```
%mul.i.i = shl i64 %7, 8
                                                                               %add.i = add nsw i32 %3, 1, !llvm.access.group !12
                                                                               %mul.i = mul nsw i32 %5, %3
                                                                               %cmp574.i = icmp sgt i32 %4, 0
                                                                               %11 = \text{sext i} 32 \% 5 \text{ to i} 64
                                                                               %12 = \text{sext i} 32 \% 3 \text{ to i} 64
                                                                               %wide.trip.count84.i = zext i32 %4 to i64
                                                                               %arrayidx2786.i = getelementptr inbounds float, float* %2, i64 %12
                                                                               %exitcond.not89.i = icmp eq i32 %4, 1
                                                                               %13 = add nsw i64 %wide.trip.count84.i, -1
                                                                               %min.iters.check.i = icmp ugt i64 %13, 15
                                                                               %ident.check.not.i = icmp eq i32 %5, 1
                                                                               %or.cond.i = and i1 %ident.check.not.i, %min.iters.check.i
                                                                               %14 = add \text{ nsw } i64 \%12.1
                                                                               %scevgep6.i = getelementptr float, float* %2, i64 %14
                                                                               %15 = add nsw i64 %wide.trip.count84.i, %12
                                                                               %scevgep8.i = getelementptr float, float* %2, i64 %15
                                                                               %n.vec.i = and i64 %13, -16
                                                                               %ind.end.i = or i64 %n.vec.i, 1
                                                                               %cmp.n.i = icmp eq i64 %13, %n.vec.i
                                                                               br label %pregion for entry.entry.i
                                                                         pregion for entry.entry.i:
                                                                         % local id x.0 = phi i64 [ 0, %10 ], [ %55, %if.end.i ]
                                                                         %add1.i.i = add nuw nsw i64 % local id x.0, %mul.i.i, !llvm.access.group !12
                                                                          %16 = \text{trunc } i64 \% \text{add} 1.i.i \text{ to } i3\overline{2}, !llvm.access.group !12
                                                                         %conv2.i = add i32 %add.i, %16, !llvm.access.group !12
                                                                         %cmp.i = icmp slt i32 %conv2.i, %5, !llvm.access.group !12
                                                                          br i1 %cmp.i, label %if.then.i, label %if.end.i, !llvm.access.group !12
                                                    if.then.i:
                                                     %add4.i = add nsw i32 %conv2.i, %mul.i, !llvm.access.group !12
                                                     %idxprom.i = sext i32 %add4.i to i64, !llvm.access.group !12
                                                     %arrayidx.i = getelementptr float, float* %1, i64 %idxprom.i,
                                                    ...!llvm.access.group!12
                                                     store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !14,
                                                    ...!llvm.access.group!12
                                                    br i1 %cmp574.i, label %for.body.preheader.i, label %if.end.i,
                                                    ...!llvm.access.group!12
                                                                                                                 F
                                                                       Τ
                                     for.body.preheader.i:
                                     %17 = sext i32 %conv2.i to i64, !llvm.access.group !12
                                     br label %for.body.i, !llvm.access.group !12
                       for.bodv.i:
                        %indvars.iv.next80.i5 = phi i64 [ %indvars.iv.next80.i, %for.body.i ], [ 0,
                        .. %for.body.preheader.i ]
                        %42 = phi float [ %48, %for.body.i ], [ 0.000000e+00, %for.body.preheader.i ]
                        %43 = mul nsw i64 %indvars.iv.next80.i5, %11, !llvm.access.group !12
                        %44 = add nsw i64 %43, %12, !llvm.access.group !12
                        %arrayidx10.i = getelementptr inbounds float, float* %2, i64 %44,
                        ... !llvm.access.group !12
                        %45 = load float, float* %arrayidx10.i, align 4, !tbaa !14,
                        ...!llvm.access.group!12
                       %46 = add nsw i64 %43, %17, !llvm.access.group !12
                        %arrayidx14.i = getelementptr inbounds float, float* %0, i64 %46,
                        ...!llvm.access.group!12
                       %47 = load float, float* %arrayidx14.i, align 4, !tbaa !14,
                        ...!llvm.access.group!12
                        %48 = tail call float @llvm.fmuladd.f32(float %45, float %47, float %42) #2,
                        ...!llvm.access.group!12
                       store float %48, float* %arrayidx.i, align 4, !tbaa !14, !llvm.access.group
                        ... !12
                        %indvars.iv.next80.i = add nuw nsw i64 %indvars.iv.next80.i5, 1,
                        ...!llvm.access.group!12
                        %exitcond85.not.i = icmp eq i64 %indvars.iv.next80.i, %wide.trip.count84.i,
                        ...!llvm.access.group!12
                        br i1 %exitcond85.not.i, label %for.body23.preheader.i, label %for.body.i,
                       ... !llvm.loop !30, !llvm.access.group !12
                                                                                               F
for.body23.preheader.i:
%.lcssa = phi float [ %48, %for.body.i ]
%18 = load float, float* %arrayidx2786.i, align 4, !tbaa !14,
...!llvm.access.group!12
%arrayidx3687.i = getelementptr inbounds float, float* %0, i64 %17,
...!llvm.access.group!12
%19 = load float, float* %arrayidx3687.i, align 4, !tbaa !14,
...!llvm.access.group!12
%neg88.i = fneg float %18, !llvm.access.group !12
%20 = tail call float @llvm.fmuladd.f32(float %neg88.i, float %.lcssa, float
... %19) #2, !llvm.access.group !12
store float %20, float* %arrayidx3687.i, align 4, !tbaa !14,
...!llvm.access.group!12
br i1 %exitcond.not89.i, label %if.end.i, label
... %for.body23.for.body23 crit edge.preheader.i, !llvm.loop !18,
...!llvm.access.group!12
                                                                      F
                             for.body23.for.body23 crit edge.preheader.i:
                             br i1 %or.cond.i, label %vector.memcheck.i, label
                             ... %for.body23.for.body23 crit edge.i.preheader, !llvm.access.group !12
                                                                                                F
vector.memcheck.i:
%21 = add nsw i64 %17, 1, !llvm.access.group !12
%scevgep.i = getelementptr float, float* %0, i64 %21, !llvm.access.group !12
 %scevgep1.i = bitcast float* %scevgep.i to i8*, !llvm.access.group !12
%22 = add nsw i64 %17, %wide.trip.count84.i, !llvm.access.group !12
%scevgep2.i = getelementptr float, float* %0, i64 %22, !llvm.access.group !12
%scevgep45.i = bitcast float* %arrayidx.i to i8*, !llvm.access.group !12
%uglygep.i = getelementptr i8, i8* %scevgep45.i, i64 1, !llvm.access.group
... !12
%bound0.i = icmp ugt i8* %uglygep.i, %scevgep1.i, !llvm.access.group !12
%bound1.i = icmp ult float* %arrayidx.i, %scevgep2.i, !llvm.access.group !12
%found.conflict.i = and i1 %bound1.i, %bound0.i, !llvm.access.group !12
%bound010.i = icmp ult float* %scevgep.i, %scevgep8.i, !llvm.access.group !12
%bound111.i = icmp ult float* %scevgep6.i, %scevgep2.i, !llvm.access.group
%found.conflict12.i = and i1 %bound010.i, %bound111.i, !llvm.access.group !12
%conflict.rdx.i = or i1 %found.conflict12.i, %found.conflict.i,
...!llvm.access.group!12
br i1 %conflict.rdx.i, label %for.body23.for.body23 crit edge.i.preheader,
... label %vector.ph.i, !llvm.access.group !12
                 vector.ph.i:
                  %23 = load float, float* %arrayidx.i, align 4, !tbaa !14, !alias.scope !20,
                  ...!llvm.access.group!12
                  %24 = insertelement <8 x float> undef, float %23, i32 0, !llvm.access.group
                  ... !12
                  %25 = \text{shufflevector} < 8 \times \text{float} > \%24, < 8 \times \text{float} > \text{undef}, < 8 \times \text{i}32 > \%25
                 ... zeroinitializer, !llvm.access.group !12
                  br label %vector.body.i, !llvm.access.group !12
              vector.body.i:
              %index.next.i7 = phi i64 [ %index.next.i, %vector.body.i ], [ 0,
              ... %vector.ph.i ]
              %offset.idx.i = or i64 %index.next.i7, 1, !llvm.access.group !12
              %26 = mul nsw i64 %offset.idx.i, %11, !llvm.access.group !12
              %27 = add nsw i64 %26, %12, !llvm.access.group !12
              %28 = getelementptr inbounds float, float* %2, i64 %27, !llvm.access.group
              %29 = bitcast float* %28 to <8 x float>*, !llvm.access.group !12
              %wide.load.i = load <8 x float>, <8 x float>* %29, align 4, !tbaa !14,
              ... !alias.scope !23, !llvm.access.group !12
              %30 = getelementptr inbounds float, float* %28, i64 8, !llvm.access.group !12
              %31 = bitcast float* %30 to <8 x float>*, !llvm.access.group !12
              %wide.load13.i = load <8 x float>, <8 x float>* %31, align 4, !tbaa !14,
               .. !alias.scope !23, !llvm.access.group !12
              %32 = add nsw i64 %26, %17, !llvm.access.group !12
              %33 = getelementptr inbounds float, float* %0, i64 %32, !llvm.access.group
              ... !12
              %34 = bitcast float* %33 to <8 x float>*, !llvm.access.group !12
              %wide.load14.i = load <8 x float>, <8 x float>* %34, align 4, !tbaa !14,
              ... !alias.scope !25, !noalias !27, !llvm.access.group !12
              %35 = getelementptr inbounds float, float* %33, i64 8, !llvm.access.group !12
              %36 = bitcast float* %35 to <8 x float>*, !llvm.access.group !12
              %wide.load15.i = load <8 x float>, <8 x float>* %36, align 4, !tbaa !14,
              ... !alias.scope !25, !noalias !27, !llvm.access.group !12
              %37 = fneg <8 x float> %wide.load.i, !llvm.access.group !12
              %38 = fneg <8 x float> %wide.load13.i, !llvm.access.group !12
              %39 = tail call < 8 x float > @llvm.fmuladd.v8f32(< 8 x float > %37, < 8 x float > %37,
              ... %25, <8 x float> %wide.load14.i) #2, !llvm.access.group !12
              \%40 = \text{tail call} < 8 \text{ x float} > \text{@llvm.fmuladd.v8f32} (< 8 \text{ x float} > \%38, < 8 \text{ x float} >
              ... %25, <8 x float> %wide.load15.i) #2, !llvm.access.group !12
              store <8 x float> %39, <8 x float> * %34, align 4, !tbaa !14, !alias.scope
              ... !25, !noalias !27, !llvm.access.group !12
              store <8 x float> %40, <8 x float>* %36, align 4, !tbaa !14, !alias.scope
              ... !25, !noalias !27, !llvm.access.group !12
              %index.next.i = add i64 %index.next.i7, 16, !llvm.access.group !12
              %41 = icmp eq i64 %index.next.i, %n.vec.i, !llvm.access.group !12
              br i1 %41, label %middle.block.i, label %vector.body.i, !llvm.loop !28,
              ...!llvm.access.group!12
         middle.block.i:
         br i1 %cmp.n.i, label %if.end.i, label
         ... %for.body23.for.body23 crit edge.i.preheader, !llvm.access.group !12
                            for.body23.for.body23 crit edge.i.preheader:
                            %indvars.iv.next.i9.ph = phi i64 [ 1, ... %for.body23_crit_edge.preheader.i ], [ 1, %vector.memcheck.i ], [
                            ... %ind.end.i, %middle.block.i ]
                            br label %for.body23.for.body23 crit edge.i
                            for.body23.for.body23 crit edge.i:
                             %indvars.iv.next.i9 = phi i64 [ %indvars.iv.next.i,
                             ... %for.body23.for.body23 crit edge.i ], [ %indvars.iv.next.i9.ph,
                            ... %for.body23.for.body23 crit edge.i.preheader ]
                             %.pre.i = load float, float* %arrayidx.i, align 4, !tbaa !14,
                             .. !llvm.access.group !12
                             %49 = mul nsw i64 %indvars.iv.next.i9, %11, !llvm.access.group !12
                             %50 = add nsw i64 %49, %12, !llvm.access.group !12
                             %arrayidx27.i = getelementptr inbounds float, float* %2, i64 %50,
                             ...!llvm.access.group!12
                             %51 = load float, float* %arrayidx27.i, align 4, !tbaa !14,
                             ...!llvm.access.group!12
                             %52 = add nsw i64 %49, %17, !llvm.access.group !12
                             %arrayidx36.i = getelementptr inbounds float, float* %0, i64 %52,
                             ...!llvm.access.group!12
                             %53 = load float, float* %arrayidx36.i, align 4, !tbaa !14,
                             ...!llvm.access.group!12
                            %neg.i = fneg float %51, !llvm.access.group !12
%54 = tail call float @llvm.fmuladd.f32(float %neg.i, float %.pre.i, float
                             ... %53) #2, !llvm.access.group !12
                            store float %54, float* %arrayidx36.i, align 4, !tbaa !14,
                             ...!llvm.access.group!12
                             %indvars.iv.next.i = add nuw nsw i64 %indvars.iv.next.i9, 1,
                             ...!llvm.access.group!12
                             %exitcond.not.i = icmp eq i64 %indvars.iv.next.i, %wide.trip.count84.i,
                             ...!llvm.access.group!12
                             br i1 %exitcond.not.i, label %if.end.i.loopexit, label
                             ... %for.body23.for.body23 crit edge.i, !llvm.loop !31, !llvm.access.group !12
                                                                         if.end.i.loopexit:
                                                                         br label %if.end.i
                                                                if.end.i:
                                                                 \%55 = \text{add nuw nsw } i64 \% \text{ local id } x.0, 1
                                                                 %exitcond.not = icmp eq i\overline{6}4 %55, \overline{2}56
                                                                br i1 %exitcond.not, label %gramschmidt kernel3.exit, label
                                                                 ... %pregion for entry.entry.i, !llvm.loop !32
                                                                                                                        F
                                                                                 Τ
                                                                 gramschmidt kernel3.exit:
```

ret void

CFG for '_pocl_kernel_gramschmidt kernel3' function

%10: