```
%12 = \text{shl } i64 \%8, 3
                                                                                      %cmp638.i = icmp sgt i32 %5, 0
                                                                                      %13 = sext i32 %4 to i64
                                                                                      %wide.trip.count.i = zext i32 %5 to i64
                                                                                      %14 = add nsw i64 %wide.trip.count.i, -1
                                                                                      %xtraiter.i = and i64 %wide.trip.count.i, 3
                                                                                      %15 = icmp ult i64 %14, 3
                                                                                      %unroll iter.i = sub nuw nsw i64 %wide.trip.count.i, %xtraiter.i
                                                                                      %lcmp.mod.i = icmp eq i64 %xtraiter.i, 0
                                                                                      br label %pregion for entry.pregion for init.i
                                                                                      pregion for entry.pregion for init.i:
                                                                                      % local id y.0 = phi i64 [0, \sqrt{3}10], [%54, %pregion_for_end.i]
                                                                                      \%\overline{1}6 = \overline{a}d\overline{d} nuw nsw i64 % local id y.0, %12
                                                                                       %conv2.i = trunc i64 %16 to i32
                                                                                      %cmp.i = icmp slt i32 %conv2.i, %3
                                                                                      %mul.i = mul nsw i32 %conv2.i, %4
                                                                                      %mul8.i = mul nsw i32 %conv2.i, %5
                                                                                       %17 = \text{sext i} 32 \% \text{mul} 8.i \text{ to i} 64
                                                                                      br label %pregion for entry.entry.i
                                                             pregion for entry.entry.i:
                                                              % [ocal] id [x.0] = phi i64 [ 0, %pregion for entry.pregion for init.i ], [
                                                             ... \( \bar{855}, \bar{8} \) if end. i \( \bar{1} \)
                                                              %18 = add nuw nsw i64 % local id x.0, %11
                                                              %conv.i = trunc i64 %18 to i32
                                                              %cmp4.i = icmp slt i32 %conv.i, %4
                                                              %or.cond.i = and i1 %cmp.i, %cmp4.i
                                                              br i1 %or.cond.i, label %if.then.i, label %if.end.i
                                                                                                                     F
                                if.then.i:
                                 %add.i = add nsw i32 %mul.i, %conv.i
                                 %idxprom.i = sext i32 %add.i to i64
                                 %arrayidx.i = getelementptr inbounds float, float* %2, i64 %idxprom.i
                                 store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !12,
                                 ...!llvm.access.group!16
                                 br i1 %cmp638.i, label %for.body.lr.ph.i, label %if.end.i
                                                                                         F
for.body.lr.ph.i:
%sext.i = shl i64 %18, 32
%19 = ashr exact i64 %sext.i, 32
br i1 %15, label %if.end.loopexit.unr-lcssa.i, label %for.body.i.preheader
                 Τ
                                                          F
                                               for.body.i.preheader:
                                                br label %for.bodv.i
                 for.body.i:
                 %niter.nsub.3.i7 = phi i64 [ %niter.nsub.3.i, %for.body.i ], [
                 ... %unroll iter.i, %for.body.i.preheader ]
                 %indvars.iv.next.3.i4 = phi i64 [ %indvars.iv.next.3.i, %for.body.i ], [ 0,
                 ... %for.body.i.preheader ]
                 %20 = phi float [ %44, %for.body.i ], [ 0.000000e+00, %for.body.i.preheader ]
                 %21 = add nsw i64 %indvars.iv.next.3.i4, %17
                 %arrayidx11.i = getelementptr inbounds float, float* %0, i64 %21
                 %22 = load float, float* %arrayidx11.i, align 4, !tbaa !12
                 %23 = mul nsw i64 %indvars.iv.next.3.i4, %13
                 %24 = add nsw i64 %23, %19
                 %arrayidx15.i = getelementptr inbounds float, float* %1, i64 %24
                 %25 = load float, float* %arrayidx15.i, align 4, !tbaa !12
                 %26 = tail call float @llvm.fmuladd.f32(float %22, float %25, float %20) #2
                 store float %26, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                 ... !16
                 %indvars.iv.next.i = or i64 %indvars.iv.next.3.i4, 1
                 %27 = add nsw i64 %indvars.iv.next.i, %17
                 %arrayidx11.1.i = getelementptr inbounds float, float* %0, i64 %27
                 %28 = load float, float* %arrayidx11.1.i, align 4, !tbaa !12
                 %29 = mul nsw i64 %indvars.iv.next.i, %13
                 %30 = add nsw i64 %29, %19
                 %arrayidx15.1.i = getelementptr inbounds float, float* %1, i64 %30
                 %31 = load float, float* %arrayidx15.1.i, align 4, !tbaa !12
                 %32 = tail call float @llvm.fmuladd.f32(float %28, float %31, float %26) #2
                 store float %32, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                 .. !16
                 %indvars.iv.next.1.i = or i64 %indvars.iv.next.3.i4, 2
                 %33 = add nsw i64 %indvars.iv.next.1.i, %17
                 %arrayidx11.2.i = getelementptr inbounds float, float* %0, i64 %33
                 %34 = load float, float* %arrayidx11.2.i, align 4, !tbaa !12
                 %35 = mul nsw i64 %indvars.iv.next.1.i, %13
                 %36 = add nsw i64 %35, %19
                 %arrayidx15.2.i = getelementptr inbounds float, float* %1, i64 %36
                 %37 = load float, float* %arrayidx15.2.i, align 4, !tbaa !12
                 %38 = tail call float @llvm.fmuladd.f32(float %34, float %37, float %32) #2
                 store float %38, float* %arravidx.i, align 4, !tbaa !12, !llvm.access.group
                 .. !16
                 %indvars.iv.next.2.i = or i64 %indvars.iv.next.3.i4, 3
                 %39 = add nsw i64 %indvars.iv.next.2.i, %17
                 %arrayidx11.3.i = getelementptr inbounds float, float* %0, i64 %39
                 %40 = load float, float* %arrayidx11.3.i, align 4, !tbaa !12
                 %41 = mul nsw i64 %indvars.iv.next.2.i, %13
                 %42 = add nsw i64 %41, %19
                 %arrayidx15.3.i = getelementptr inbounds float, float* %1, i64 %42
                 %43 = load float, float* %arrayidx15.3.i, align 4, !tbaa !12
                 %44 = tail call float @llvm.fmuladd.f32(float %40, float %43, float %38) #2
                 store float %44, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                 .. !16
                 %indvars.iv.next.3.i = add nuw nsw i64 %indvars.iv.next.3.i4, 4
                 %niter.nsub.3.i = add i64 %niter.nsub.3.i7, -4
                 %niter.ncmp.3.i = icmp eq i64 %niter.nsub.3.i, 0
                 br i1 %niter.ncmp.3.i, label %if.end.loopexit.unr-lcssa.i.loopexit, label
                 ... %for.body.i
        if.end.loopexit.unr-lcssa.i.loopexit:
        %.lcssa = phi float [ %44, %for.body.i ]
        %indvars.iv.next.3.i.lcssa = phi i64 [ %indvars.iv.next.3.i, %for.body.i ]
        br label %if.end.loopexit.unr-lcssa.i
if.end.loopexit.unr-lcssa.i:
 \%45 = \text{phi float } [\ 0.000000e+00, \% \text{for.body.lr.ph.i}\ ], [\ \%.lcssa,]
 ... %if.end.loopexit.unr-lcssa.i.loopexit ]
 %46 = phi i64 [ 0, %for.body.lr.ph.i ], [ %indvars.iv.next.3.i.lcssa,
 ... %if.end.loopexit.unr-lcssa.i.loopexit ]
 br i1 %lcmp.mod.i, label %if.end.i, label %for.body.epil.i.preheader
                                           for.body.epil.i.preheader:
                                            br label %for.body.epil.i
                   for.body.epil.i:
                   %epil.iter.sub.i13 = phi i64 [ %epil.iter.sub.i, %for.body.epil.i ], [
                     %xtraiter.i, %for.body.epil.i.preheader ]
                   %indvars.iv.next.epil.i11 = phi i64 [ %indvars.iv.next.epil.i,
                   ... %for.body.epil.i ], [ %46, %for.body.epil.i.preheader ] %47 = phi float [ %53, %for.body.epil.i ], [ %45, %for.body.epil.i.preheader
                   %48 = add nsw i64 %indvars.iv.next.epil.i11, %17
                   %arrayidx11.epil.i = getelementptr inbounds float, float* %0, i64 %48
                   %49 = load float, float* %arrayidx11.epil.i, align 4, !tbaa !12
                   %50 = mul nsw i64 %indvars.iv.next.epil.i11, %13
                   %51 = add nsw i64 %50, %19
                   %arrayidx15.epil.i = getelementptr inbounds float, float* %1, i64 %51 %52 = load float, float* %arrayidx15.epil.i, align 4, !tbaa !12
                   %53 = tail call float @llvm.fmuladd.f32(float %49, float %52, float %47) #2 store float %53, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                   ... !16
                   %indvars.iv.next.epil.i = add nuw nsw i64 %indvars.iv.next.epil.i11, 1
                   %epil.iter.sub.i = add nsw i64 %epil.iter.sub.i13, -1
                   %epil.iter.cmp.i = icmp eq i64 %epil.iter.sub.i, 0
                   br il %epil.iter.cmp.i, label %if.end.i.loopexit, label %for.body.epil.i,
                   ...!llvm.loop!19
                                                                                F
                                                        if.end.i.loopexit:
                                                         br label %if.end.i
                                                      if.end.i:
                                                      %55 = add nuw nsw i64 \% local id x.0, 1
                                                      %exitcond = icmp eq i64 \%55, 3\overline{2}
                                                      br il %exitcond, label %pregion for end.i, label %pregion for entry.entry.i,
                                                      ...!llvm.loop!23
                                                                        Τ
                                                                                                                  F
                                                                      pregion for end.i:
                                                                       0.054 = add nuw nsw i64 % local id y.0, 1
                                                                       \%exitcond14 = icmp eq i6\overline{4} %54\overline{7} 8
                                                                       br i1 %exitcond14, label %mm3 kernel1.exit, label
                                                                       ... %pregion for entry.pregion for init.i, !llvm.loop !21
                                                                          mm3 kernel1.exit:
```

ret void

CFG for 'pocl kernel mm3 kernel1' function

%10:

%11 = shl i64 %7, 5