```
%7:
                                                   %8 = shl i64 %4, 8
                                                    %add.i = add nsw i32 %1, 1
                                                   %mul.i = mul nsw i32 %2, %1
                                                    %add6.i = add nsw i32 %mul.i, %1
                                                    %idxprom7.i = sext i32 %add6.i to i64
                                                    %arrayidx8.i = getelementptr inbounds float, float* %0, i64 %idxprom7.i
                                                   br label %pregion for entry.entry.i
                                                          pregion for entry.entry.i:
                                                          __local_id_x.0 = phi i64 [ 0, %7 ], [ %18, %if.end.r_exit.i.1 ]
                                                          \%\overline{9} = \overline{add} nuw nsw i64 % local id x.0, %8
                                                           %10 = \text{trunc } i64 \%9 \text{ to } i3\overline{2}
                                                          %conv2.i = add i32 %add.i, %10
                                                           %cmp.i = icmp slt i32 %conv2.i, %2
                                                          br i1 %cmp.i, label %if.then.i, label %if.end.r exit.i
                    if.then.i:
                     %add4.i = add nsw i32 %conv2.i, %mul.i
                     %idxprom.i = sext i32 %add4.i to i64
                     %arrayidx.i = getelementptr inbounds float, float* %0, i64 %idxprom.i
                     %11 = load float, float* %arrayidx.i, align 4, !tbaa !12
                     %12 = load float, float* %arrayidx8.i, align 4, !tbaa !12
                     %div.i = fdiv float %11, %12, !fpmath !16
                     store float %div.i, float* %arrayidx.i, align 4, !tbaa !12,
                    ...!llvm.access.group!17
                     br label %if.end.r exit.i
                               if.end.r exit.i:
                                %13 = \text{ or } i64 \% \text{ local id } x.0, 1
                                %14 = add nuw nsw i64 %13, %8
                                %15 = \text{trunc } i64 \%14 \text{ to } i32
                                %conv2.i.1 = add i32 %add.i, %15
                                %cmp.i.1 = icmp slt i32 %conv2.i.1, %2
                                br i1 %cmp.i.1, label %if.then.i.1, label %if.end.r exit.i.1
                                                                            F
%add4.i.1 = add nsw i32 %conv2.i.1, %mul.i
%idxprom.i.1 = sext i32 %add4.i.1 to i64
%arrayidx.i.1 = getelementptr inbounds float, float* %0, i64 %idxprom.i.1
%16 = load float, float* %arrayidx.i.1, align 4, !tbaa !12
%17 = load float, float* %arrayidx8.i, align 4, !tbaa !12
%div.i.1 = fdiv float %16, %17, !fpmath !16
store float %div.i.1, float* %arrayidx.i.1, align 4, !tbaa !12,
...!llvm.access.group!17
br label %if.end.r_exit.i.1
                                         if.end.r exit.i.1:
                                         %18 = add nuw nsw i64 \% local id x.0, 2
                                         \%exitcond.1 = icmp eq i64 \%18, 256
                                         br i1 %exitcond.1, label %lu kernel1.exit, label %pregion for entry.entry.i,
                                         ...!llvm.loop!19
                                                           Т
                                                                                                    F
                                                   lu kernel1.exit:
                                                    ret void
```

CFG for 'pocl kernel lu kernel1' function

if.then.i.1: