```
%11:
                                                                                                                                                                                   %12 = \text{sext i} 32 \% 6 \text{ to i} 64
                                                                                                                                                                                   %13 = icmp slt i64 %12, 32
                                                                                                                                                                                   %14 = select i1 %13, i64 %12, i64 32
                                                                                                                                                                                   %15 = icmp slt i64 \%12, 8
                                                                                                                                                                                   %16 = select i1 %15, i64 %12, i64 8
                                                                                                                                                                                   %mul.i.i = shl i64 %8, 5
                                                                                                                                                                                   %mul3.i.i = shl i64 %9, 3
                                                                                                                                                                                   %cmp761.i = icmp sgt i32 %5, 0, !llvm.access.group !12
                                                                                                                                                                                   %wide.trip.count.i = zext i32 %5 to i64
                                                                                                                                                                                   %17 = icmp ugt i64 \%14, 1
                                                                                                                                                                                   %umax = select i1 %17, i64 %14, i64 1
                                                                                                                                                                                   %18 = icmp ugt i64 \%16, 1
                                                                                                                                                                                   %umax4 = select i1 %18, i64 %16, i64 1
                                                                                                                                                                                   %min.iters.check = icmp ult i64 %umax4, 8
                                                                                                                                                                                   br i1 %min.iters.check, label
                                                                                                                                                                                   ... %pregion for entry.pregion for init.i.preheader, label %vector.ph
                                                                                                                                                                                                                                  vector.ph:
                                                                                                                                                                                                                                   %n.vec = and i64 %umax4, -8
                                                                                                                                                                                                                                   %broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul3.i.i, i32 0
                                                                                                                                                                                                                                   %broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
                                                                                                                                                                                                                                   ... undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                   %broadcast.splatinsert6 = insertelement <8 x i32> undef, i32 %6, i32 0
                                                                                                                                                                                                                                   %broadcast.splat7 = shufflevector <8 x i32> %broadcast.splatinsert6, <8 x
                                                                                                                                                                                                                                   ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                   %broadcast.splatinsert8 = insertelement <8 x i32> undef, i32 %5, i32 0
                                                                                                                                                                                                                                   %broadcast.splat9 = shufflevector <8 x i32> %broadcast.splatinsert8, <8 x
                                                                                                                                                                                                                                   ... i32> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                   %broadcast.splatinsert11 = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
                                                                                                                                                                                                                                   %broadcast.splat12 = shufflevector <8 x i64> %broadcast.splatinsert11, <8 x
                                                                                                                                                                                                                                   ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                   %broadcast.splatinsert13 = insertelement <8 x float> undef, float %4, i32 0
                                                                                                                                                                                                                                   %broadcast.splat14 = shufflevector <8 x float> %broadcast.splatinsert13, <8
                                                                                                                                                                                                                                   ... x float> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                   %broadcast.splatinsert22 = insertelement <8 x float> undef, float %3, i32 0
                                                                                                                                                                                                                                   %broadcast.splat23 = shufflevector <8 x float> %broadcast.splatinsert22, <8
                                                                                                                                                                                                                                   ... x float> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                   %broadcast.splatinsert27 = insertelement <8 x i64> undef, i64
                                                                                                                                                                                                                                  ... %wide.trip.count.i, i32 0
                                                                                                                                                                                                                                   %broadcast.splat28 = shufflevector <8 x i64> %broadcast.splatinsert27, <8 x
                                                                                                                                                                                                                                   ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                   %broadcast.splatinsert31 = insertelement <8 x i64> undef, i64 %umax, i32 0
                                                                                                                                                                                                                                   %broadcast.splat32 = shufflevector <8 x i64> %broadcast.splatinsert31, <8 x
                                                                                                                                                                                                                                   ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                   br label %vector.body
                                                                                                                                                                                                                                                            vector.body:
                                                                                                                                                                                                                                                             %index = phi i64 [ 0, %vector.ph ], [ %index.next, %pregion for end.i33 ]
                                                                                                                                                                                                                                                              %vec.ind = phi < 8 \times 164 > [< 164 \ 0, 164 \ 1, 164 \ 2, 164 \ 3, 164 \ 4, 164 \ 5, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6, 164 \ 6
                                                                                                                                                                                                                                                            ... i64 7>, %vector.ph ], [ %vec.ind.next, %pregion_for_end.i33 ]
                                                                                                                                                                                                                                                             %19 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
                                                                                                                                                                                                                                                             %20 = trunc < 8 \times i64 > %19 \text{ to } < 8 \times i32 >, !llvm.access.group !12
                                                                                                                                                                                                                                                             %21 = mul nsw <8 x i32> %broadcast.splat7, %20, !llvm.access.group !12 %22 = mul nsw <8 x i32> %broadcast.splat9, %20
                                                                                                                                                                                                                                                             %23 = \text{sext} < 8 \times i32 > \%22 \text{ to } < 8 \times i64 > 3
                                                                                                                                                                                                                                                              br label %pregion for entry.entry.i10
                                                                                                                                                                                                                                                      pregion for entry.entry.i10:
                                                                                                                                                                                                                                                      %vec.phi = phi <8 x i64> [ zeroinitializer, %vector.body ], [ %47,
                                                                                                                                                                                                                                                      ... %if.end.r exit.i30 1
                                                                                                                                                                                                                                                      %24 = add <8 x i64 > %vec.phi, %broadcast.splat12, !llvm.access.group !12
                                                                                                                                                                                                                                                      %25 = trunc <8 x i64> %24 to <8 x i32>, !llvm.access.group !12
                                                                                                                                                                                                                                                      %26 = add nsw <8 x i32> %21, %25, !llvm.access.group !12
                                                                                                                                                                                                                                                      %27 = \text{sext} < 8 \times i32 > \%26 \text{ to} < 8 \times i64 >, !llvm.access.group !12
                                                                                                                                                                                                                                                      \%28 = getelementptr inbounds float, float* \%2, <8 x i64> \%27,
                                                                                                                                                                                                                                                      ...!llvm.access.group!12
                                                                                                                                                                                                                                                      %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                                                                                                                                                      ... x float*> %28, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true,
                                                                                                                                                                                                                                                      ... i1 true, i1 true, i1 true, <8 x float> undef), !tbaa !15, !llvm.access.group
                                                                                                                                                                                                                                                      ...!12
                                                                                                                                                                                                                                                      %29 = fmul <8 x float> %wide.masked.gather, %broadcast.splat14,
                                                                                                                                                                                                                                                      ...!llvm.access.group!12
                                                                                                                                                                                                                                                      call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %29, <8 x float*> ... %28, i32 4, <8 x i1> <i1 true, i1 true, i
                                                                                                                                                                                                                                                      ... i1 true, i1 true>), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                                                                                      br i1 %cmp761.i, label %for.body.lr.ph.i17, label %if.end.r exit.i30
                                                                                                                                                                                                      for.body.lr.ph.i17:
                                                                                                                                                                                                        %30 = mul nsw <8 x i32> %broadcast.splat9, %25, !llvm.access.group !12
                                                                                                                                                                                                       %31 = sext <8 x i32> %30 to <8 x i64>, !llvm.access.group !12
                                                                                                                                                                                                       br label %for.body.i18
                                                                                                                                                                                       for.body.i18:
                                                                                                                                                                                        \text{%vec.phi19} = \text{phi} < 8 \times \text{i64} > [\text{%44}, \text{%for.body.i18}], [\text{zeroinitializer},]
                                                                                                                                                                                        ... %for.body.lr.ph.i17 ]
                                                                                                                                                                                        \text{%vec.phi20} = \text{phi} < 8 \text{ x float} > [\%43, \%\text{for.body.i18}], [\%29, ]
                                                                                                                                                                                        ... %for.body.lr.ph.i17 ]
                                                                                                                                                                                        %32 = add nsw <8 x i64> %vec.phi19, %23, !llvm.access.group !12
                                                                                                                                                                                       %33 = getelementptr inbounds float, float* %0, <8 x i64> %32,
                                                                                                                                                                                        ...!llvm.access.group!12
                                                                                                                                                                                       %wide.masked.gather21 = call <8 x float> ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %33, i32 4, <8 x i1> <i1 true,
                                                                                                                                                                                       ... il true, ce x float>
                                                                                                                                                                                       ... undef), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                       %34 = fmul <8 x float> %wide.masked.gather21, %broadcast.splat23,
                                                                                                                                                                                       ...!llvm.access.group!12
                                                                                                                                                                                       %35 = add nsw <8 x i64> %vec.phi19, %31, !llvm.access.group !12
                                                                                                                                                                                        %36 = getelementptr inbounds float, float* %1, <8 x i64> %35,
                                                                                                                                                                                        ...!llvm.access.group!12
                                                                                                                                                                                        %wide.masked.gather24 = call <8 x float>
                                                                                                                                                                                       ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %36, i32 4, <8 x i1> <i1 true,
                                                                                                                                                                                        ... i1 true, <8 x float>
                                                                                                                                                                                       ... undef), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                        %37 = getelementptr inbounds float, float* %1, <8 x i64> %32,
                                                                                                                                                                                        ...!llvm.access.group!12
                                                                                                                                                                                       %wide.masked.gather25 = call <8 x float>
                                                                                                                                                                                       ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %37, i32 4, <8 x i1> <i1 true, i1 true, i1
                                                                                                                                                                                       ... undef), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                       %38 = fmul <8 x float> %wide.masked.gather25, %broadcast.splat23,
                                                                                                                                                                                        ...!llvm.access.group!12
                                                                                                                                                                                       %39 = getelementptr inbounds float, float* %0, <8 x i64> %35,
                                                                                                                                                                                       ...!llvm.access.group!12
                                                                                                                                                                                       %wide.masked.gather26 = call <8 x float>
                                                                                                                                                                                       ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %39, i32 4, <8 x i1> <i1 true, i1 true, i1
                                                                                                                                                                                       ... undef), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                       %40 = fmul <8 x float> %38, %wide.masked.gather26, !llvm.access.group !12 %41 = fmul <8 x float> %34, %wide.masked.gather24, !llvm.access.group !12 %42 = fadd <8 x float> %41, %40, !llvm.access.group !12
                                                                                                                                                                                       %43 = fadd <8 x float> %vec.phi20, %42, !llvm.access.group !12 call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %43, <8 x float*> ... %28, i32 4, <8 x i1> <i1 true, i1 true, i
                                                                                                                                                                                       ... i1 true, i1 true>), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                       %44 = add nuw nsw <8 x i64> %vec.phi19, <i64 1, i64 1, i64 1, i64 1,
                                                                                                                                                                                       ... i64 1, i64 1, i64 1>, !llvm.access.group !12
%45 = icmp eq <8 x i64> %44, %broadcast.splat28, !llvm.access.group !12
                                                                                                                                                                                        %46 = \text{extractelement} < 8 \times i1 > %45, i32 0
                                                                                                                                                                                       br i1 %46, label %if.end.r_exit.i30.loopexit, label %for.body.i18
                                                                                                                                                                                                                                                                 if.end.r_exit.i30.loopexit:
                                                                                                                                                                                                                                                                 br label %if.end.r exit.i30
                                                                                                                                                                                                                                                                 if.end.r exit.i30:
                                                                                                                                                                                                                                                                 %47 = add nuw <8 x i64> %vec.phi, <i64 1, i64 1, i64 1, i64 1, i64 1, i64 1,
                                                                                                                                                                                                                                                                 ... i64 1, i64 1>
                                                                                                                                                                                                                                                                  %48 = icmp eq <8 x i64> %47, %broadcast.splat32
                                                                                                                                                                                                                                                                  %49 = \text{extractelement} < 8 \text{ x i1} > %48, i32 0
                                                                                                                                                                                                                                                                  br i1 %49, label %pregion for end.i33, label %pregion for entry.entry.i10
                                                                                                                                                                                                                                       pregion for end.i33:
                                                                                                                                                                                                                                         %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>
                                                                                                                                                                                                                                         %50 = icmp eq i64 %index.next, %n.vec
                                                                                                                                                                                                                                         br i1 %50, label %middle.block, label %vector.body, !llvm.loop !19
                                                                                                                                                                                                                                                                                                                                                                       F
                                                                                                                                                                                                                        middle.block:
                                                                                                                                                                                                                         %cmp.n = icmp eq i64 %umax4, %n.vec
br i1 %cmp.n, label %syr2k_kernel.exit, label
                                                                                                                                                                                                                        ... %pregion_tor_entry.pregion for init.i.preheader
                                                                                                  pregion_for_entry.pregion_for_init.i.preheader:
                                                                                                  \begin{call} \be
                                                                                                   br label %pregion for entry.pregion for init.i
                                                                                    pregion_for_entry.pregion_for_init.i:
                                                                                     %_local_id_y.0 = phi i64 [-%63, %pregion_for_end.i ], [ %_local_id_y.0.ph,
                                                                                    %mul9.i = mul nsw i32 %conv2.i, %5
                                                                                     %51 = sext i32 %mul9.i to i64
                                                                                     br label %pregion for entry.entry.i
                                                    pregion for entry.entry.i:
                                                     % local id \bar{x}.0 = phi i64 [ 0, %pregion for entry.pregion for init.i ], [
                                                      %add1.i.i = add 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 %add.i = add nsw i32 %mul.i, %conv.i, !llvm.access.group !12
                                                     %idxprom.i = sext i32 %add.i to i64, !llvm.access.group !12
                                                     %arrayidx.i = getelementptr inbounds float, float* %2, i64 %idxprom.i,
                                                      .. !llvm.access.group !12
                                                     %52 = load float, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
                                                     ...!12
                                                     %mul6.i = fmul float %52, %4, !llvm.access.group !12 store float %mul6.i, float* %arrayidx.i, align 4, !tbaa !15,
                                                      ..!llvm.access.group!12
                                                     br i1 %cmp761.i, label %for.body.lr.ph.i, label %if.end.r exit.i,
                                                     ...!llvm.access.group!12
                                                                                                                                                                                  F
                 for.bodv.lr.ph.i:
                  %mul14.i = mul nsw i32 %conv.i, %5, !llvm.access.group !12
                  %53 = sext i32 %mul14.i to i64, !llvm.access.group !12
                  br label %for.body.i, !llvm.access.group !12
for.bodv.i:
 %indvars.iv.next.i3 = phi i64 [ %indvars.iv.next.i, %for.body.i ], [ 0,
 ... %for.body.lr.ph.i ]
%add33.i1 = phi float [ %add33.i, %for.body.i ], [ %mul6.i,
  .. %for.body.lr.ph.i ]
 %54 = add nsw i64 %indvars.iv.next.i3, %51, !llvm.access.group !12
 %arrayidx12.i = getelementptr inbounds float, float* %0, i64 %54,
  ..!llvm.access.group!12
 %55 = load float, float* %arrayidx12.i, align 4, !tbaa !15,
     !llvm.access.group !12
 %mul13.i = fmul float %55, %3, !llvm.access.group !12
 %56 = add nsw i64 %indvars.iv.next.i3, %53, !llvm.access.group !12
 %arrayidx17.i = getelementptr inbounds float, float* %1, i64 %56,
   .!llvm.access.group!12
 %57 = load float, float* %arrayidx17.i, align 4, !tbaa !15,
  .. !llvm.access.group !12
 %arrayidx22.i = getelementptr inbounds float, float* %1, i64 %54,
   .!llvm.access.group!12
 %58 = load float, float* %arrayidx22.i, align 4, !tbaa !15,
  ..!llvm.access.group!12
 %mul23.i = fmul float %58, %3, !llvm.access.group !12
 %arrayidx27.i = getelementptr inbounds float, float* %0, i64 %56,
   .!llvm.access.group!12
 %59 = load float, float* %arrayidx27.i, align 4, !tbaa !15,
  .. !llvm.access.group !12
%mul28.i = fmul float %mul23.i, %59, !llvm.access.group !12
%60 = fmul float %mul13.i, %57, !llvm.access.group !12
%61 = fadd float %60, %mul28.i, !llvm.access.group !12
%add33.i = fadd float %add33.i1, %61, !llvm.access.group !12 store float %add33.i, float* %arrayidx.i, align 4, !tbaa !15,
 ...!llvm.access.group!12
%indvars.iv.next.i = add nuw nsw i64 %indvars.iv.next.i3, 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!22,!llvm.access.group!12
                                                            if.end.r exit.i.loopexit:
                                                             br label %if.end.r exit.i
                                                                                                      if.end.r exit.i:
                                                                                                      \%62 = add nuw i64 \% local id x.0, 1
                                                                                                      %exitcond.not = icmp eq i6\overline{4} %62, %umax
                                                                                                      br i1 %exitcond.not, label %pregion for end.i, label
                                                                                                      ... %pregion for entry.entry.i, !llvm.loop 124
                                                                                                             pregion for end.i:
                                                                                                              %63 = add nuw i64 %_local_id_y.0, 1
                                                                                                             %exitcond5.not = icmp eq i64 %63, %umax4
br i1 %exitcond5.not, label %syr2k_kernel.exit.loopexit, label
                                                                                                              ... %pregion for entry.pregion for init.i, !llvm.loop !27
                                                                                                                                                    syr2k kernel.exit.loopexit:
                                                                                                                                                      br label %syr2k kernel.exit
```

ret void

CFG for '_pocl_kernel_syr2k_kernel' function

syr2k kernel.exit: