```
%9:
                                          %mul.i.i = shl i64 %6, 5
                                          %cmp217.i = icmp sqt i32 %3, 0, !llvm.access.group !12
                                          %10 = \text{sext i} 32 \% 4 \text{ to i} 64
                                          %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, %9 ], [ %17, %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, %4, !llvm.access.group !12
                                %or.cond.i = and i1 %cmp217.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
                                                  Τ
                                                                                            F
   for.body.lr.ph.i:
   %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
   %.pre.i = load float, float* %arrayidx8.i, align 4, !tbaa !14,
   ...!llvm.access.group!12
   br label %for.body.i, !llvm.access.group !12
for.bodv.i:
%indvars.iv.next.i3 = phi i64 [ %indvars.iv.next.i, %for.bodv.i ], [ 0,
... %for.body.lr.ph.i l
%11 = phi float [ %16, %for.body.i ], [ %.pre.i, %for.body.lr.ph.i ]
%12 = mul nsw i64 %indvars.iv.next.i3, %10, !llvm.access.group !12
%13 = add nsw i64 %12, %idxprom7.i, !llvm.access.group !12
%arrayidx.i = getelementptr inbounds float, float* %0, i64 %13,
...!llvm.access.group!12
%14 = load float, float* %arrayidx.i, align 4, !tbaa !14, !llvm.access.group
... !12
%arrayidx5.i = getelementptr inbounds float, float* %2, i64
... %indvars.iv.next.i3, !llvm.access.group !12
%15 = load float, float* %arrayidx5.i, align 4, !tbaa !14,
...!llvm.access.group!12
%16 = tail call float @llvm.fmuladd.f32(float %14, float %15, float %11) #3,
...!llvm.access.group!12
store float %16, float* %arrayidx8.i, align 4, !tbaa !14, !llvm.access.group
...!12
%indvars.iv.next.i = add nuw nsw i64 %indvars.iv.next.i3, 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:
                                               %17 = add nuw nsw i64 \% local id x.0, 1
                                               %exitcond.not = icmp eq \overline{164} %17, \overline{32}
                                              br i1 %exitcond.not, label %atax kernel2.exit, label
                                              ... %pregion for entry.entry.i, !llvm.loop !20
                                                                                      F
                                                          Т
                                                 atax kernel2.exit:
                                                  ret void
                                 CFG for 'pocl kernel atax kernel2' function
```