```
%14 = shl i64 %10, 5
                                                                                  %15 = \text{shl } i64 \%11, 3
                                                                                   %cmp639.i = icmp sgt i32 %5, 0
                                                                                   %16 = \text{sext i} 32 \% 4 \text{ to i} 64
                                                                                   %wide.trip.count.i = zext i32 %5 to i64
                                                                                   %17 = add nsw i64 %wide.trip.count.i, -1
                                                                                   %xtraiter.i = and i64 %wide.trip.count.i, 3
                                                                                   %18 = icmp ult i64 %17, 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}13], [%57, %pregion_for_end.i]
                                                                                   \%\overline{19} = \overline{a}d\overline{d} nuw nsw i64 % local id y.0, %15
                                                                                   %conv2.i = trunc i64 %19 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
                                                                                   %20 = \text{sext i} 32 \% \text{mul} 8.i \text{ to i} 64
                                                                                   br label %pregion for entry.entry.i
                                                           pregion for entry.entry.i:
                                                           ... \( \bar{8} 58, \bar{8} \) if end. i \( \bar{1} \)
                                                           %21 = add nuw nsw i64 % local id x.0, %14
                                                            %conv.i = trunc i64 %21 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* %0, i64 %idxprom.i
                                store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !12,
                               ...!llvm.access.group!16
                                br i1 %cmp639.i, label %for.body.lr.ph.i, label %if.end.i
                                                                                      F
for.body.lr.ph.i:
%sext.i = shl i64 %21, 32
%22 = ashr exact i64 %sext.i, 32
br i1 %18, label %if.end.loopexit.unr-lcssa.i, label %for.body.i.preheader
                                                       F
                 Τ
                                             for.body.i.preheader:
                                              br label %for.body.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 ]
                %23 = phi float [ %47, %for.body.i ], [ 0.000000e+00, %for.body.i.preheader ]
                 %24 = add nsw i64 %indvars.iv.next.3.i4, %20
                %arrayidx11.i = getelementptr inbounds float, float* %1, i64 %24
                %25 = load float, float* %arrayidx11.i, align 4, !tbaa !12
                %mul12.i = fmul float %25, %7
                %26 = mul nsw i64 %indvars.iv.next.3.i4, %16
                %27 = add nsw i64 %26, %22
                 %arrayidx16.i = getelementptr inbounds float, float* %2, i64 %27
                %28 = load float, float* %arrayidx16.i, align 4, !tbaa !12
                 %29 = tail call float @llvm.fmuladd.f32(float %mul12.i, float %28, float
                ... %23) #2
                store float %29, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                 %indvars.iv.next.i = or i64 %indvars.iv.next.3.i4, 1
                 %30 = add nsw i64 %indvars.iv.next.i, %20
                 %arrayidx11.1.i = getelementptr inbounds float, float* %1, i64 %30
                 %31 = load float, float* %arrayidx11.1.i, align 4, !tbaa !12
                 %mul12.1.i = fmul float %31, %7
                %32 = mul nsw i64 %indvars.iv.next.i, %16
                %33 = add nsw i64 %32, %22
                %arrayidx16.1.i = getelementptr inbounds float, float* %2, i64 %33
                %34 = load float, float* %arrayidx16.1.i, align 4, !tbaa !12
                %35 = tail call float @llvm.fmuladd.f32(float %mul12.1.i, float %34, float
                ... %29) #2
                store float %35, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                %indvars.iv.next.1.i = or i64 %indvars.iv.next.3.i4, 2
                 %36 = add nsw i64 %indvars.iv.next.1.i, %20
                 %arrayidx11.2.i = getelementptr inbounds float, float* %1, i64 %36
                %37 = load float, float* %arrayidx11.2.i, align 4, !tbaa !12
                 %mul12.2.i = fmul float %37, %7
                 %38 = mul nsw i64 %indvars.iv.next.1.i, %16
                %39 = add nsw i64 %38, %22
                %arrayidx16.2.i = getelementptr inbounds float, float* %2, i64 %39
                %40 = load float, float* %arrayidx16.2.i, align 4, !tbaa !12
                %41 = tail call float @llvm.fmuladd.f32(float %mul12.2.i, float %40, float
                ... %35) #2
                store float %41, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                ... !16
                 %indvars.iv.next.2.i = or i64 %indvars.iv.next.3.i4, 3
                %42 = add nsw i64 %indvars.iv.next.2.i, %20
                %arrayidx11.3.i = getelementptr inbounds float, float* %1, i64 %42
                %43 = load float, float* %arrayidx11.3.i, align 4, !tbaa !12
                %mul12.3.i = fmul float %43, %7
                %44 = mul nsw i64 %indvars.iv.next.2.i, %16
                 %45 = add nsw i64 %44, %22
                %arrayidx16.3.i = getelementptr inbounds float, float* %2, i64 %45
                 %46 = load float, float* %arrayidx16.3.i, align 4, !tbaa !12
                 %47 = tail call float @llvm.fmuladd.f32(float %mul12.3.i, float %46, float
                ... %41) #2
                store float %47, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                %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 [ %47, %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:
  %48 = phi float [ 0.000000e+00, %for.body.lr.ph.i ], [ %.lcssa,
  ... %if.end.loopexit.unr-lcssa.i.loopexit ]
  %49 = 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 ], [ %49, %for.body.epil.i.preheader ]
                    %50 = phi float [ %56, %for.body.epil.i ], [ %48, %for.body.epil.i.preheader
                    %51 = add nsw i64 %indvars.iv.next.epil.i11, %20
                    %arrayidx11.epil.i = getelementptr inbounds float, float* %1, i64 %51
                    %52 = load float, float* %arrayidx11.epil.i, align 4, !tbaa !12
                    %mul12.epil.i = fmul float %52, %7
                    %53 = mul nsw i64 %indvars.iv.next.epil.i11, %16
                    %54 = add nsw i64 %53, %22
                    %arrayidx16.epil.i = getelementptr inbounds float, float* %2, i64 %54
                    %55 = load float, float* %arrayidx16.epil.i, align 4, !tbaa !12
                    %56 = tail call float @llvm.fmuladd.f32(float %mul12.epil.i, float %55,
                    ... float %50) #2
                    store float %56, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                    %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 i1 %epil.iter.cmp.i, label %if.end.i.loopexit, label %for.body.epil.i,
                    ... !llvm.loop !19
                                                       if.end.i.loopexit:
                                                        br label %if.end.i
                                                    if.end.i:
                                                    %58 = add nuw nsw i64 %_local_id_x.0, 1
                                                    %exitcond = icmp eq i64 \%58, 3\overline{2}
                                                    br i1 %exitcond, label %pregion for end.i, label %pregion for entry.entry.i,
                                                    ...!llvm.loop!23
                                                                                                              F
                                                                    pregion for end.i:
                                                                    %57 = add nuw nsw i64 %_local_id_y.0, 1
%exitcond14 = icmp eq i64 %57, 8
                                                                    br i1 %exitcond14, label %mm2 kernel1.exit, label
                                                                    ... %pregion for entry.pregion for init.i, !llvm.loop !21
                                                                                                              F
                                                                       mm2 kernel1.exit:
```

ret void

CFG for 'pocl kernel mm2 kernel1' function

%13: