```
%7:
                                                            %mul.i.i = shl i64 %4, 5
                                                            %add.i = add nsw i32 %1, 1, !llvm.access.group !12
                                                            %mul3.i.i = shl i64 %5, 3
                                                            %mul16.i = mul nsw i32 %2, %1
                                                            br label %pregion for entry.pregion for init.i
                                              pregion for entry.pregion for init.i:
                                               \%_{local\_id\_y.0} = phi i64 [0, \%7], [\%15, \%pregion_for_end.i]
                                               %add6.i.i = add nuw nsw i64 % local id y.0, %mul3.i.i, !llvm.access.group !12
                                               %8 = trunc i64 %add6.i.i to i32, !llvm.access.group !12
                                               %conv7.i = add i32 %add.i, %8, !llvm.access.group !12
                                               %cmp.i = icmp slt i32 %conv7.i, %2, !llvm.access.group !12
                                               %mul.i = mul nsw i32 %conv7.i, %2
                                               %add13.i = add nsw i32 %mul.i, %1
                                               %idxprom14.i = sext i32 %add13.i to i64
                                               %arrayidx15.i = getelementptr inbounds float, float* %0, i64 %idxprom14.i
                                               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 ], [
                        ... \(\bar{8}\)14, \(\bar{8}\)if.end.r exit.i ]
                         %add1.i.i = add nuw nsw i64 % local id x.0, %mul.i.i, !llvm.access.group !12
                         %9 = trunc i64 %add1.i.i to i32, !llvm.access.group !12
                         %conv2.i = add i32 %add.i, %9, !llvm.access.group !12
                         %cmp9.i = icmp slt i32 %conv2.i, %2, !llvm.access.group !12
                         %or.cond.i = and i1 %cmp.i, %cmp9.i, !llvm.access.group !12
                         br i1 %or.cond.i, label %if.then.i, label %if.end.r exit.i,
                        ...!llvm.access.group!12
                                                                                    F
if.then.i:
%add11.i = add nsw i32 %conv2.i, %mul.i, !llvm.access.group !12
%idxprom.i = sext i32 %add11.i to i64, !llvm.access.group !12
%arrayidx.i = getelementptr inbounds float, float* %0, i64 %idxprom.i,
...!llvm.access.group!12
%10 = load float, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
...!12
%11 = load float, float* %arrayidx15.i, align 4, !tbaa !15,
...!llvm.access.group!12
%add17.i = add nsw i32 %conv2.i, %mul16.i, !llvm.access.group !12
%idxprom18.i = sext i32 %add17.i to i64, !llvm.access.group !12
%arrayidx19.i = getelementptr inbounds float, float* %0, i64 %idxprom18.i,
...!llvm.access.group!12
%12 = load float, float* %arrayidx19.i, align 4, !tbaa !15,
...!llvm.access.group!12
%neg.i = fneg float %11, !llvm.access.group !12
%13 = tail call float @llvm.fmuladd.f32(float %neg.i, float %12, float %10)
... #3, !llvm.access.group !12
store float %13, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
br label %if.end.r_exit.i, !llvm.access.group !12
                                               if.end.r exit.i:
                                               %14 = add nuw nsw i64 \% local id x.0, 1
                                               %exitcond.not = icmp eq i\overline{64} %14, \overline{32}
                                               br i1 %exitcond.not, label %pregion for end.i, label
                                               ... %pregion for entry.entry.i, !llvm.loop 19
                                                 pregion for end.i:
                                                  %15 = add nuw nsw i64 % local id y.0, 1
                                                  \%exitcond1.not = icmp eq \overline{i}64 \%\overline{15}, 8
                                                  br i1 %exitcond1.not, label %lu kernel2.exit, label
                                                 ... %pregion for entry.pregion for init.i, !llvm.loop!22
                                                      lu kernel2.exit:
                                                       ret void
                                              CFG for 'pocl kernel lu kernel2' function
```