```
%9:
                                         %mul.i.i = shl i64 %6, 5
                                         %cmp14.i = icmp sgt i32 %4, 0, !llvm.access.group !12
                                         %wide.trip.count.i = zext i32 %4 to i64
                                         br label %pregion for entry.entry.i
                             pregion for entry.entry.i:
                             % local id x.0 = phi i64 [0, \%9], [\%16, \%for.end.r exit.i]
                             %add1.i.i = add nuw nsw i64 % local id x.0, %mul.i.i, !llvm.access.group !12
                             br i1 %cmp14.i, label %for.body.lr.ph.i, label %for.end.r exit.i,
                             ...!llvm.access.group!12
                                                                                         F
 for.body.lr.ph.i:
  %conv.i = trunc i64 %add1.i.i to i32, !llvm.access.group !12
  %mul.i = mul nsw i32 %conv.i, %4, !llvm.access.group !12
  %sext.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
  %idxprom5.i = ashr exact i64 %sext.i, 32, !llvm.access.group !12
  %arrayidx6.i = getelementptr inbounds float, float* %2, i64 %idxprom5.i,
 ...!llvm.access.group!12
  %10 = sext i32 %mul.i to i64, !llvm.access.group !12
  %.pre.i = load float, float* %arrayidx6.i, align 4, !tbaa !14,
 ...!llvm.access.group!12
 br label %for.body.i, !llvm.access.group !12
for.bodv.i:
%indvars.iv.next.i2 = phi i64 [ %indvars.iv.next.i, %for.bodv.i ], [ 0,
... %for.body.lr.ph.i ]
%11 = phi float [ %15, %for.body.i ], [ %.pre.i, %for.body.lr.ph.i ]
%12 = add nsw i64 %indvars.iv.next.i2, %10, !llvm.access.group !12
%arrayidx.i = getelementptr inbounds float, float* %0, i64 %12,
...!llvm.access.group!12
%13 = load float, float* %arrayidx.i, align 4, !tbaa !14, !llvm.access.group
...!12
%arrayidx3.i = getelementptr inbounds float, float* %1, i64
... %indvars.iv.next.i2, !llvm.access.group !12
%14 = load float, float* %arrayidx3.i, align 4, !tbaa !14,
...!llvm.access.group!12
%15 = call float @llvm.fmuladd.f32(float %13, float %14, float %11) #3,
...!llvm.access.group!12
store float %15, float* %arrayidx6.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 %for.end.r exit.i.loopexit, label %for.body.i,
...!llvm.loop!18,!llvm.access.group!12
                 Τ
                                                         F
                             for.end.r exit.i.loopexit:
                             br label %for.end.r exit.i
                                            for.end.r exit.i:
                                            %16 = add nuw nsw i64 \% local id x.0, 1
                                            %exitcond.not = icmp slt i\overline{6}4 %\overline{16}, %3
                                            br i1 %exitcond.not, label %atax kernel1.exit, label
                                            ... %pregion for entry.entry.i, !llvm.loop !20
                                                       Т
                                                                                    F
                                               atax kernel1.exit:
                                               ret void
```

CFG for 'pocl kernel atax kernel1' function