```
%8:
                                        %mul.i.i = shl i64 %5, 5
                                        %cmp218.i = icmp sqt i32 %3, 0, !llvm.access.group !12
                                        %wide.trip.count.i = zext i32 %3 to i64
                                        br label %pregion for entry.entry.i
                             pregion for entry.entry.i:
                             % local id x.0 = phi i64 [0, \%8], [\%16, \%if.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
                             %cmp.i = icmp slt i32 %conv.i, %3, !llvm.access.group !12
                             %or.cond.i = and i1 %cmp218.i, %cmp.i, !llvm.access.group !12
                             br i1 %or.cond.i, label %for.body.lr.ph.i, label %if.end.r exit.i,
                             ...!llvm.access.group!12
                                                                                         F
 for.body.lr.ph.i:
  %mul.i = mul nsw i32 %conv.i, %3, !llvm.access.group !12
  %sext.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
  %idxprom7.i = ashr exact i64 %sext.i, 32, !llvm.access.group !12
  %arrayidx8.i = getelementptr inbounds float, float* %1, i64 %idxprom7.i,
 ...!llvm.access.group!12
  %9 = sext i32 %mul.i to i64, !llvm.access.group !12
  %.pre.i = load float, float* %arrayidx8.i, align 4, !tbaa !14,
 ...!llvm.access.group!12
  br label %for.bodv.i, !llvm.access.group !12
for.body.i:
%indvars.iv.next.i2 = phi i64 [ %indvars.iv.next.i, %for.bodv.i ], [ 0,
... %for.body.lr.ph.i 1
%10 = phi float [ %15, %for.body.i ], [ %.pre.i, %for.body.lr.ph.i ]
%11 = add nsw i64 %indvars.iv.next.i2, %9, !llvm.access.group !12
%arrayidx.i = getelementptr inbounds float, float* %0, i64 %11,
...!llvm.access.group!12
%12 = load float, float* %arrayidx.i, align 4, !tbaa !14, !llvm.access.group
... !12
%arrayidx5.i = getelementptr inbounds float, float* %2, i64
... %indvars.iv.next.i2, !llvm.access.group !12
%13 = load float, float* %arrayidx5.i, align 4, !tbaa !14,
...!llvm.access.group!12
%14 = fmul float %12, %13, !llvm.access.group !12
%15 = fadd float %10, %14, !llvm.access.group !12
store float %15, float* %arrayidx8.i, align 4, !tbaa !14, !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 %if.end.r exit.i.loopexit, label %for.body.i,
...!llvm.loop!18.!llvm.access.group!12
                              if.end.r exit.i.loopexit:
                              br label %if.end.r exit.i
                                            if.end.r exit.i:
                                            %16 = add nuw nsw i64 \% local id x.0, 1
                                            %exitcond.not = icmp eq i\overline{64} %1\overline{6}. \overline{32}
                                            br i1 %exitcond.not, label %mvt kernel1.exit, label
                                            ... %pregion for entry.entry.i, !llvm.loop !20
                                               mvt kernel1.exit:
                                               ret void
                                CFG for 'pocl kernel mvt kernel1' function
```