```
%9:
                                                               %10 = shl i64 %6, 8
                                                               %cmp219.i = icmp sqt i32 %4, 0
                                                               %wide.trip.count.i = zext i32 %4 to i64
                                                               %11 = add nsw i64 %wide.trip.count.i, -1
                                                               %xtraiter.i = and i64 %wide.trip.count.i, 3
                                                               %12 = icmp ult i64 \%11, 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.entry.i
                                                                   pregion for entry.entry.i:
                                                                    % local id x.0 = phi i64 [ 0, %9 ], [ %39, %if.end.i ]
                                                                    \%\overline{1}3 = \overline{a}d\overline{d} nuw nsw i64 % local id x.0, %10
                                                                    %conv.i = trunc i64 %13 to i32
                                                                    %cmp.i = icmp slt i32 %conv.i, %4
                                                                    %or.cond.i = and i1 %cmp219.i, %cmp.i
                                                                    br i1 %or.cond.i, label %for.body.lr.ph.i, label %if.end.i
                                                                                Τ
for.body.lr.ph.i:
%mul.i = mul nsw i32 %conv.i, %4
%sext.i = shl i64 %13, 32
%idxprom8.i = ashr exact i64 %sext.i, 32
%arrayidx9.i = getelementptr inbounds float, float* %2, i64 %idxprom8.i
%14 = sext i32 %mul.i to i64
%.pre.i114 = load float, float* %arrayidx9.i, align 4, !tbaa !12
br i1 %12, label %if.end.loopexit.unr-lcssa.i, label %for.body.i.preheader
                                              for.body.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.iv.next.3.i4 = phi i64 [ %indvars.iv.next.3.i, %for.body.i ], [ 0,
                  ... %for.body.i.preheader ]
                  %15 = phi float [ %31, %for.body.i ], [ %.pre.i114, %for.body.i.preheader ]
                  %16 = add nsw i64 %indvars.iv.next.3.i4, %14
                  %arrayidx.i = getelementptr inbounds float, float* %0, i64 %16
                  %17 = load float, float* %arrayidx.i, align 4, !tbaa !12
                  %mul4.i = fmul float %17, %3
                  %arrayidx6.i = getelementptr inbounds float, float* %1, i64
                  ... %indvars.iv.next.3.i4
                  %18 = load float, float* %arrayidx6.i, align 4, !tbaa !12
                  %19 = tail call float @llvm.fmuladd.f32(float %mul4.i, float %18, float %15)
                  ... #2
                  store float %19, float* %arrayidx9.i, align 4, !tbaa !12, !llvm.access.group
                   %indvars.iv.next.i = or i64 %indvars.iv.next.3.i4, 1
                   %20 = add nsw i64 %indvars.iv.next.i, %14
                  %arrayidx.1.i = getelementptr inbounds float, float* %0, i64 %20
                  %21 = load float, float* %arrayidx.1.i, align 4, !tbaa !12
                  %mul4.1.i = fmul float %21, %3
                  %arrayidx6.1.i = getelementptr inbounds float, float* %1, i64
                  ... %indvars.iv.next.i
                  %22 = load float, float* %arrayidx6.1.i, align 4, !tbaa !12
                  %23 = tail call float @llvm.fmuladd.f32(float %mul4.1.i, float %22, float
                  ... %19) #2
                  store float %23, float* %arrayidx9.i, align 4, !tbaa !12, !llvm.access.group
                  ... !16
                   %indvars.iv.next.1.i = or i64 %indvars.iv.next.3.i4, 2
                   %24 = add nsw i64 %indvars.iv.next.1.i, %14
                   %arrayidx.2.i = getelementptr inbounds float, float* %0, i64 %24
                   %25 = load float, float* %arrayidx.2.i, align 4, !tbaa !12
                  %mul4.2.i = fmul float %25, %3
                  %arrayidx6.2.i = getelementptr inbounds float, float* %1, i64
                  ... %indvars.iv.next.1.i
                  %26 = load float, float* %arrayidx6.2.i, align 4, !tbaa !12
                  %27 = tail call float @llvm.fmuladd.f32(float %mul4.2.i, float %26, float
                  ... %23) #2
                  store float %27, float* %arrayidx9.i, align 4, !tbaa !12, !llvm.access.group
                  ... !16
                   %indvars.iv.next.2.i = or i64 %indvars.iv.next.3.i4, 3
                  %28 = add nsw i64 %indvars.iv.next.2.i, %14
                  %arrayidx.3.i = getelementptr inbounds float, float* %0, i64 %28
                   %29 = load float, float* %arrayidx.3.i, align 4, !tbaa !12
                  %mul4.3.i = fmul float %29, %3
                  %arrayidx6.3.i = getelementptr inbounds float, float* %1, i64
                  ... %indvars.iv.next.2.i
                  %30 = load float, float* %arrayidx6.3.i, align 4, !tbaa !12
                  %31 = tail call float @llvm.fmuladd.f32(float %mul4.3.i, float %30, float
                  ... %27) #2
                  store float %31, float* %arrayidx9.i, align 4, !tbaa !12, !llvm.access.group
                   %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 [ %31, %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:
  %32 = phi float [ %.pre.i114, %for.body.lr.ph.i ], [ %.lcssa,
  ... %if.end.loopexit.unr-lcssa.i.loopexit ]
  %33 = 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 ], [ %33, %for.body.epil.i.preheader ]
                    %34 = phi float [ %38, %for.body.epil.i ], [ %32, %for.body.epil.i.preheader
                    %35 = add nsw i64 %indvars.iv.next.epil.i11, %14
                    %arrayidx.epil.i = getelementptr inbounds float, float* %0, i64 %35
                    %36 = load float, float* %arrayidx.epil.i, align 4, !tbaa !12
                    %mul4.epil.i = fmul float %36, %3
                    %arrayidx6.epil.i = getelementptr inbounds float, float* %1, i64
                    ... %indvars.iv.next.epil.i11
                    %37 = load float, float* %arrayidx6.epil.i, align 4, !tbaa !12
                    %38 = tail call float @llvm.fmuladd.f32(float %mul4.epil.i, float %37, float
                    ... %34) #2
                    store float %38, float* %arrayidx9.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!18
                                                                             F
                                          if.end.i.loopexit:
                                          br label %if.end.i
                                                   if.end.i:
                                                    %39 = add nuw nsw i64 % local id x.0, 1
                                                    \%exitcond = icmp eq i64 \%39, 256
                                                   br i1 %exitcond, label %gemver kernel3.exit, label
                                                   ... %pregion for entry.entry.i, !llvm.loop !20
                                                                                           F
                                                    gemver kernel3.exit:
                                                     ret void
```

CFG for 'pocl kernel gemver kernel3' function