* %55, align 4, !tbaa !15,
                                                          ...!llvm.access.group!12
                                                          %56 = fmul <8 x float> %broadcast.splat, %wide.load7, !llvm.access.group !12
                                                          %57 = fmul <8 x float> %broadcast.splat10, %wide.load8, !llvm.access.group
                                                          ...!12
                                                          %58 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %broadcast.splat12,
                                                          ... <8 x float> %wide.load, <8 x float> %56), !llvm.access.group !12
                                                          %59 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %broadcast.splat14,
                                                          ... <8 x float> %wide.load6, <8 x float> %57), !llvm.access.group !12
                                                          %60 = add nsw i32 %mul.i, %43, !llvm.access.group !12
                                                          %61 = sext i32 %60 to i64, !llvm.access.group !12
                                                          %62 = getelementptr inbounds float, float* %0, i64 %61, !llvm.access.group
                                                          ...!12
                                                          \%63 = bitcast float* \%62 to <8 x float>*
                                                          %wide.load15 = load <8 x float>, <8 x float>* %63, align 4, !tbaa !15,
                                                          ...!llvm.access.group!12
                                                          %64 = getelementptr inbounds float, float* %62, i64 8
                                                          \%65 = bitcast float* \%64 to < 8 x float>*
                                                          %wide.load16 = load <8 x float>, <8 x float>* %65, align 4, !tbaa !15,
                                                          ...!llvm.access.group!12
                                                          %66 = fadd <8 x float> %wide.load15, %58, !llvm.access.group !12
                                                          %67 = fadd <8 x float> %wide.load16, %59, !llvm.access.group !12
                                                          \%68 = bitcast float* \%62 to <8 x float>*
                                                          store <8 x float> %66, <8 x float>* %68, align 4, !tbaa !15,
                                                          ...!llvm.access.group!12
                                                          \%69 = bitcast float* \%64 to <8 x float>*
                                                          store <8 x float> %67, <8 x float>* %69, align 4, !tbaa !15,
                                                          ...!llvm.access.group!12
                                                          %index.next = add i64 %index, 16
                                                          %70 = icmp eq i64 %index.next, %n.vec
                                                          br i1 %70, label %middle.block, label %vector.body, !llvm.loop !19
                                                       middle.block:
                                                        %cmp.n = icmp eq i64 %umax, %n.vec
                                                       br i1 %cmp.n, label %pregion_for_end.i, label
                                                       ... %pregion for entry.entry.i.preheader
     pregion for entry.entry.i.preheader:
      \%_{local\_id\_x.0.ph} = phi i64 [0, %vector.scevcheck], [0, %vector.scevcheck]
      ... %pregion for entry.pregion for init.i ], [ %n.vec, %middle.block ]
      br label %pregion for entry.entry.i
pregion for entry.entry.i:
% local id x.0 = phi i64 [ %77, %pregion for entry.entry.i ], [
... \(\overline{\pi}\)_local_id_x.0.ph, \(\psi\)pregion_for_entry.entry.i.preheader ]
%add1.i.i = add i64 % local id x.0, %mul.i.i, !llvm.access.group !12
%conv.i = trunc i64 %add1.i.i to i32, !llvm.access.group !12
%71 = load float, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
...!12
%sext26.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
%idxprom6.i = ashr exact i64 %sext26.i, 32, !llvm.access.group !12
%arrayidx7.i = getelementptr inbounds float, float* %1, i64 %idxprom6.i,
...!llvm.access.group!12
%72 = load float, float* %arrayidx7.i, align 4, !tbaa !15,
...!llvm.access.group!12
%73 = load float, float* %arrayidx9.i, align 4, !tbaa !15,
...!llvm.access.group!12
%arrayidx11.i = getelementptr inbounds float, float* %2, i64 %idxprom6.i,
...!llvm.access.group!12
%74 = load float, float* %arrayidx11.i, align 4, !tbaa !15,
...!llvm.access.group!12
%mul12.i = fmul float %73, %74, !llvm.access.group !12
%75 = tail call float @llvm.fmuladd.f32(float %71, float %72, float
... %mul12.i) #3, !llvm.access.group !12
%add.i = add nsw i32 %mul.i, %conv.i, !llvm.access.group !12
%idxprom13.i = sext i32 %add.i to i64, !llvm.access.group !12
%arrayidx14.i = getelementptr inbounds float, float* %0, i64 %idxprom13.i,
...!llvm.access.group!12
%76 = load float, float* %arrayidx14.i, align 4, !tbaa !15,
...!llvm.access.group!12
%add15.i = fadd float %76, %75, !llvm.access.group !12
store float %add15.i, float* %arrayidx14.i, align 4, !tbaa !15,
...!llvm.access.group!12
\%77 = \text{add nuw } i64\% \text{ local id } x.0, 1
\%exitcond.not = icmp eq i6\overline{4} \%77, \%umax
br i1 %exitcond.not, label %pregion for end.i.loopexit, label
... %pregion for entry entry.i, !llvm.loop 122
                                                                         F
                                   pregion for end.i.loopexit:
                                    br label %pregion for end.i
                                                                       pregion for end.i:
                                                                        \%78 = add nuw i64 \% local id v.0, 1
                                                                        %exitcond2.not = icm\overline{p} eq i\overline{6}4 \overline{\%}78, %umax1
                                                                        br i1 %exitcond2.not, label %gemver kernel1.exit, label
                                                                       ... %pregion for entry.pregion for init.i, !llvm.loop !23
                                                                                       Т
                                                                                                                             F
```

gemver\_kernel1.exit: