```
%7:
                                                                                   %mul3.i.i = shl i64 %5, 3
                                                                                   %mul.i.i = shl i64 %4, 5
                                                                                   %sub.i = add nsw i32 %2, -1
                                                                                   br label %pregion for entry.pregion for init.i
                                                                  pregion for entry.pregion for init.i:
                                                                  \c^{1}%_local_id_y.0 = phi i64 [0, \c^{1}7], [%13, %pregion_for_end.i]
                                                                  %add6.i.i = add nuw nsw i64 % local id y.0, %mul3.i.i, !llvm.access.group !12
                                                                   %conv.i = trunc i64 %add6.i.i to i32, !llvm.access.group !12
                                                                  %cmp.i = icmp sgt i32 %conv.i, 0, !llvm.access.group !12
                                                                   %cmp4.i = icmp sgt i32 %sub.i, %conv.i
                                                                   %mul.i = mul nsw i32 %conv.i, %2
                                                                   %add25.i = add nuw nsw i32 %conv.i, 1
                                                                   %mul26.i = mul nsw i32 %add25.i, %2
                                                                   %sub31.i = add nsw i32 %conv.i, -1
                                                                   %mul32.i = mul nsw i32 %sub31.i, %2
                                                                   br label %pregion for entry.entry.i
                                            pregion for entry.entry.i:
                                             % local id \bar{x}.0 = phi i64 [ 0, %pregion for entry pregion for init.i ], [
                                            ... \(\overline{\pi}\)14, \(\overline{\pi}\)if.end.i \(\overline{\pi}\)
                                            %add1.i.i = add nuw nsw i64 % local id x.0, %mul.i.i, !llvm.access.group !12
                                             %conv2.i = trunc i64 %add1.i.i to i32, !llvm.access.group !12
                                             br i1 %cmp.i, label %land.lhs.true.i, label %if.end.i, !llvm.access.group !12
                     land.lhs.true.i:
                      %cmp7.i = icmp sqt i32 %conv2.i, 0, !llvm.access.group !12
                      %or.cond.i = and i1 %cmp4.i, %cmp7.i, !llvm.access.group !12
                      %cmp11.i = icmp sgt i32 %sub.i, %conv2.i, !llvm.access.group !12
                      %or.cond69.i = and i1 %cmp11.i, %or.cond.i, !llvm.access.group !12
                      br i1 %or.cond69.i, label %if.then.i, label %if.end.i, !llvm.access.group !12
if.then.i:
%add.i = add i32 %mul.i, %conv2.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
%8 = load float, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
%add15.i = add i32 %add.i, -1, !llvm.access.group !12
%idxprom16.i = sext i32 %add15.i to i64, !llvm.access.group !12
%arrayidx17.i = getelementptr inbounds float, float* %0, i64 %idxprom16.i,
...!llvm.access.group!12
%9 = load float, float* %arrayidx17.i, align 4, !tbaa !15,
...!llvm.access.group!12
%add18.i = fadd float %8, %9, !llvm.access.group !12
%add21.i = add i32 %add.i, 1, !llvm.access.group !12
%idxprom22.i = sext i32 %add21.i to i64, !llvm.access.group !12
%arravidx23.i = getelementptr inbounds float, float* %0, i64 %idxprom22.i,
...!llvm.access.group!12
%10 = load float, float* %arrayidx23.i, align 4, !tbaa !15,
...!llvm.access.group!12
%add24.i = fadd float %add18.i, %10, !llvm.access.group !12
%add27.i = add nsw i32 %mul26.i, %conv2.i, !llvm.access.group !12
%idxprom28.i = sext i32 %add27.i to i64, !llvm.access.group !12
%arrayidx29.i = getelementptr inbounds float, float* %0, i64 %idxprom28.i,
...!llvm.access.group!12
%11 = load float, float* %arrayidx29.i, align 4, !tbaa !15,
...!llvm.access.group!12
%add30.i = fadd float %add24.i, %11, !llvm.access.group !12
%add33.i = add nsw i32 %mul32.i, %conv2.i, !llvm.access.group !12
%idxprom34.i = sext i32 %add33.i to i64, !llvm.access.group !12
%arrayidx35.i = getelementptr inbounds float, float* %0, i64 %idxprom34.i,
...!llvm.access.group!12
%12 = load float, float* %arrayidx35.i, align 4, !tbaa !15,
...!llvm.access.group!12
%add36.i = fadd float %add30.i, %12, !llvm.access.group !12
%mul37.i = fmul float %add36.i, 0x3FC99999A0000000, !llvm.access.group !12
%arrayidx41.i = getelementptr inbounds float, float* %1, i64 %idxprom.i,
...!llvm.access.group!12
store float %mul37.i, float* %arrayidx41.i, align 4, !tbaa !15,
...!llvm.access.group!12
br label %if.end.i, !llvm.access.group !12
                                                              if.end.i:
                                                               %14 = add nuw nsw i64 % local id x.0, 1
                                                               \%exitcond.not = icmp eq i64 %14, 32
                                                               br i1 %exitcond.not, label %pregion for end.i, label
                                                               ... %pregion for entry.entry.i, !llvm.loop !22
                                                              pregion for end.i:
                                                               ^{\circ}%13 = add nuw nsw i64 % local id y.0, 1
                                                               %exitcond1.not = icmp eq \overline{164} %13, 8
                                                               br i1 %exitcond1.not, label %runJacobi2D kernel1.exit, label
                                                               ... %pregion for entry.pregion for init.i, !llvm.loop !19
                                                                            Τ
                                                                                                             F
                                                                runJacobi2D kernel1.exit:
                                                                ret void
```

CFG for 'pocl kernel runJacobi2D kernel1' function