```
%8:
                                                                                      %mul.i.i = shl i64 %5, 8
                                                                                      %sub.i = add nsw i32 %2, -1, !llvm.access.group !12
                                                                                      %cmp662.i = icmp sgt i32 %3, 0
                                                                                      \%9 = \text{sext i} 32 \%2 \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, %8 ], [ %24, %if.end.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 sgt i32 %sub.i, %conv.i, !llvm.access.group !12
                                                                          br i1 %cmp.i, label %if.then.i, label %if.end.i, !llvm.access.group !12
                                                  if.then.i:
                                                   %mul.i = mul nsw i32 %conv.i, %2, !llvm.access.group !12
                                                   %add.i = add nsw i32 %mul.i, %conv.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
                                                  store float 1.000000e+00, float* %arrayidx.i, align 4, !tbaa !14,
                                                  ...!llvm.access.group!12
                                                  %j2.064.i = add nsw i32 %conv.i, 1, !llvm.access.group !12
                                                  %cmp365.i = icmp slt i32 %j2.064.i, %2, !llvm.access.group !12
                                                   br i1 %cmp365.i, label %for.cond5.preheader.lr.ph.i, label %if.end.i,
                                                  ...!llvm.access.group!12
                                        for.cond5.preheader.lr.ph.i:
                                         %sext.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
                                         %10 = ashr exact i64 %sext.i, 32, !llvm.access.group !12
                                         %11 = add nsw i64 %10, 1, !llvm.access.group !12
                                         %12 = sext i32 %mul.i to i64, !llvm.access.group !12
                                         br label %for.cond5.preheader.i, !llvm.access.group !12
                                for.cond5.preheader.i:
                                %indvars.iv.next71.i5 = phi i64 [ %indvars.iv.next71.i, %for.end.i ], [ %11,
                                 ... %for.cond5.preheader.lr.ph.i ]
                                %13 = add nsw i64 %indvars.iv.next71.i5, %12, !llvm.access.group !12
                                 %arrayidx21.i = getelementptr inbounds float, float* %0, i64 %13,
                                 ...!llvm.access.group!12
                                %.pre.i = load float, float* %arrayidx21.i, align 4, !tbaa !14,
                                 ...!llvm.access.group!12
                                br i1 %cmp662.i, label %for.body8.i.preheader, label %for.end.i,
                                ...!llvm.access.group!12
                                    for.body8.i.preheader:
                                     br label %for.body8.i
for.body8.i:
%indvars.iv.next.i3 = phi i64 [ %indvars.iv.next.i, %for.body8.i ], [ 0,
... %for.body8.i.preheader ]
%14 = phi float [ %20, %for.body8.i ], [ %.pre.i, %for.body8.i.preheader ]
%15 = mul nsw i64 %indvars.iv.next.i3, %9, !llvm.access.group !12
%16 = add nsw i64 %15, %10, !llvm.access.group !12
%arrayidx12.i = getelementptr inbounds float, float* %1, i64 %16,
...!llvm.access.group!12
%17 = load float, float* %arrayidx12.i, align 4, !tbaa !14,
...!llvm.access.group!12
%18 = add nsw i64 %15, %indvars.iv.next71.i5, !llvm.access.group !12
%arrayidx16.i = getelementptr inbounds float, float* %1, i64 %18,
...!llvm.access.group!12
%19 = load float, float* %arrayidx16.i, align 4, !tbaa !14,
...!llvm.access.group!12
%20 = tail call float @llvm.fmuladd.f32(float %17, float %19, float %14) #3,
...!llvm.access.group!12
store float %20, float* %arrayidx21.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 %for.end.i.loopexit, label %for.body8.i,
...!llvm.loop!18,!llvm.access.group!12
                  for.end.i.loopexit:
                   %.lcssa = phi float [ %20, %for.body8.i ]
                   br label %for.end.i
                            for.end.i:
                            %21 = phi float [ %.pre.i, %for.cond5.preheader.i ], [ %.lcssa,
                            ... %for.end.i.loopexit ]
                            %22 = mul nsw i64 %indvars.iv.next71.i5, %9, !llvm.access.group !12
                            %23 = add nsw i64 %22, %10, !llvm.access.group !12
                             %arrayidx29.i = getelementptr inbounds float, float* %0, i64 %23,
                            ...!llvm.access.group!12
                            store float %21, float* %arrayidx29.i, align 4, !tbaa !14,
                            ...!llvm.access.group!12
                             %indvars.iv.next71.i = add nsw i64 %indvars.iv.next71.i5, 1,
                             ...!llvm.access.group!12
                             %lftr.wideiv.i = trunc i64 %indvars.iv.next71.i to i32, !llvm.access.group
                            ...!12
                             %exitcond76.not.i = icmp eq i32 %lftr.wideiv.i, %2, !llvm.access.group !12
                            br i1 %exitcond76.not.i, label %if.end.i.loopexit, label
                            ... %for.cond5.preheader.i, !llvm.loop !20, !llvm.access.group !12
                                              Τ
                                                                                     F
                                                               if.end.i.loopexit:
                                                                br label %if.end.i
                                                                                        if.end.i:
                                                                                         %24 = add nuw nsw i64 % local id x.0, 1
                                                                                         \%exitcond.not = icmp eq i64 \%24, 256
                                                                                         br i1 %exitcond.not, label %corr kernel.exit, label
                                                                                        ... %pregion for entry.entry.i, !llvm.loop !21
                                                                                            corr kernel.exit:
                                                                                            ret void
```