```
%10:
                                                                                                %11 = shl i64 %7, 8
                                                                                                %add.i = add nsw i32 %3, 1
                                                                                                %mul.i = mul nsw i32 %5, %3
                                                                                                %cmp574.i = icmp sgt i32 %4, 0
                                                                                                %12 = \text{sext i} 32 \% 5 \text{ to i} 64
                                                                                                %13 = \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 %13
                                                                                                \%exitcond89.i = icmp eq i32 \%4, 1
                                                                                                %14 = add \text{ nsw } i64 \text{ %wide.trip.count84.i, -1}
                                                                                                %min.iters.check.i = icmp ugt i64 %14, 15
                                                                                                %ident.check.i = icmp eq i32 %5, 1
                                                                                                %or.cond.i = and i1 %ident.check.i, %min.iters.check.i
                                                                                                %15 = add nsw i64 %13, 1
                                                                                                %scevgep6.i = getelementptr float, float* %2, i64 %15
                                                                                                %16 = add nsw i64 %wide.trip.count84.i, %13
                                                                                                %scevgep8.i = getelementptr float, float* %2, i64 %16
                                                                                                %n.vec.i = and i64 %14, -16
                                                                                                %ind.end.i = or i64 %n.vec.i, 1
                                                                                                %cmp.n.i = icmp eq i64 %14, %n.vec.i
                                                                                                br label %pregion for entry.entry.i
                                                                                                           pregion for entry.entry.i:
                                                                                                           % local id x.0 = phi i64 [ 0, %10 ], [ %57, %if.end.i ]
                                                                                                           \%\overline{17} = \overline{a}d\overline{d} nuw nsw i64 % local id x.0, %11
                                                                                                           %18 = \text{trunc } i64 \%17 \text{ to } i3\overline{2}
                                                                                                           %conv2.i = add i32 %add.i, %18
                                                                                                           %cmp.i = icmp slt i32 %conv2.i, %5
                                                                                                           br i1 %cmp.i, label %if.then.i, label %if.end.i
                                                           if.then.i:
                                                            %add4.i = add nsw i32 %conv2.i, %mul.i
                                                            %idxprom.i = sext i32 %add4.i to i64
                                                            %arrayidx.i = getelementptr float, float* %1, i64 %idxprom.i
                                                            store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !12,
                                                            ..!llvm.access.group!16
                                                           br i1 %cmp574.i, label %for.body.preheader.i, label %if.end.i
                                                          for.body.preheader.i:
                                                           %19 = \text{sext i} 32 \% \text{conv} 2.i \text{ to i} 64
                                                           br label %for.bodv.i
                          for.bodv.i:
                           %indvars.iv.next80.i5 = phi i64 [ %indvars.iv.next80.i, %for.body.i ], [ 0,
                          ... %for.bodv.preheader.i 1
                           %44 = phi float [ %50, %for.body.i ], [ 0.000000e+00, %for.body.preheader.i ]
                           %45 = mul nsw i64 %indvars.iv.next80.i5, %12
                           %46 = add nsw i64 %45, %13
                          %arrayidx10.i = getelementptr inbounds float, float* %2, i64 %46
                          %47 = load float, float* %arrayidx10.i, align 4, !tbaa !12
                           %48 = add nsw i64 %45, %19
                          %arrayidx14.i = getelementptr inbounds float, float* %0, i64 %48 %49 = load float, float* %arrayidx14.i, align 4, !tbaa !12
                          %50 = tail call float @llvm.fmuladd.f32(float %47, float %49, float %44) #2
                          store float %50, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                          ... !16
                          %indvars.iv.next80.i = add nuw nsw i64 %indvars.iv.next80.i5, 1
                          %exitcond85.i = icmp eq i64 %indvars.iv.next80.i, %wide.trip.count84.i
                          br i1 %exitcond85.i, label %for.body23.preheader.i, label %for.body.i
                                                                                                           F
for.body23.preheader.i:
%.lcssa = phi float [ %50, %for.body.i ]
%20 = load float, float* %arrayidx2786.i, align 4, !tbaa !12
%arrayidx3687.i = getelementptr inbounds float, float* %0, i64 %19
%21 = load float, float* %arrayidx3687.i, align 4, !tbaa !12
%neg88.i = fneg float %20
%22 = tail call float @llvm.fmuladd.f32(float %neg88.i, float %.lcssa, float
... %21) #2
store float %22, float* %arrayidx3687.i, align 4, !tbaa !12,
...!llvm.access.group!16
br i1 %exitcond89.i, label %if.end.i, label
... %for.body23.for.body23 crit edge.preheader.i
                                                                              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
           vector.memcheck.i:
           %23 = add nsw i64 %19, 1
           %scevgep.i = getelementptr float, float* %0, i64 %23
           %scevgep1.i = bitcast float* %scevgep.i to i8*
           %24 = add nsw i64 %19, %wide.trip.count84.i
           %scevgep2.i = getelementptr float, float* %0, i64 %24 %scevgep45.i = bitcast float* %arrayidx.i to i8* %uglygep.i = getelementptr i8, i8* %scevgep45.i, i64 1 %bound0.i = icmp ugt i8* %uglygep.i, %scevgep1.i
           %bound1.i = icmp ult float* %arrayidx.i, %scevgep2.i
           %found.conflict.i = and i1 %bound1.i, %bound0.i
           %bound010.i = icmp ult float* %scevgep.i, %scevgep8.i
           %bound111.i = icmp ult float* %scevgep6.i, %scevgep2.i
           %found.conflict12.i = and i1 %bound010.i, %bound111.i
           %conflict.rdx.i = or i1 %found.conflict12.i, %found.conflict.i
           br i1 %conflict.rdx.i, label %for.body23.for.body23 crit edge.i.preheader,
           ... label %vector.ph.i
                                  Τ
                                                                                        F
                             vector.ph.i:
                              %25 = load float, float* %arrayidx.i, align 4, !tbaa !12, !alias.scope !18
                              %26 = insertelement <8 x float> undef, float %25, i32 0
                              %27 = shufflevector <8 x float> %26, <8 x float> undef, <8 x i32>
                              ... zeroinitializer
                              br label %vector.body.i
                   vector.body.i:
                    %index.next.i7 = phi i64 [ %index.next.i, %vector.body.i ], [ 0,
                    ... %vector.ph.i ]
                    %offset.id\dot{x}.i = or i64 %index.next.i7, 1
                    %28 = mul nsw i64 %offset.idx.i, %12
                    %29 = add nsw i64 %28, %13
                    %30 = getelementptr inbounds float, float* %2, i64 %29
                    %31 = bitcast float* %30 to <8 x float>*
                    %wide.load.i = load <8 x float>, <8 x float>* \%31, align 4, !tbaa !12,
                    ... !alias.scope !21
                    %32 = getelementptr inbounds float, float* %30, i64 8
                    %33 = bitcast float* %32 to <8 x float>*
                    %wide.load14.i = load <8 x float>, <8 x float>* %33, align 4, !tbaa !12,
                    ... !alias.scope !21
                    %34 = add nsw i64 %28, %19
                    %35 = getelementptr inbounds float, float* %0, i64 %34
                    %36 = bitcast float* %35 to <8 x float>*
                    %wide.load15.i = load <8 x float>, <8 x float>* %36, align 4, !tbaa !12,
                    ... !alias.scope !23, !noalias !25
                    %37 = getelementptr inbounds float, float* %35, i64 8
                    %38 = bitcast float* %37 to <8 x float>*
                    %wide.load16.i = load <8 x float>, <8 x float>* %38, align 4, !tbaa !12,
                    ... !alias.scope !23, !noalias !25
                    %39 = \text{fneg} < 8 \text{ x float} > \% \text{wide.load.i}
                    %40 = fneg < 8 x float > %wide.load14.i
                    %41 = tail call < 8 \times float > @llvm.fmuladd.v8f32(< 8 \times float > %39, < 8 \times float > %39,
                    ... %27, <8 x float> %wide.load15.i) #2
                    \%42 = \text{tail call} < 8 \times \text{float} > \text{@llvm.fmuladd.v8f32} (< 8 \times \text{float} > \%40, < 8 \times \text{float} >
                   ... %27, <8 x float> %wide.load16.i) #2
                    store <8 x float> %41, <8 x float>* %36, align 4, !tbaa !12, !alias.scope
                   ...!23, !noalias!25, !llvm.access.group!16
                    store <8 x float> %42, <8 x float>* %38, align 4, !tbaa !12, !alias.scope
                   ... !23, !noalias !25, !llvm.access.group !16
                    %index.next.i = add i64 %index.next.i7, 16
                    %43 = icmp eq i64 %index.next.i, %n.vec.i
                    br i1 %43, label %middle.block.i, label %vector.body.i, !llvm.loop !26
                                                                                                       F
                   middle.block.i:
                    br i1 %cmp.n.i, label %if.end.i, label
                   ... %for.body23.for.body23 crit edge.i.preheader
                                  Τ
                                                                      F
                               for.body23.for.body23 crit edge.i.preheader:
                                %indvars.iv.next.i9.ph = phi i64 [ 1,
                               ... %for.body23.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 !12
                                     %51 = mul nsw i64 %indvars.iv.next.i9, %12
                                     %52 = add nsw i64 %51, %13
                                     %arrayidx27.i = getelementptr inbounds float, float* %2, i64 %52
                                     %53 = load float, float* %arrayidx27.i, align 4, !tbaa !12
                                     %54 = add nsw i64 %51, %19
                                     %arrayidx36.i = getelementptr inbounds float, float* %0, i64 %54
                                     %55 = load float, float* %arrayidx36.i, align 4, !tbaa !12
                                     %neg.i = fneg float %53
                                     %56 = tail call float @llvm.fmuladd.f32(float %neg.i, float %.pre.i, float
                                     ... %55) #2
                                     store float %56, float* %arrayidx36.i, align 4, !tbaa !12,
                                     ...!llvm.access.group!16
                                     %indvars.iv.next.i = add nuw nsw i64 %indvars.iv.next.i9, 1
                                     %exitcond.i = icmp eq i64 %indvars.iv.next.i, %wide.trip.count84.i
                                     br i1 %exitcond.i, label %if.end.i.loopexit, label
                                     ... %for.body23.for.body23 crit edge.i, !llvm.loop !28
                                                                                   if.end.i.loopexit:
                                                                                    br label %if.end.i
                                                                            if.end.i:
                                                                            %57 = add nuw nsw i64 % local id x.0, 1
                                                                            \%exitcond = icmp eq i64 \%57, 256
                                                                            br i1 %exitcond, label %gramschmidt kernel3.exit, label
                                                                            ... %pregion for entry.entry.i, !llvm.loop !29
                                                                           gramschmidt kernel3.exit:
                                                                            ret void
```

CFG for 'pocl kernel gramschmidt kernel3' function