```
%13:
                                                                                                                                                                                                                           %14 = \text{sext i} 32 \% 6 \text{ to i} 64
                                                                                                                                                                                                                            %15 = icmp slt i64 %14, 32
                                                                                                                                                                                                                            %16 = select i1 %15, i64 %14, i64 32
                                                                                                                                                                                                                            %17 = \text{sext i} 32 \% 3 \text{ to i} 64
                                                                                                                                                                                                                            %18 = icmp slt i64 %17, 8
                                                                                                                                                                                                                            %19 = select i1 %18, i64 %17, i64 8
                                                                                                                                                                                                                            %mul.i.i = shl i64 %10, 5
                                                                                                                                                                                                                            %mul3.i.i = shl i64 %11, 3
                                                                                                                                                                                                                            %cmp739.i = icmp sgt i32 %4, 0, !llvm.access.group !12
                                                                                                                                                                                                                            %wide.trip.count.i = zext i32 %4 to i64
                                                                                                                                                                                                                            %20 = icmp ugt i64 %16, 1
                                                                                                                                                                                                                            %umax = select i1 %20, i64 %16, i64 1
                                                                                                                                                                                                                            %21 = icmp ugt i64 %19, 1
                                                                                                                                                                                                                            %umax3 = select i1 %21, i64 %19, i64 1
                                                                                                                                                                                                                            %min.iters.check = icmp ult i64 %umax3, 8
                                                                                                                                                                                                                            br i1 %min.iters.check, label
                                                                                                                                                                                                                           ... %pregion for entry.pregion for init.i.preheader, label %vector.ph
                                                                                                                                                                                                                                                                                vector.ph:
                                                                                                                                                                                                                                                                                  %n.vec = and i64 %umax3, -8
                                                                                                                                                                                                                                                                                  %broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul3.i.i, i32 0
                                                                                                                                                                                                                                                                                  %broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
                                                                                                                                                                                                                                                                                  ... undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                                  %broadcast.splatinsert5 = insertelement <8 x i32> undef, i32 %6, i32 0
                                                                                                                                                                                                                                                                                  %broadcast.splat6 = shufflevector <8 x i32> %broadcast.splatinsert5, <8 x
                                                                                                                                                                                                                                                                                  ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                                  %broadcast.splatinsert7 = insertelement <8 x i32> undef, i32 %4, i32 0
                                                                                                                                                                                                                                                                                  %broadcast.splat8 = shufflevector <8 x i32> %broadcast.splatinsert7, <8 x
                                                                                                                                                                                                                                                                                  ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                                  %broadcast.splatinsert10 = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
                                                                                                                                                                                                                                                                                  %broadcast.splat11 = shufflevector <8 x i64> %broadcast.splatinsert10, <8 x
                                                                                                                                                                                                                                                                                  ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                                  %broadcast.splatinsert12 = insertelement <8 x float> undef, float %8, i32 0
                                                                                                                                                                                                                                                                                  %broadcast.splat13 = shufflevector <8 x float> %broadcast.splatinsert12, <8
                                                                                                                                                                                                                                                                                  ... x float> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                                  %broadcast.splatinsert21 = insertelement <8 x i64> undef, i64 %14, i32 0
                                                                                                                                                                                                                                                                                  %broadcast.splat22 = shufflevector <8 x i64> %broadcast.splatinsert21, <8 x
                                                                                                                                                                                                                                                                                  ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                                  %broadcast.splatinsert24 = insertelement <8 x i64> undef, i64
                                                                                                                                                                                                                                                                                 ... %wide.trip.count.i, i32 0
                                                                                                                                                                                                                                                                                  %broadcast.splat25 = shufflevector <8 x i64> %broadcast.splatinsert24, <8 x
                                                                                                                                                                                                                                                                                  ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                                  %broadcast.splatinsert28 = insertelement <8 x i64> undef, i64 %umax, i32 0
                                                                                                                                                                                                                                                                                  %broadcast.splat29 = shufflevector <8 x i64> %broadcast.splatinsert28, <8 x
                                                                                                                                                                                                                                                                                  ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                                  br label %vector.body
                                                                                                                                                                                                                                                                                                              vector.body:
                                                                                                                                                                                                                                                                                                                %index = phi i64 [0, %vector.ph], [%index.next, %pregion for end.i30]
                                                                                                                                                                                                                                                                                                                %vec.ind = phi < 8 \times i64 > [< i64 \ 0, i64 \ 1, i64 \ 2, i64 \ 3, i64 \ 4, i64 \ 5, i64 \ 6, i64 \ 6
                                                                                                                                                                                                                                                                                                                ... i64 7>, %vector.ph ], [ %vec.ind.next, %pregion_for_end.i30 ]
                                                                                                                                                                                                                                                                                                                %22 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
                                                                                                                                                                                                                                                                                                                %23 = trunc <8 x i64> %22 to <8 x i32>, !llvm.access.group !12
                                                                                                                                                                                                                                                                                                                \%24 = \text{mul nsw} < 8 \times \text{i}32 > \% \text{broadcast.splat6}, \%23, !llvm.access.group !12
                                                                                                                                                                                                                                                                                                                %25 = mul nsw <8 x i32> %broadcast.splat8, %23
                                                                                                                                                                                                                                                                                                                %26 = \text{sext} < 8 \times i32 > \%25 \text{ to} < 8 \times i64 > 3 \times i
                                                                                                                                                                                                                                                                                                                br label %pregion for entry.entry.i9
                                                                                                                                                                                                                                                                                                      pregion for entry.entry.i9:
                                                                                                                                                                                                                                                                                                       %vec.phi = phi <8 x i64> [ zeroinitializer, %vector.body ], [ %44,
                                                                                                                                                                                                                                                                                                       ... %if.end.r exit.i27 1
                                                                                                                                                                                                                                                                                                       %27 = add <8 x i64 > %vec.phi, %broadcast.splat11, !llvm.access.group !12
                                                                                                                                                                                                                                                                                                      %28 = trunc <8 x i64> %27 to <8 x i32>, !llvm.access.group !12 %29 = add nsw <8 x i32> %24, %28, !llvm.access.group !12
                                                                                                                                                                                                                                                                                                       %30 = \text{sext} < 8 \times i32 > %29 \text{ to} < 8 \times i64 >, !llvm.access.group !12
                                                                                                                                                                                                                                                                                                       %31 = \text{getelementptr inbounds float, float* } \%2, <8 \times i64 > \%30,
                                                                                                                                                                                                                                                                                                      ...!llvm.access.group!12
%wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                                                                                                                                                                                                      ... x float*> %31, i32 4, <8 x i1> <i1 true, i1 
                                                                                                                                                                                                                                                                                                      ... i1 true, i1 true, i1 true>, <8 x float> undef), !tbaa !15, !llvm.access.group
                                                                                                                                                                                                                                                                                                       %32 = fmul <8 x float> %wide.masked.gather, %broadcast.splat13,
                                                                                                                                                                                                                                                                                                       ...!llvm.access.group!12
                                                                                                                                                                                                                                                                                                       call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %32, <8 x float*>
                                                                                                                                                                                                                                                                                                      ... %31, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1 true,
                                                                                                                                                                                                                                                                                                      ... i1 true, i1 true>), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                                                                                                                                       br i1 %cmp739.i, label %for.body.lr.ph.i16, label %if.end.r exit.i27
                                                                                                                                                                                                                                              for.body.lr.ph.i16:
                                                                                                                                                                                                                                               %33 = shl <8 x i64> %27, <i64 32, i64 32, i64 32, i64 32, i64 32, i64 32, i64 32,
                                                                                                                                                                                                                                                .. i64 32, i64 32>, !llvm.access.group !12
                                                                                                                                                                                                                                               %34 = ashr exact < 8 \times i64 > %33, < i64 32, i64 32, i64 32, i64 32, i64 32, i64 32, i64
                                                                                                                                                                                                                                               ... 32, i64 32, i64 32>, !llvm.access.group !12
                                                                                                                                                                                                                                               br label %for.body.i17
                                                                                                                                                                                                                                for.body.i17:
                                                                                                                                                                                                                                 \text{%vec.phi18} = \text{phi} < 8 \times \text{i64} > [\text{%41}, \text{%for.body.i17}], [\text{zeroinitializer},]
                                                                                                                                                                                                                                ... %for.body.lr.ph.i16 ]
%vec.phi19 = phi <8 x float> [ %40, %for.body.i17 ], [ %32, ... %for.body.lr.ph.i16 ]
                                                                                                                                                                                                                                 %35 = add nsw <8 x i64> %vec.phi18, %26, !llvm.access.group !12 %36 = getelementptr inbounds float, float* %0, <8 x i64> %35,
                                                                                                                                                                                                                                 ...!llvm.access.group!12
                                                                                                                                                                                                                                 %wide.masked.gather20 = call <8 x float>
                                                                                                                                                                                                                                ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %36, i32 4, <8 x i1> <i1 true,
                                                                                                                                                                                                                               ... i1 true, i7 true, i1 true,
                                                                                                                                                                                                                                ... undef), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                                                                 %37 = mul nsw <8 x i64> %vec.phi18, %broadcast.splat22, !llvm.access.group
                                                                                                                                                                                                                                 %38 = add nsw <8 x i64> %37, %34, !llvm.access.group !12
                                                                                                                                                                                                                                 %39 = getelementptr inbounds float, float* %1, <8 x i64> %38,
                                                                                                                                                                                                                                 ...!llvm.access.group!12
                                                                                                                                                                                                                                 %wide.masked.gather23 = call <8 x float>
                                                                                                                                                                                                                                ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %39, i32 4, <8 x i1> <i1 true,
                                                                                                                                                                                                                               ... i1 true, i7 true, i1 true,
                                                                                                                                                                                                                                ... undef), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                                                                %40 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float>
                                                                                                                                                                                                                               ... %wide.masked.gather20, <8 x float> %wide.masked.gather23, <8 x float>
                                                                                                                                                                                                                                ... %vec.phi19), !llvm.access.group !12
                                                                                                                                                                                                                               call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %40, <8 x float*> ... %31, i32 4, <8 x i1> <i1 true, i1 true, i
                                                                                                                                                                                                                                ... i1 true, i1 true>), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                                                                 %41 = add nuw nsw <8 x i64> %vec.phi18, <i64 1, i64 1, i64 1, i64 1, i64 1,
                                                                                                                                                                                                                                 ... i64 1, i64 1>, !llvm.access.group !12
%42 = icmp eq <8 x i64> %41, %broadcast.splat25, !llvm.access.group !12
                                                                                                                                                                                                                                 %43 = \text{extractelement} < 8 \times i1 > %42, i32 0
                                                                                                                                                                                                                                 br i1 %43, label %if.end.r exit.i27.loopexit, label %for.body.i17
                                                                                                                                                                                                                                                                                                                   if.end.r exit.i27.loopexit:
                                                                                                                                                                                                                                                                                                                    br label %if.end.r exit.i27
                                                                                                                                                                                                                                                                                                                   if.end.r exit.i27:
                                                                                                                                                                                                                                                                                                                    %44 = add nuw <8 x i64> %vec.phi, <i64 1, i64 1, i64 1, i64 1, i64 1, i64 1,
                                                                                                                                                                                                                                                                                                                    ... i64 1, i64 1>
                                                                                                                                                                                                                                                                                                                    %45 = icmp eq <8 x i64> %44, %broadcast.splat29
                                                                                                                                                                                                                                                                                                                    %46 = \text{extractelement} < 8 \times i1 > %45, i32 0
                                                                                                                                                                                                                                                                                                                    br i1 %46, label %pregion for end.i30, label %pregion for entry.entry.i9
                                                                                                                                                                                                                                                                                       pregion for end.i30:
                                                                                                                                                                                                                                                                                        %index.next = add i64 %index, 8
                                                                                                                                                                                                                                                                                        %vec.ind.next = add <8 x i64> %vec.ind, <i64 8, i64 8, i64 8, i64 8, i64 8,
                                                                                                                                                                                                                                                                                        ... i64 8, i64 8, i64 8>
                                                                                                                                                                                                                                                                                       %47 = icmp eq i64 %index.next, %n.vec
br i1 %47, label %middle.block, label %vector.body, !llvm.loop !19
                                                                                                                                                                                                                                                                     middle.block:
                                                                                                                                                                                                                                                                      %cmp.n = icmp eq i64 %umax3, %n.vec
br i1 %cmp.n, label %mm2_kernel2.exit, label
                                                                                                                                                                                                                                                                      ... %pregion for entry.pregion for init.i.preheader
                                                                                                                                pregion_for_entry.pregion_for_init.i.preheader:
                                                                                                                                 br label %pregion for entry.pregion for init.i
                                                                                                              pregion_for_entry.pregion_for_init.i:
%_local_id_y.0 = phi i64 [ %59, %pregion_for_end.i ], [ %_local_id_y.0.ph,
... %pregion_for_entry.pregion_for_init.i.preheader ]
%add6.i.i = add i64 %_local_id_y.0, %mul3.i.i, !llvm.access.group !12
%conv2.i = trunc i64 %add6.i.i to i32, !llvm.access.group !12
%mul.i = mul nsw i32 %conv2.i, %6, !llvm.access.group !12
                                                                                                                  %mul9.i = mul nsw i32 %conv2.i, %4
                                                                                                                  %48 = sext i32 %mul9.i to i64
                                                                                                                  br label %pregion for entry.entry.i
                                                                            pregion_for_entry.entry.i:
    %_local_id_x.0 = phi i64 [ 0, %pregion_for_entry.pregion_for_init.i ], [
                                                                             ... \( \frac{1}{3} \) 58, \( \frac{1}{3} \) if.end.r exit.i \( \frac{1}{3} \)
                                                                             %add1.i.i = add i64 %_local_id_x.0, %mul.i.i, !llvm.access.group !12 %conv.i = trunc i64 %add1.i.i to i32, !llvm.access.group !12
                                                                              %add.i = add nsw i32 %mul.i, %conv.i, !llvm.access.group !12
                                                                              %idxprom.i = sext i32 %add.i to i64, !llvm.access.group !12
                                                                              %arrayidx.i = getelementptr inbounds float, float* %2, i64 %idxprom.i,
                                                                              ...!llvm.access.group!12
                                                                             %49 = load float, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
                                                                             ...!12
                                                                            %mul6.i = fmul float %49, %8, !llvm.access.group !12 store float %mul6.i, float* %arrayidx.i, align 4, !tbaa !15,
                                                                              ..!llvm.access.group!12
                                                                            br i1 %cmp739.i, label %for.body.lr.ph.i, label %if.end.r_exit.i,
                                                                            ...!llvm.access.group!12
                                        for.body.lr.ph.i:
                                        %sext.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
                                         %50 = ashr exact i64 %sext.i, 32, !llvm.access.group !12
                                         br label %for.body.i, !llvm.access.group !12
for.body.i:
 %indvars.iv.next.i2 = phi i64 [ %indvars.iv.next.i, %for.body.i ], [ 0,
 ... %for.body.lr.ph.i ]
 %51 = phi float [ %57, %for.body.i ], [ %mul6.i, %for.body.lr.ph.i ]
 %52 = add nsw i64 %indvars.iv.next.i2, %48, !llvm.access.group !12
 %arrayidx12.i = getelementptr inbounds float, float* %0, i64 %52,
...!llvm.access.group!12
%53 = load float, float* %arrayidx12.i, align 4, !tbaa!15,
 ...!llvm.access.group!12
%54 = mul nsw i64 %indvars.iv.next.i2, %14, !llvm.access.group !12 %55 = add nsw i64 %54, %50, !llvm.access.group !12
 %arrayidx16.i = getelementptr inbounds float, float* %1, i64 %55,
...!llvm.access.group!12
%56 = load float, float* %arrayidx16.i, align 4,!tbaa!15,
 ...!llvm.access.group!12
%57 = tail call float @llvm.fmuladd.f32(float %53, float %56, float %51) #5,
...!llvm.access.group!12
 store float %57, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
 %indvars.iv.next.i = add nuw nsw i64 %indvars.iv.next.i2, 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 %if.end.r exit.i.loopexit, label %for.body.i,
...!llvm.loop!22,!llvm.access.group!12
                                                                               if.end.r exit.i.loopexit:
                                                                                br label %if.end.r exit.i
                                                                                                                         if.end.r exit.i:
                                                                                                                         %58 = add nuw i64 %_local_id_x.0, 1
                                                                                                                         %exitcond.not = icmp eq i64 %58, %umax
                                                                                                                         br i1 %exitcond.not, label %pregion for end.i, label
                                                                                                                          ... %pregion for entry.entry.i, !llvm.loop !24
                                                                                                                                           pregion for end.i:
                                                                                                                                           .
%59 = add nuw i64 %_local_id_y.0, 1
                                                                                                                                           %exitcond4.not = icmp eq i64 %59, %umax3
br i1 %exitcond4.not, label %mm2_kernel2.exit.loopexit, label
                                                                                                                                           ... %pregion for entry.pregion for init.i, !llvm.loop !27
                                                                                                                                                                                      mm2 kernel2.exit.loopexit:
                                                                                                                                                                                        br label %mm2 kernel2.exit
```

mm2\_kernel2.exit:
 ret void

CFG for ' pocl kernel mm2 kernel2' function