```
%9:
                                        %mul.i.i = shl i64 %6, 8
                                        %cmp219.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], [\%17, \%if.end.r exit.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 slt i32 %conv.i, %4, !llvm.access.group !12
                             %or.cond.i = and i1 %cmp219.i, %cmp.i, !llvm.access.group !12
                             br i1 %or.cond.i, label %for.body.lr.ph.i, label %if.end.r exit.i,
                             ...!llvm.access.group!12
                                                                                         F
 for.body.lr.ph.i:
  %mul.i = mul nsw i32 %conv.i, %4, !llvm.access.group !12
  %sext.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
  %idxprom8.i = ashr exact i64 %sext.i, 32, !llvm.access.group !12
  %arrayidx9.i = getelementptr inbounds float, float* %2, i64 %idxprom8.i,
 ...!llvm.access.group!12
  %10 = sext i32 %mul.i to i64, !llvm.access.group !12
  %.pre.i = load float, float* %arrayidx9.i, align 4, !tbaa !14,
 ... !llvm.access.group !12
  br label %for.body.i, !llvm.access.group !12
for.body.i:
%indvars.iv.next.i2 = phi i64 [ %indvars.iv.next.i, %for.body.i ], [ 0,
... %for.body.lr.ph.i ]
%11 = phi float [ %16, %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
%mul4.i = fmul float %13, %3, !llvm.access.group !12
%arrayidx6.i = getelementptr inbounds float, float* %1, i64
... %indvars.iv.next.i2, !llvm.access.group !12
%14 = load float, float* %arrayidx6.i, align 4, !tbaa !14,
...!llvm.access.group!12
%15 = fmul float %mul4.i, %14, !llvm.access.group !12
%16 = fadd float %11, %15, !llvm.access.group !12
store float %16, float* %arrayidx9.i, align 4, !tbaa !14, !llvm.access.group
%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 %if.end.r exit.i.loopexit, label %for.body.i,
...!llvm.loop!18,!llvm.access.group!12
                              if.end.r exit.i.loopexit:
                               br label %if.end.r exit.i
                                         if.end.r exit.i:
                                          \%17 = add \text{ nuw nsw } i64 \% \text{ local } id x.0, 1
                                         %exitcond.not = icmp eq i\overline{6}4 %17, \overline{2}56
                                         br i1 %exitcond.not, 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