```
%15 = \text{shl } i64 \%11, 3
                                                                                   %cmp739.i = icmp sgt i32 %4, 0
                                                                                   %16 = sext i32 %6 to i64
                                                                                   %wide.trip.count.i = zext i32 %4 to i64
                                                                                   %17 = add nsw i64 %wide.trip.count.i, -1
                                                                                   %xtraiter.i = and i64 %wide.trip.count.i, 3
                                                                                   %18 = icmp ult i64 %17, 3
                                                                                   %unroll iter.i = sub nuw nsw i64 %wide.trip.count.i, %xtraiter.i
                                                                                   %lcmp.mod.i = icmp eq i64 %xtraiter.i, 0
                                                                                   br label %pregion for entry.pregion for init.i
                                                                                   pregion for entry.pregion for init.i:
                                                                                    %_local_id_y.0 = phi i64 [ 0, %13 ], [ %58, %pregion_for_end.i ] %19 = add nuw nsw i64 %_local_id_y.0, %15
                                                                                    %conv2.i = trunc i64 %19 to i32
                                                                                    %cmp.i = icmp slt i32 %conv2.i, %3
%mul.i = mul nsw i32 %conv2.i, %6
                                                                                    %mul9.i = mul nsw i32 %conv2.i, %4
                                                                                    %20 = sext i32 %mul9.i to i64
                                                                                    br label %pregion for entry.entry.i
                                                            pregion for entry.entry.i:
                                                            %21 = add nuw nsw i64 %_local_id_x.0, %14
                                                            %conv.i = trunc i64 %21 to i32
                                                            %cmp4.i = icmp slt i32 %conv.i, %6
                                                            %or.cond.i = and i1 %cmp.i, %cmp4.i
                                                            br i1 %or.cond.i, label %if.then.i, label %if.end.i
                                                                                                                  F
                               if.then.i:
                               %add.i = add nsw i32 %mul.i, %conv.i
                               %idxprom.i = sext i32 %add.i to i64
                               %arrayidx.i = getelementptr inbounds float, float* %2, i64 %idxprom.i %22 = load float, float* %arrayidx.i, align 4, !tbaa !12
                               %mul6.i = fmul float %22, %8
store float %mul6.i, float* %arrayidx.i, align 4, !tbaa !12,
                               ...!llvm.access.group!16
                               br i1 %cmp739.i, label %for.body.lr.ph.i, label %if.end.i
                                                                                       F
for.body.lr.ph.i:
%sext.i = shl i64 %21, 32
%23 = ashr exact i64 %sext.i, 32
br i1 %18, label %if.end.loopexit.unr-lcssa.i, label %for.body.i.preheader
                 Τ
                                              for.bodv.i.preheader:
                                               br label %for.body.i
                 for.body.i:
                  %niter.nsub.3.i7 = phi i64 [ %niter.nsub.3.i, %for.body.i ], [
                  ... %unroll iter.i, %for.body.i.preheader ]
                  %indvars.īv.next.3.i4 = phi i64 [ %indvars.iv.next.3.i, %for.body.i ], [ 0,
                  ... %for.body.i.preheader ]
                  %24 = phi float [ %48, %for.body.i ], [ %mul6.i, %for.body.i.preheader ]
                  %25 = add nsw i64 %indvars.iv.next.3.i4, %20
                  %arrayidx12.i = getelementptr inbounds float, float* %0, i64 %25
                  %26 = load float, float* %arrayidx12.i, align 4, !tbaa !12
                  %27 = mul nsw i64 %indvars.iv.next.3.i4, %16
                  %28 = add nsw i64 %27, %23
                  %arrayidx16.i = getelementptr inbounds float, float* %1, i64 %28
                  %29 = load float, float* %arrayidx16.i, align 4, !tbaa !12
                  %30 = tail call float @llvm.fmuladd.f32(float %26, float %29, float %24) #2
                  store float %30, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                  %indvars.iv.next.i = or i64 %indvars.iv.next.3.i4, 1
                  %31 = add nsw i64 %indvars.iv.next.i, %20
                  %arrayidx12.1.i = getelementptr inbounds float, float* %0, i64 %31
                  %32 = load float, float* %arrayidx12.1.i, align 4, !tbaa !12
                  %33 = mul nsw i64 %indvars.iv.next.i, %16
                  %34 = add nsw i64 %33, %23
                  %arrayidx16.1.i = getelementptr inbounds float, float* %1, i64 %34
                  %35 = load float, float* %arrayidx16.1.i, align 4, !tbaa !12
                  %36 = tail call float @llvm.fmuladd.f32(float %32, float %35, float %30) #2
                  store float %36, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                  %indvars.iv.next.1.i = or i64 %indvars.iv.next.3.i4, 2
                  %37 = add nsw i64 %indvars.iv.next.1.i, %20
                  %arrayidx12.2.i = getelementptr inbounds float, float* %0, i64 %37
                  %38 = load float, float* %arrayidx12.2.i, align 4, !tbaa !12
                  %39 = mul nsw i64 %indvars.iv.next.1.i, %16
                  %40 = add nsw i64 %39, %23
                  %arrayidx16.2.i = getelementptr inbounds float, float* %1, i64 %40
                  %41 = load float, float* %arrayidx16.2.i, align 4, !tbaa !12
                  %42 = tail call float @llvm.fmuladd.f32(float %38, float %41, float %36) #2
                  store float %42, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                  ... !16
                  %indvars.iv.next.2.i = or i64 %indvars.iv.next.3.i4, 3
                  %43 = add nsw i64 %indvars.iv.next.2.i, %20
                  %arrayidx12.3.i = getelementptr inbounds float, float* %0, i64 %43
                  %44 = load float, float* %arrayidx12.3.i, align 4, !tbaa !12
                  %45 = mul nsw i64 %indvars.iv.next.2.i, %16
                  %46 = add nsw i64 %45, %23
                  %arrayidx16.3.i = getelementptr inbounds float, float* %1, i64 %46
                  %47 = load float, float* %arrayidx16.3.i, align 4, !tbaa !12
                  %48 = tail call float @llvm.fmuladd.f32(float %44, float %47, float %42) #2
                  store float %48, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                  ... !16
                  %indvars.iv.next.3.i = add nuw nsw i64 %indvars.iv.next.3.i4, 4
                  %niter.nsub.3.i = add i64 %niter.nsub.3.i7, -4
                  %niter.ncmp.3.i = icmp eq i64 %niter.nsub.3.i, 0
                  br i1 %niter.ncmp.3.i, label %if.end.loopexit.unr-lcssa.i.loopexit, label
                  ... %for.body.i
        if.end.loopexit.unr-lcssa.i.loopexit:
         %.lcssa = phi float [ %48, %for.body.i ]
         %indvars.iv.next.3.i.lcssa = phi i64 [ %indvars.iv.next.3.i, %for.body.i ]
         br label %if.end.loopexit.unr-lcssa.i
if.end.loopexit.unr-lcssa.i:
%49 = phi float [ %mul6.i, %for.body.lr.ph.i ], [ %.lcssa,
... %if.end.loopexit.unr-lcssa.i.loopexit ]
%50 = phi i64 [ 0, %for.body.lr.ph.i ], [ %indvars.iv.next.3.i.lcssa,
... %if.end.loopexit.unr-lcssa.i.loopexit ]
br i1 %lcmp.mod.i, label %if.end.i, label %for.body.epil.i.preheader
                                          for.body.epil.i.preheader:
                                          br label %for.body.epil.i
                 for.body.epil.i:
                  %epil.iter.sub.i13 = phi i64 [ %epil.iter.sub.i, %for.body.epil.i ], [
                  ... %xtraiter.i, %for.body.epil.i.preheader ]
                  %indvars.iv.next.epil.i11 = phi i64 [ %indvars.iv.next.epil.i,
                 ... %for.body.epil.i ], [ %50, %for.body.epil.i.preheader ] %51 = phi float [ %57, %for.body.epil.i ], [ %49, %for.body.epil.i.preheader
                  %52 = add nsw i64 %indvars.iv.next.epil.i11, %20
                  %arrayidx12.epil.i = getelementptr inbounds float, float* %0, i64 %52
                  %53 = load float, float* %arrayidx12.epil.i, align 4, !tbaa !12
                  %54 = mul nsw i64 %indvars.iv.next.epil.i11, %16
                  %55 = add nsw i64 %54, %23
                  %arrayidx16.epil.i = getelementptr inbounds float, float* %1, i64 %55 %56 = load float, float* %arrayidx16.epil.i, align 4, !tbaa !12
                  %57 = tail call float @llvm.fmuladd.f32(float %53, float %56, float %51) #2
                  store float %57, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                  ... !16
                  %indvars.iv.next.epil.i = add nuw nsw i64 %indvars.iv.next.epil.i11, 1
                  %epil.iter.sub.i = add nsw i64 %epil.iter.sub.i13, -1
                  %epil.iter.cmp.i = icmp eq i64 %epil.iter.sub.i, 0
                  br i1 %epil.iter.cmp.i, label %if.end.i.loopexit, label %for.body.epil.i,
                  ...!llvm.loop!19
                                                                             F
                                                       if.end.i.loopexit:
                                                       br label %if.end.i
                                                    if.end.i:
                                                    %59 = add nuw nsw i64 % local id x.0, 1
                                                    \%exitcond = icmp eq i64 \%59, 3\overline{2}
                                                    br i1 %exitcond, label %pregion for end.i, label %pregion for entry.entry.i,
                                                    ...!llvm.loop!23
                                                                                                                F
                                                                    pregion for end.i:
                                                                     %58 = add nuw nsw i64 % local id y.0, 1
                                                                     \%exitcond14 = icmp eq i6\overline{4} %58, 8
                                                                     br i1 %exitcond14, label %mm2 kernel2.exit, label
                                                                     ... %pregion for entry pregion for init.i, !llvm.loop !21
                                                                                                               F
                                                                        mm2 kernel2.exit:
```

%13:

%14 = shl i64 %10, 5