```
%9:
                                                                     %10 = shl i64 %6, 8
                                                                     %cmp221.i = icmp sgt i32 %3, 0
                                                                     %11 = \text{sext i} 32 \% 4 \text{ to i} 64
                                                                     %wide.trip.count.i = zext i32 %3 to i64
                                                                     %12 = add nsw i64 %wide.trip.count.i, -1
                                                                     %xtraiter.i = and i64 %wide.trip.count.i, 3
                                                                     %13 = icmp ult i64 %12, 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:
                                                                           \%\overline{14} = \overline{a}d\overline{d} nuw nsw i64 % local id x.0, %10
                                                                           %conv.i = trunc i64 %14 to i32
                                                                           %cmp.i = icmp slt i32 %conv.i, %4
                                                                           br i1 %cmp.i, label %if.then.i, label %if.end.i
                                if.then.i:
                                %sext.i = shl i64 %14, 32
                                %idxprom.i = ashr exact i64 %sext.i, 32
                                %arrayidx.i = getelementptr inbounds float, float* %2, i64 %idxprom.i
                                store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !12,
                                ...!llvm.access.group!16
                                br i1 %cmp221.i, label %for.body.preheader.i, label %if.end.i
                                                                                      F
 for.body.preheader.i:
 br i1 %13, label %if.end.loopexit.unr-lcssa.i, label %for.body.i.preheader
                  Τ
                                                        F
                                              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 [ %35, %for.body.i ], [ 0.000000e+00, %for.body.i.preheader ]
                 %16 = mul nsw i64 %indvars.iv.next.3.i4, %11
                 %17 = add nsw i64 %16, %idxprom.i
                 %arrayidx5.i = getelementptr inbounds float, float* %0, i64 %17
                  %18 = load float, float* %arrayidx5.i, align 4, !tbaa !12
                  %arrayidx7.i = getelementptr inbounds float, float* %1, i64
                  .. %indvars.iv.next.3.i4
                 %19 = load float, float* %arrayidx7.i, align 4, !tbaa !12
                 %20 = tail call float @llvm.fmuladd.f32(float %18, float %19, float %15) #2
                 store float %20, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                 %indvars.iv.next.i = or i64 %indvars.iv.next.3.i4, 1
                 %21 = mul nsw i64 %indvars.iv.next.i, %11
                 %22 = add nsw i64 %21, %idxprom.i
                 %arrayidx5.1.i = getelementptr inbounds float, float* %0, i64 %22
                 %23 = load float, float* %arrayidx5.1.i, align 4, !tbaa !12
                 %arrayidx7.1.i = getelementptr inbounds float, float* %1, i64
                 ... %indvars.iv.next.i
                 %24 = load float, float* %arrayidx7.1.i, align 4, !tbaa !12
                 %25 = tail call float @llvm.fmuladd.f32(float %23, float %24, float %20) #2
                 store float %25, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                 %indvars.iv.next.1.i = or i64 %indvars.iv.next.3.i4, 2
                 %26 = mul nsw i64 %indvars.iv.next.1.i, %11
                 %27 = add nsw i64 %26, %idxprom.i
                 %arrayidx5.2.i = getelementptr inbounds float, float* %0, i64 %27
                 %28 = load float, float* %arrayidx5.2.i, align 4, !tbaa !12
                 %arrayidx7.2.i = getelementptr inbounds float, float* %1, i64
                  .. %indvars.iv.next.1.i
                 %29 = load float, float* %arrayidx7.2.i, align 4, !tbaa !12
                 %30 = tail call float @llvm.fmuladd.f32(float %28, float %29, float %25) #2
                 store float %30, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                 ... !16
                 %indvars.iv.next.2.i = or i64 %indvars.iv.next.3.i4, 3
                 %31 = mul nsw i64 %indvars.iv.next.2.i, %11
                 %32 = add nsw i64 %31, %idxprom.i
                 %arrayidx5.3.i = getelementptr inbounds float, float* %0, i64 %32
                 %33 = load float, float* %arrayidx5.3.i, align 4, !tbaa !12
                 %arrayidx7.3.i = getelementptr inbounds float, float* %1, i64
                 ... %indvars.iv.next.2.i
                 %34 = load float, float* %arrayidx7.3.i, align 4, !tbaa !12
                 %35 = tail call float @llvm.fmuladd.f32(float %33, float %34, float %30) #2
                 store float %35, 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.bodv.i
        if.end.loopexit.unr-lcssa.i.loopexit:
         %.lcssa = phi float [ %35, %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:
%36 = phi float [ 0.000000e+00, %for.body.preheader.i ], [ %.lcssa,
... %if.end.loopexit.unr-lcssa.i.loopexit ]
%37 = phi i64 [ 0, %for.body.preheader.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 ], [ %37, %for.body.epil.i.preheader ]
                   %38 = phi float [ %43, %for.body.epil.i ], [ %36, %for.body.epil.i.preheader
                   %39 = mul nsw i64 %indvars.iv.next.epil.i11, %11
                   %40 = add nsw i64 %39, %idxprom.i
                   %arrayidx5.epil.i = getelementptr inbounds float, float* %0, i64 %40
                   %41 = load float, float* %arrayidx5.epil.i, align 4, !tbaa !12
                   %arrayidx7.epil.i = getelementptr inbounds float, float* %1, i64
                   ... %indvars.iv.next.epil.i11
                   %42 = load float, float* %arrayidx7.epil.i, align 4, !tbaa !12
                   %43 = tail call float @llvm.fmuladd.f32(float %41, float %42, float %38) #2
                   store float %43, 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 il %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:
                                                    %44 = add nuw nsw i64 %_local_id_x.0, 1
                                                    \%exitcond = icmp eq i64 \%44, 2\overline{5}6
                                                    br i1 %exitcond, label %bicgKernel2.exit, label %pregion_for_entry.entry.i,
                                                    ...!llvm.loop!20
                                                                      Τ
                                                             bicgKernel2.exit:
                                                              ret void
```

CFG for 'pocl kernel bicgKernel2' function