```
%9:
                                         %10 = \text{sext i} 32 \% 4 \text{ to i} 64
                                          %11 = icmp slt i64 %10, 256
                                          %12 = select i1 %11, i64 %10, i64 256
                                          %mul.i.i = shl i64 %6, 8
                                          %mul.i = mul nsw i32 %4, %3, !llvm.access.group !12
                                          %sub.i = add nsw i32 %3, -1, !llvm.access.group !12
                                          %mul2.i = mul nsw i32 %sub.i, %4, !llvm.access.group !12
                                          %13 = icmp ugt i64 \%12, 1
                                          %umax = select i1 %13, i64 %12, i64 1
                                          %min.iters.check = icmp ult i64 %umax, 8
                                         br i1 %min.iters.check, label %pregion for entry.entry.i.preheader, label
                                         ... %vector.scevcheck
                                                                            Τ
                                                                                                                                                           F
                                                                                     vector.scevcheck:
                                                                                      %14 = add nsw i64 %umax, -1
                                                                                      %15 = mul i32 %4, %3
                                                                                      %16 = trunc i64 %6 to i32
                                                                                      %17 = \text{shl i} 32 \%16, 8
                                                                                      %18 = add nsw i32 %15, %17
                                                                                      %19 = trunc i64 %14 to i32
                                                                                      %20 = add i32 %18, %19
                                                                                      %21 = icmp slt i32 %20, %18
                                                                                      %22 = icmp ugt i64 \%14, 4294967295
                                                                                      %23 = \text{ or i } 1 \% 21, \% 22
                                                                                      %24 = add i32 %3, -1
                                                                                      %25 = \text{mul i} 32 \%24, \%4
                                                                                      %26 = add nsw i32 %25, %17
                                                                                      %27 = trunc i64 %14 to i32
                                                                                      %28 = add i32 \%26, \%27
                                                                                      %29 = icmp slt i32 %28, %26
                                                                                      %30 = icmp ugt i64 %14, 4294967295
                                                                                      %31 = \text{ or i } 1 \%29, \%30
                                                                                      %32 = \text{ or i } 1 \%23, \%31
                                                                                      br i1 %32, label %pregion for entry.entry.i.preheader, label %vector.ph
                                                                                                                   vector.ph:
                                                                                                                    %n.vec = and i64 %umax, -8
                                                                                                                    %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.splatinsert4 = insertelement <8 x i32> undef, i32 %mul.i, i32 0
                                                                                                                    %broadcast.splat5 = shufflevector <8 x i32> %broadcast.splatinsert4, <8 x
                                                                                                                    ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                    %broadcast.splatinsert6 = insertelement <8 x i32> undef, i32 %mul2.i, i32 0
                                                                                                                    %broadcast.splat7 = shufflevector <8 x i32> %broadcast.splatinsert6, <8 x
                                                                                                                    ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                    br label %vector.bodv
                                                                                                                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 ]
                                                                                                                 %33 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
                                                                                                                 %34 = trunc <8 x i64> %33 to <8 x i32>, !llvm.access.group !12
                                                                                                                 %35 = add nsw <8 x i32> %broadcast.splat5, %34, !llvm.access.group !12
                                                                                                                 \%36 = \text{sext} < 8 \times i32 > \%35 \text{ to } < 8 \times i64 >, !llvm.access.group !12
                                                                                                                 %37 = getelementptr inbounds float, float* %2, <8 x i64> %36,
                                                                                                                ...!llvm.access.group!12
                                                                                                                 %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                ... x float*> %37, i32 4, <8 x i1> <i1 true, i1 
                                                                                                                ... i1 true, i1 true, i1 true, <8 x float> undef), !tbaa !14, !llvm.access.group
                                                                                                                 %38 = add nsw <8 x i32> %broadcast.splat7, %34, !llvm.access.group !12
                                                                                                                 \%39 = \text{sext} < 8 \times i32 > \%38 \text{ to } < 8 \times i64 >, !llvm.access.group !12
                                                                                                                 %40 = getelementptr inbounds float, float* %2, <8 x i64> %39,
                                                                                                                ...!llvm.access.group!12
                                                                                                                 %wide.masked.gather8 = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                ... x float*> %40, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true,
                                                                                                                ... i1 true, i1 true, i1 true, <8 x float> undef), !tbaa !14, !llvm.access.group
                                                                                                                ... !12
                                                                                                                %41 = getelementptr inbounds float, float* %0, <8 x i64> %36,
                                                                                                                ...!llvm.access.group!12
                                                                                                                 %wide.masked.gather9 = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                ... x float*> %41, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true,
                                                                                                                ... i1 true, i1 true, i1 true, <8 x float> undef), !tbaa !14, !llvm.access.group
                                                                                                                 %42 = fmul <8 x float> %wide.masked.gather8, %wide.masked.gather9,
                                                                                                                ...!llvm.access.group!12
                                                                                                                 %43 = \text{getelementptr inbounds float, float* } \%1, <8 \times i64 > \%39,
                                                                                                                ...!llvm.access.group!12
                                                                                                                 %wide.masked.gather10 = call <8 x float>
                                                                                                                ... @llvm.masked.gather.v8f32.v8p0f32(<8 \times float*> %43, i32 4, <8 \times i1> <i1 true,
                                                                                                                ... i1 true, <8 x float>
                                                                                                                ... undef), !tbaa !14, !llvm.access.group !12
                                                                                                                 \%44 = \text{fdiv} < 8 \times \text{float} > \%42, \% \text{wide.masked.gather10}, \% \text{fpmath } \%
                                                                                                                ...!llvm.access.group!12
                                                                                                                 %45 = fsub <8 x float> %wide.masked.gather, %44, !llvm.access.group !12
                                                                                                                 call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %45, <8 x float*>
                                                                                                                ... %37, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1 true, i1 true,
                                                                                                                ... i1 true, i1 true>), !tbaa !14, !llvm.access.group !12
                                                                                                                 \%46 = getelementptr inbounds float, float* \%1, <8 \times i64 > \%36,
                                                                                                                ...!llvm.access.group!12
                                                                                                                 %wide.masked.gather11 = call <8 x float>
                                                                                                                ... @llvm.masked.gather.v8f32.v8p0f32(<8 \times float*> %46, i32 4, <8 \times i1> <i1 true,
                                                                                                                ... i1 true, <8 x float>
                                                                                                                ... undef), !tbaa !14, !llvm.access.group !12
                                                                                                                 %wide.masked.gather12 = call <8 x float>
                                                                                                                ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %41, i32 4, <8 x i1> <i1 true,
                                                                                                                ... i1 true, <8 x float>
                                                                                                                ... undef), !tbaa !14, !llvm.access.group !12
                                                                                                                 %47 = fmul <8 x float> %wide.masked.gather12, %wide.masked.gather12,
                                                                                                                ...!llvm.access.group!12
                                                                                                                 %wide.masked.gather13 = call <8 x float>
                                                                                                                ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %43, i32 4, <8 x i1> <i1 true,
                                                                                                                ... i1 true, i7 true, i1 true,
                                                                                                                ... undef), !tbaa !14, !llvm.access.group !12
                                                                                                                %48 = fdiv <8 x float> %47, %wide.masked.gather13, !fpmath !18,
                                                                                                                ...!llvm.access.group!12
                                                                                                                 %49 = fsub <8 x float> %wide.masked.gather11, %48, !llvm.access.group !12
                                                                                                                 call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %49, <8 x float*>
                                                                                                                ... %46, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1 true, i1 true,
                                                                                                                ... i1 true, i1 true>), !tbaa !14, !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>
                                                                                                                %50 = icmp eq i64 %index.next, %n.vec
                                                                                                                 br i1 %50, label %middle.block, label %vector.body, !llvm.loop !19
                                                                                                           middle.block:
                                                                                                            %cmp.n = icmp eq i64 %umax, %n.vec
                                                                                                            br i1 %cmp.n, label %adi kernel4.exit, label
                                                                                                            ... %pregion for entry.entry.i.preheader
       pregion for entry.entry.i.preheader:
        % local id x.0.ph = phi i64 [ 0, %vector.scevcheck ], [ 0, %9 ], [ %n.vec,
       ... %middle.block ]
        br label %pregion for entry.entry.i
pregion for entry.entry.i:
\frac{1}{2} \frac{1}
... % local id x.0.ph, %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
 %add.i = add nsw i32 %mul.i, %conv.i, !llvm.access.group !12
 %idxprom.i = sext i32 %add.i to i64, !llvm.access.group !12
 %arrayidx.i = getelementptr inbounds float, float* %2, i64 %idxprom.i,
 ..!llvm.access.group!12
 %51 = load float, float* %arrayidx.i, align 4, !tbaa !14, !llvm.access.group
... !12
 %add3.i = add nsw i32 %mul2.i, %conv.i, !llvm.access.group !12
 %idxprom4.i = sext i32 %add3.i to i64, !llvm.access.group !12
 %arrayidx5.i = getelementptr inbounds float, float* %2, i64 %idxprom4.i,
...!llvm.access.group!12
%52 = load float, float* %arrayidx5.i, align 4, !tbaa !14,
...!llvm.access.group!12
 %arrayidx9.i = getelementptr inbounds float, float* %0, i64 %idxprom.i,
...!llvm.access.group!12
%53 = load float, float* %arrayidx9.i, align 4, !tbaa !14,
...!llvm.access.group!12
%mul10.i = fmul float %52, %53, !llvm.access.group !12
%arrayidx15.i = getelementptr inbounds float, float* %1, i64 %idxprom4.i,
...!llvm.access.group!12
%54 = load float, float* %arrayidx15.i, align 4, !tbaa !14,
...!llvm.access.group!12
%div.i = fdiv float %mul10.i, %54, !fpmath !18, !llvm.access.group !12
%sub16.i = fsub float %51, %div.i, !llvm.access.group !12
store float %sub16.i, float* %arrayidx.i, align 4, !tbaa !14,
 ..!llvm.access.group!12
%arrayidx24.i = getelementptr inbounds float, float* %1, i64 %idxprom.i,
...!llvm.access.group!12
%55 = load float, float* %arrayidx24.i, align 4, !tbaa !14,
...!llvm.access.group!12
%56 = load float, float* %arrayidx9.i, align 4, !tbaa !14,
...!llvm.access.group!12
 %mul33.i = fmul float %56, %56, !llvm.access.group !12
%57 = load float, float* %arrayidx15.i, align 4, !tbaa !14,
...!llvm.access.group!12
%div39.i = fdiv float %mul33.i, %57, !fpmath !18, !llvm.access.group !12
%sub40.i = fsub float %55, %div39.i, !llvm.access.group !12
store float %sub40.i, float* %arrayidx24.i, align 4, !tbaa !14,
...!llvm.access.group!12
%58 = add nuw i64\% local id x.0, 1
\%exitcond.not = icmp eq i6\overline{4} \%58, \%umax
br i1 %exitcond.not, label %adi kernel4.exit.loopexit, label
... %pregion for entry.entry.i, !llvm.loop!22
                                                                                                                      F
                                                           adi kernel4.exit.loopexit:
                                                           br label %adi kernel4.exit
```

adi kernel4.exit:

CFG for 'pocl kernel adi kernel4' function

ret void