```
%13:
                                                              %mul.i.i = shl i64 %10, 5
                                                              %mul3.i.i = shl i64 %11, 3
                                                              %cmp32.i = icmp sgt i32 %5, 0, !llvm.access.group !12
                                                              %14 = \text{sext i} 32 \% 4 \text{ to i} 64
                                                              %wide.trip.count.i = zext i32 %5 to i64
                                                              br label %pregion for entry.pregion for init.i
                                                  pregion for entry.pregion for init.i:
                                                  %add6.i.i = add nuw nsw 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, %4, !llvm.access.group !12
                                                  %mul4.i = mul nsw i32 %conv2.i, %5
                                                  %15 = \text{sext i} 32 \% \text{mul} 4.\text{i 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 ], [
                            ... %24, %fōr.end.r exit.i ]
                            %add1.i.i = add nuw nsw 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* %0, i64 %idxprom.i,
                            ...!llvm.access.group!12
                            store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !15,
                            ...!llvm.access.group!12
                            br i1 %cmp32.i, label %for.body.lr.ph.i, label %for.end.r exit.i,
                            ...!llvm.access.group!12
                                                                                       F
               for.body.lr.ph.i:
               %sext.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
               %16 = 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 ]
%17 = phi float [ %23, %for.body.i ], [ 0.000000e+00, %for.body.lr.ph.i ]
%18 = add nsw i64 %indvars.iv.next.i2, %15, !llvm.access.group !12
%arrayidx7.i = getelementptr inbounds float, float* %1, i64 %18,
...!llvm.access.group!12
%19 = load float, float* %arrayidx7.i, align 4, !tbaa !15,
...!llvm.access.group!12
%mul8.i = fmul float %19, %7, !llvm.access.group !12
%20 = mul nsw i64 %indvars.iv.next.i2, %14, !llvm.access.group !12
%21 = add nsw i64 %20, %16, !llvm.access.group !12
%arrayidx12.i = getelementptr inbounds float, float* %2, i64 %21,
...!llvm.access.group!12
%22 = load float, float* %arrayidx12.i, align 4, !tbaa !15,
...!llvm.access.group!12
%23 = call float @llvm.fmuladd.f32(float %mul8.i, float %22, float %17) #3,
...!llvm.access.group!12
store float %23, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
... !12
%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 %for.end.r exit.i.loopexit, label %for.body.i,
...!llvm.loop!19,!llvm.access.group!12
                                                          F
                              for.end.r exit.i.loopexit:
                               br label %for.end.r exit.i
                                                  for.end.r exit.i:
                                                  %24 = add nuw nsw i64 % local id x.0, 1
                                                  %exitcond.not = icmp eq \overline{164} %24, \overline{32}
                                                  br i1 %exitcond.not, label %pregion for end.i, label
                                                  ... %pregion for entry.entry.i, !llvm.loop \bar{1}21
                                                       pregion for end.i:
                                                       \%25 = add nuw nsw i64 % local id y.0, 1
                                                       \%exitcond3.not = icmp eq \overline{i}64 \% \overline{25}, 8
                                                       br i1 %exitcond3.not, label %mm2 kernel1.exit, label
                                                       ... %pregion for entry.pregion for init.i, !llvm.loop !24
                                                          mm2 kernel1.exit:
                                                          ret void
```

CFG for 'pocl kernel mm2 kernel1' function