```
%mul3.i.i = shl i64 %5, 3
                                                                                                                                                                                      %mul.i.i = shl i64 %4, 5
                                                                                                                                                                                       %sub.i = add nsw i32 %2, -1
                                                                                                                                                                                       %8 = trunc i64 %5 to i32
                                                                                                                                                                                      %9 = mul i32 %8, %2
                                                                                                                                                                                      %10 = \text{shl i} 32 \%9, 3
                                                                                                                                                                                      %11 = trunc i64 %4 to i32
                                                                                                                                                                                      %12 = shl i32 %11, 5
                                                                                                                                                                                      %13 = add i32 %10, %12
                                                                                                                                                                                      %14 = zext i32 \%2 to i64
                                                                                                                                                                                      %15 = add i32 \%13, -8
                                                                                                                                                                                      %16 = \text{ or } i32 \%15, 7
                                                                                                                                                                                      %17 = \text{ or } i32 \%13, 1
                                                                                                                                                                                      %18 = shl i32 \%8, 3
                                                                                                                                                                                      %19 = \text{ or } i32 \%18, 1
                                                                                                                                                                                      %20 = mul i32 %19, %2
                                                                                                                                                                                      %21 = add i32 %20, %12
                                                                                                                                                                                      %22 = add i32 %18, -1
                                                                                                                                                                                      %23 = mul i32 %22, %2
                                                                                                                                                                                      %24 = add i32 %23, %12
                                                                                                                                                                                      br label %pregion for entry.pregion for init.i
                                                                                                                                                                    pregion_for_entry.pregion_for_init.i:
                                                                                                                                                                     % local id y.0 = phi i64 [0, \%7], [\%85, \%pregion for end.i]
                                                                                                                                                                     %25 = mul i64 %_local_id_y.0, %14
                                                                                                                                                                     %add6.i.i = add nuw nsw i64 % local_id_y.0, %mul3.i.i, !llvm.access.group !12
                                                                                                                                                                     %conv.i = trunc i64 %add6.i.i to i32, !llvm.access.group !12
                                                                                                                                                                     %cmp.i = icmp sgt i32 %conv.i, 0, !llvm.access.group !12
                                                                                                                                                                     %mul.i = mul nsw i32 %conv.i, %2
                                                                                                                                                                     %add25.i = add nuw nsw i32 %conv.i, 1
                                                                                                                                                                     %mul26.i = mul nsw i32 %add25.i, %2
                                                                                                                                                                     %sub31.i = add nsw i32 %conv.i, -1
                                                                                                                                                                     %mul32.i = mul nsw i32 %sub31.i, %2
                                                                                                                                                                     %cmp4.i = icmp sgt i32 %sub.i, %conv.i
                                                                                                                                                                     %or.cond = and i1 %cmp.i, %cmp4.i
                                                                                                                                                                     br i1 %or.cond, label %vector.scevcheck, label %pregion for end.i
                                                                                                                                                                                       Τ
                                                                                                       vector.scevcheck:
                                                                                                        %26 = trunc i64 %25 to i32
                                                                                                        %27 = add i32 %24, %26
                                                                                                        %28 = trunc i64 %25 to i32
                                                                                                        %29 = add i32 %21, %28
                                                                                                        %30 = trunc i64 %25 to i32
                                                                                                        %31 = add i32 \%17, \%30
                                                                                                        %32 = trunc i64 %25 to i32
                                                                                                        %33 = add i32 %16, %32
                                                                                                        %34 = trunc i64 %25 to i32
                                                                                                        %35 = add i32 \%13, \%34
                                                                                                        %36 = icmp sqt i32 %35, 2147483616
                                                                                                        %37 = icmp sqt i32 %33, 2147483616
                                                                                                        %38 = or i1 %36, %37
                                                                                                        %39 = icmp sgt i32 %31, 2147483616
                                                                                                        %40 = or i1 %38, %39
%41 = icmp sgt i32 %29, 2147483616
%42 = or i1 %40, %41
                                                                                                        %43 = icmp sgt i32 %27, 2147483616
                                                                                                        %44 = or i1 %42, %43
                                                                                                        br i1 %44, label %pregion for entry.entry.i.us.us.preheader, label %vector.ph
                                                                                                                            vector.ph:
                                                                                                                            %broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
                                                                                                                            %broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
                                                            pregion for entry.entry.i.us.us.preheader:
                                                                                                                            ... undef, <8 x i32> zeroinitializer
                                                                                                                            %broadcast.splatinsert16 = insertelement <8 x i32> undef, i32 %sub.i, i32 0 %broadcast.splat17 = shufflevector <8 x i32> %broadcast.splatinsert16, <8 x
                                                            br label %pregion for entry.entry.i.us.us
                                                                                                                            ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                            br label %vector.body
                                                                                                                      vector.body:
                                                                                                                      %index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
                                                                                                                      %vec.ind = phi <8 x i64> [ <i64 0, i64 1, i64 2, i64 3, i64 4, i64 5, i64 6,
                                                                                                                       ... i64 7>, %vector.ph ], [ %vec.ind.next, %vector.body ]
                                                                                                                      %45 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group
                                                                                                                      %46 = trunc <8 x i64> %45 to <8 x i32>, !llvm.access.group !12
                                                                                                                      %47 = icmp sgt <8 x i32> %46, zeroinitializer, !llvm.access.group !12
                                                                                                                      %48 = icmp sgt <8 x i32> %broadcast.splat17, %46, !llvm.access.group !12
                                                                                                                      %49 = \text{and } < 8 \text{ x i1} > %48, %47, !llvm.access.group !12}
                                                                                                                      %50 = \text{extractelement} < 8 \text{ x i} 32 > %46, i 32 0
                                                                                                                      %51 = add i32 %mul.i, %50, !llvm.access.group !12
                                                                                                                       %52 = sext i32 %51 to i64, !llvm.access.group !12
                                                                                                                       %53 = getelementptr inbounds float, float* %0, i64 %52, !llvm.access.group
                                                                                                                       %54 = bitcast float* %53 to <8 x float>*
                                                                                                                       %wide.load = load < 8 \times float >, < 8 \times float > %54, align 4, !tbaa !15,
                                                                                                                       ...!llvm.access.group!12
                                                                                                                      %55 = add i32 %51, -1, !llvm.access.group !12
                                                                                                                      %56 = sext i32 %55 to i64, !llvm.access.group !12
                                                                                                                       %57 = getelementptr inbounds float, float* %0, i64 %56, !llvm.access.group
                                                                                                                       %58 = bitcast float* %57 to <8 x float>*
                                                                                                                      %wide.load18 = load <8 x float>, <8 x float>* %58, align 4, !tbaa !15,
                                                                                                                       ..!llvm.access.group!12
                                                                                                                      %59 = fadd <8 x float> %wide.load, %wide.load18, !llvm.access.group !12
                                                                                                                       %60 = add i32 %51, 1, !llvm.access.group !12
                            pregion_for_entry.i.us.us:
                                                                                                                       %61 = sext i32 %60 to i64, !llvm.access.group !12
                            \%_local_id_x.0.us.us = phi i64 [ %84, %if.end.i.us.us ], [ 0,
                                                                                                                       %62 = getelementptr inbounds float, float* %0, i64 %61, !llvm.access.group
                            ... %pregion_for_entry.i.us.us.preheader ]
%add1.i.i.us.us = add nuw nsw i64 %_local_id_x.0.us.us, %mul.i.i,
                                                                                                                       \%63 = bitcast float* \%62 to <8 x float>*
                            ...!llvm.access.group!12
                                                                                                                       %wide.load19 = load <8 x float>, <8 x float>* %63, align 4, !tbaa !15,
                             %conv2.i.us.us = trunc i64 %add1.i.i.us.us to i32, !llvm.access.group !12
                                                                                                                       .. !llvm.access.group !12
                            %cmp7.i.us.us = icmp sgt i32 %conv2.i.us.us, 0, !llvm.access.group !12
                                                                                                                      %64 = fadd <8 x float> %59, %wide.load19, !llvm.access.group !12
%65 = add nsw i32 %mul26.i, %50, !llvm.access.group !12
%66 = sext i32 %65 to i64, !llvm.access.group !12
                             %cmp11.i.us.us = icmp sgt i32 %sub.i, %conv2.i.us.us, !llvm.access.group !12
                             %or.cond70.i.us.us = and i1 %cmp11.i.us.us, %cmp7.i.us.us,
                            ...!llvm.access.group!12
                                                                                                                       %67 = getelementptr inbounds float, float* %0, i64 %66, !llvm.access.group
                             br i1 %or.cond70.i.us.us, label %if.then.i.us.us, label %if.end.i.us.us,
                                                                                                                       ... !12
                             .. !llvm.access.group !12
                                                                                                                       \%68 = bitcast float* \%67 to <8 x float>*
                                                                                                                       \text{wide.load20} = \text{load} < 8 \text{ x float} > \text{, } < 8 \text{ x float} > \text{* } \%68, \text{ align 4, !tbaa !15, }
                                                                                                                       ...!llvm.access.group!12
                                                                                                                      %69 = fadd <8 x float> %64, %wide.load20, !llvm.access.group !12
                                                                                                                      %70 = add nsw i32 %mul32.i, %50, !llvm.access.group !12 %71 = sext i32 %70 to i64, !llvm.access.group !12
                                                                                                                       %72 = getelementptr inbounds float, float* %0, i64 %71, !llvm.access.group
                                                                                                                       \%73 = bitcast float* \%72 to < 8 x float>*
                                                                                                                       %wide.load21 = load <8 x float>, <8 x float>* %73, align 4, !tbaa !15,
                                                                                                                       ...!llvm.access.group!12
                                                                                                                      %74 = fadd <8 x float> %69, %wide.load21, !llvm.access.group !12
                                                                                                                      %75 = fmul <8 x float> %74, <float 0x3FC99999A0000000, float
                                                                                                                      ... 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
                                                                                                                      ... 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
                                                                                                                      ... 0x3FC99999A0000000>, !llvm.access.group !12
                                                                                                                      %76 = getelementptr inbounds float, float* %1, i64 %52, !llvm.access.group
                                                                                                                      %77 = bitcast float* %76 to <8 x float>*
                                                                                                                      call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %75, <8 x float>*
                                                                                                                       ... %77, i32 4, <8 x i1> %49), !tbaa !15, !llvm.access.group !12
                                                                                                                      %index.next = add i64 %index, 8
                                                                                                                       %vec.ind.next = add <8 x i64> %vec.ind, <i64 8, i64 8, i64 8, i64 8, i64 8,
                                                                                                                      ... i64 8, i64 8, i64 8>
                                                                                                                      %78 = icmp eq i64 %index.next, 32
                                                                                                                      br i1 %78, label %pregion for end.i.loopexit23, label %vector.body,
                                                                                                                      ...!llvm.loop!19
if.then.i.us.us:
%add.i.us.us = add i32 %mul.i, %conv2.i.us.us, !llvm.access.group !12
%idxprom.i.us.us = sext i32 %add.i.us.us to i64, !llvm.access.group !12
%arrayidx.i.us.us = getelementptr inbounds float, float* %0, i64
.. %idxprom.i.us.us, !llvm.access.group !12
%79 = load float, float* %arrayidx.i.us.us, align 4, !tbaa !15,
...!llvm.access.group!12
%add15.i.us.us = add i32 %add.i.us.us, -1, !llvm.access.group !12 %idxprom16.i.us.us = sext i32 %add15.i.us.us to i64, !llvm.access.group !12
%arrayidx17.i.us.us = getelementptr inbounds float, float* %0, i64
... %idxprom16.i.us.us, !llvm.access.group !12
%80 = load float, float* %arrayidx17.i.us.us, align 4, !tbaa !15,
...!llvm.access.group!12
%add18.i.us.us = fadd float %79, %80, !llvm.access.group !12
%add21.i.us.us = add i32 %add.i.us.us, 1, !llvm.access.group !12
%idxprom22.i.us.us = sext i32 %add21.i.us.us to i64, !llvm.access.group !12
%arrayidx23.i.us.us = getelementptr inbounds float, float* %0, i64
... %idxprom22.i.us.us, !llvm.access.group !12
%81 = load float, float* %arrayidx23.i.us.us, align 4, !tbaa !15,
...!llvm.access.group!12
%add24.i.us.us = fadd float %add18.i.us.us, %81, !llvm.access.group !12 %add27.i.us.us = add nsw i32 %mul26.i, %conv2.i.us.us, !llvm.access.group !12
%idxprom28.i.us.us = sext i32 %add27.i.us.us to i64, !llvm.access.group !12
%arrayidx29.i.us.us = getelementptr inbounds float, float* %0, i64
.. %idxprom28.i.us.us, !llvm.access.group !12
%82 = load float, float* %arrayidx29.i.us.us, align 4, !tbaa !15,
...!llvm.access.group!12
%add30.i.us.us = fadd float %add24.i.us.us, %82, !llvm.access.group !12
%add33.i.us.us = add nsw i32 %mul32.i, %conv2.i.us.us, !llvm.access.group !12
%idxprom34.i.us.us = sext i32 %add33.i.us.us to i64, !llvm.access.group !12 %arrayidx35.i.us.us = getelementptr inbounds float, float* %0, i64
... %idxprom34.i.us.us, !llvm.access.group !12
%83 = load float, float* %arrayidx35.i.us.us, align 4, !tbaa !15,
...!llvm.access.group!12
%add36.i.us.us = fadd float %add30.i.us.us, %83, !llvm.access.group !12
%mul37.i.us.us = fmul float %add36.i.us.us, 0x3FC99999A00000000,
...!llvm.access.group!12
%arrayidx41.i.us.us = getelementptr inbounds float, float* %1, i64 ... %idxprom.i.us.us, !llvm.access.group !12
store float %mul37.i.us.us, float* %arrayidx41.i.us.us, align 4, !tbaa !15,
...!llvm.access.group!12
br label %if.end.i.us.us, !llvm.access.group !12
                                                      if.end.i.us.us:
                                                      %84 = add nuw nsw i64 % local id x.0.us.us, 1
                                                      %exitcond.not = icmp eq i\overline{6}4 %8\overline{4}, \overline{3}2
                                                                                                                                                           pregion for end.i.loopexit23:
                                                      br i1 %exitcond.not, label %pregion_for_end.i.loopexit, label
                                                                                                                                                            br label %pregion for end.i
                                                      ... %pregion for entry.entry.i.us.us, !llvm.loop !22
                                                                                                     pregion for end.i.loopexit:
                                                                                                     br label %pregion for end.i
                                                                                                                                                                    pregion for end.i:
                                                                                                                                                                     \%85 = add nuw nsw i64 % local id y.0, 1
                                                                                                                                                                     \%exitcond3.not = icmp eq \overline{i}64 \%85, 8
                                                                                                                                                                     br i1 %exitcond3.not, label %runJacobi2D kernel1.exit, label
                                                                                                                                                                     ... %pregion for entry.pregion for init.i, !llvm.loop !23
                                                                                                                                                                     runJacobi2D kernel1.exit:
                                                                                                                                                                      ret void
```