```
%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
                                                                 %cmp572.i = icmp sgt i32 %4, 0
                                                                 %11 = sext i32 %5 to i64
                                                                 %12 = \text{sext i} 32 \% 3 \text{ to i} 64
                                                                 %wide.trip.count.i = zext i32 %4 to i64
                                                                 %arrayidx2786.i = getelementptr inbounds float, float* %2, i64 %12
                                                                 %exitcond85.not89.i = icmp eq i32 %4, 1
                                                                 %13 = add nsw i64 %wide.trip.count.i, -1
                                                                 %min.iters.check.i = icmp ugt i64 %13, 7
                                                                 %ident.check.not.i = icmp eq i32 %5, 1
                                                                 %or.cond.i = and i1 %ident.check.not.i, %min.iters.check.i
                                                                 %n.vec.i = and i64 %13, -8
                                                                 %ind.end.i = or i64 %n.vec.i, 1
                                                                 %broadcast.splatinsert1.i = insertelement <8 x i64> undef, i64 %11, i32 0
                                                                 %broadcast.splat2.i = shufflevector <8 x i64> %broadcast.splatinsert1.i, <8
                                                                ... x i64> undef, <8 x i32> zeroinitializer
                                                                 %broadcast.splatinsert3.i = insertelement <8 x i64> undef, i64 %12, i32 0
                                                                 %broadcast.splat4.i = shufflevector <8 x i64> %broadcast.splatinsert3.i, <8
                                                                ... x i64> undef, <8 x i32> zeroinitializer
                                                                 %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], [%40, %if.end.i]
                                                                 %add1.i.i = add nuw nsw i64 % local id x.0, %mul.i.i, !llvm.access.group !12
                                                                 %14 = \text{trunc } i64 \% \text{add} 1.i.i \text{ to } i3\overline{2}, !llvm.access.group !12
                                                                 %conv2.i = add i32 %add.i, %14, !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
                                                                                   Τ
                                         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 inbounds 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 %cmp572.i, label %for.body.preheader.i, label %if.end.i,
                                         ...!llvm.access.group!12
                                                         Τ
                                                                                               F
                              for.body.preheader.i:
                              %15 = sext i32 %conv2.i to i64, !llvm.access.group !12
                              br label %for.body.i, !llvm.access.group !12
                   for.body.i:
                   %indvars.iv.next.i5 = phi i64 [ %indvars.iv.next.i, %for.body.i ], [ 0,
                   .. %for.body.preheader.i ]
                   %27 = phi float [ %33, %for.body.i ], [ 0.000000e+00, %for.body.preheader.i ]
                   %28 = mul nsw i64 %indvars.iv.next.i5, %11, !llvm.access.group !12
                   %29 = add nsw i64 %28, %12, !llvm.access.group !12
                   %arrayidx10.i = getelementptr inbounds float, float* %2, i64 %29,
                   ..!llvm.access.group!12
                   %30 = load float, float* %arrayidx10.i, align 4, !tbaa !14,
                   ...!llvm.access.group!12
                   %31 = add nsw i64 %28, %15, !llvm.access.group !12
                   %arrayidx14.i = getelementptr inbounds float, float* %0, i64 %31,
                   ...!llvm.access.group!12
                   %32 = load float, float* %arrayidx14.i, align 4, !tbaa !14,
                   ...!llvm.access.group!12
                   %33 = tail call float @llvm.fmuladd.f32(float %30, float %32, float %27) #5,
                   ...!llvm.access.group!12
                   store float %33, float* %arrayidx.i, align 4, !tbaa !14, !llvm.access.group
                   ... !12
                   %indvars.iv.next.i = add nuw nsw i64 %indvars.iv.next.i5, 1,
                   ...!llvm.access.group!12
                   %exitcond.not.i = icmp eq i64 %indvars.iv.next.i, %wide.trip.count.i,
                   ..!llvm.access.group!12
                   br i1 %exitcond.not.i, label %for.body23.preheader.i, label %for.body.i,
                   ...!llvm.loop!22,!llvm.access.group!12
for.body23.preheader.i:
%.lcssa = phi float [ %33, %for.body.i ]
%16 = load float, float* %arrayidx2786.i, align 4, !tbaa !14,
...!llvm.access.group!12
%arrayidx3687.i = getelementptr inbounds float, float* %0, i64 %15,
...!llvm.access.group!12
%17 = load float, float* %arrayidx3687.i, align 4, !tbaa !14,
...!llvm.access.group!12
%neg88.i = fneg float %16, !llvm.access.group !12
%18 = tail call float @llvm.fmuladd.f32(float %neg88.i, float %.lcssa, float
... %17) #5, !llvm.access.group !12
store float %18, float* %arrayidx3687.i, align 4, !tbaa !14,
...!llvm.access.group!12
br i1 %exitcond85.not89.i, label %if.end.i, label
... %for.body23.for.body23 crit edge.preheader.i, !llvm.loop !18,
...!llvm.access.group!12
                                                         F
                                for.bodv23.for.bodv23 crit edge.preheader.i:
                                br i1 %or.cond.i, label %vector.ph.i, label
                                ... %for.body23.for.body23 crit edge.i.preheader, !llvm.access.group !12
                                                                                       F
          vector.ph.i:
           %broadcast.splatinsert.i = insertelement <8 x float*> undef, float*
           .. %arrayidx.i, i32 0, !llvm.access.group !12
          %broadcast.splat.i = shufflevector <8 x float*> %broadcast.splatinsert.i, <8
          ... x float*> undef, <8 x i32> zeroinitializer, !llvm.access.group !12
          %broadcast.splatinsert6.i = insertelement <8 x i64> undef, i64 %15, i32 0,
          ...!llvm.access.group!12
          %broadcast.splat7.i = shufflevector <8 x i64> %broadcast.splatinsert6.i, <8
          ... x i64> undef, <8 x i32> zeroinitializer, !llvm.access.group !12
          br label %vector.body.i, !llvm.access.group !12
       vector.body.i:
       %vec.ind.next.i9 = phi < 8 \times i64 > [%vec.ind.next.i, %vector.body.i ], [< i64
       ... 1, i64 2, i64 3, i64 4, i64 5, i64 6, i64 7, i64 8>, %vector.ph.i ]
       %index.next.i7 = phi i64 [ %index.next.i, %vector.body.i ], [ 0,
       ... %vector.ph.i ]
       %wide.masked.gather.i = tail call <8 x float>
       ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %broadcast.splat.i, 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) #5, !tbaa !14, !llvm.access.group !12
       %19 = mul nsw <8 x i64> %vec.ind.next.i9, %broadcast.splat2.i,
       ...!llvm.access.group!12
       %20 = add nsw <8 x i64> %19, %broadcast.splat4.i, !llvm.access.group !12
       %21 = getelementptr inbounds float, float* %2, <8 x i64> %20,
       ...!llvm.access.group!12
       %wide.masked.gather5.i = tail call <8 x float>
       ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %21, i32 4, <8 x i1> <i1 true,
       ... i1 true, <8 x float>
       ... undef) #5, !tbaa !14, !llvm.access.group !12
       %22 = add nsw <8 x i64> %19, %broadcast.splat7.i, !llvm.access.group !12
       \%23 = \text{getelementptr inbounds float, float* } \%0, <8 \times i64 > \%22,
       ...!llvm.access.group!12
       %wide.masked.gather8.i = tail call <8 x float>
       ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %23, i32 4, <8 x i1> <i1 true,
       ... il true, ce x float
       ... undef) #5, !tbaa !14, !llvm.access.group !12
       %24 = fneg <8 x float> %wide.masked.gather5.i, !llvm.access.group !12
       %25 = tail call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %24, <8 x float>
       ... %wide.masked.gather.i, <8 x float> %wide.masked.gather8.i) #5,
       ...!llvm.access.group!12
       tail call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %25, <8 x
       ... float*> %23, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1
       ... true, i1 true, i1 true>) #5, !tbaa !14, !llvm.access.group !12
       %index.next.i = add i64 %index.next.i7, 8, !llvm.access.group !12
       %vec.ind.next.i = add <8 x i64> %vec.ind.next.i9, <i64 8, i64 8, i64 8, i64
       ... 8, i64 8, i64 8, i64 8, i64 8>, !llvm.access.group !12
       %26 = icmp eq i64 %index.next.i, %n.vec.i, !llvm.access.group !12
       br i1 %26, label %middle.block.i, label %vector.body.i, !llvm.loop !20,
       ...!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_crit_edge.i.preheader:
                           %indvars.iv.next80.i11.ph = phi i64 [ 1,
                           ... %for.body23.for.body23 crit edge.preheader.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.next80.i1\overline{1} = \overline{p}hi i64 [ %indvars.iv.next80.i,
                            ... %for.body23_crit_edge.i ], [ %indvars.iv.next80.i11.ph,
                            ... %for.body23.for.body23_crit_edge.i.preheader ]
                            %.pre.i = load float, float* %arrayidx.i, align 4, !tbaa !14,
                            ...!llvm.access.group!12
                            %34 = mul nsw i64 %indvars.iv.next80.i11, %11, !llvm.access.group !12
                            %35 = add nsw i64 %34, %12, !llvm.access.group !12
                            %arrayidx27.i = getelementptr inbounds float, float* %2, i64 %35,
                            ...!llvm.access.group!12
                            %36 = load float, float* %arrayidx27.i, align 4, !tbaa !14,
                            ...!llvm.access.group!12
                            %37 = add nsw i64 %34, %15, !llvm.access.group !12
                            %arrayidx36.i = getelementptr inbounds float, float* %0, i64 %37,
                            ...!llvm.access.group!12
                            %38 = load float, float* %arrayidx36.i, align 4, !tbaa !14,
                            ...!llvm.access.group!12
                            %neg.i = fneg float %36, !llvm.access.group !12
%39 = tail call float @llvm.fmuladd.f32(float %neg.i, float %.pre.i, float
                            ... %38) #5, !llvm.access.group !12
                            store float %39, float* %arrayidx36.i, align 4, !tbaa !14,
                            ...!llvm.access.group!12
                            %indvars.iv.next80.i = add nuw nsw i64 %indvars.iv.next80.i11, 1,
                            ...!llvm.access.group!12
                            %exitcond85.not.i = icmp eq i64 %indvars.iv.next80.i, %wide.trip.count.i,
                            ...!llvm.access.group!12
                            br i1 %exitcond85.not.i, label %if.end.i.loopexit, label
                            ... %for.body23.for.body23 crit edge.i, !llvm.loop !23, !llvm.access.group !12
                                                    if.end.i.loopexit:
                                                    br label %if.end.i
                                                          if.end.i:
                                                          %40 = add nuw nsw i64 % local id x.0.1
                                                          %exitcond.not = icmp eq i64 %40, 256
                                                          br i1 %exitcond.not, label %gramschmidt kernel3.exit, label
                                                          ... %pregion for entry.entry.i, !llvm.loop !24
```

%10:

gramschmidt kernel3.exit: