T
F
T
F
T
F
T
F
T
F
T
F

```
%10:
 %11 = sext i32 %4 to i64
 %12 = icmp slt i64 %11, 32
 %13 = select i1 %12, i64 %11, i64 32
 %14 = sext i32 %5 to i64
 %15 = icmp slt i64 %14, 8
 %16 = select i1 %15, i64 %14, i64 8
 %mul.i.i = shl i64 %7, 5
 %mul3.i.i = shl i64 %8, 3
 %17 = tail call float @llvm.sqrt.f32(float %3) #5, !llvm.access.group !12
 %18 = icmp ugt i64 %13, 1
 %umax = select i1 %18, i64 %13, i64 1
 %19 = icmp ugt i64 %16, 1
 %umax1 = select i1 %19, i64 %16, i64 1
 %min.iters.check = icmp ult i64 %umax1, 8
 br i1 %min.iters.check, label
... %pregion_for_entry.pregion_for_init.i.preheader, label %vector.ph
```

```
vector.ph:
 %n.vec = and i64 %umax1, -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.splatinsert3 = insertelement <8 x i32> undef, i32 %4, i32 0
 %broadcast.splat4 = shufflevector <8 x i32> %broadcast.splatinsert3, <8 x
... i32> undef, <8 x i32> zeroinitializer
 %broadcast.splatinsert6 = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
 %broadcast.splat7 = shufflevector <8 x i64> %broadcast.splatinsert6, <8 x
... i64> undef, <8 x i32> zeroinitializer
 %broadcast.splatinsert10 = insertelement <8 x float> undef, float %17, i32 0
 %broadcast.splat11 = shufflevector <8 x float> %broadcast.splatinsert10, <8
... x float> undef, <8 x i32> zeroinitializer
 %broadcast.splatinsert12 = insertelement <8 x i64> undef, i64 %umax, i32 0
 %broadcast.splat13 = shufflevector <8 x i64> %broadcast.splatinsert12, <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.i14 ]
 %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, %pregion_for_end.i14 ]
 %20 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
 %21 = trunc <8 x i64> %20 to <8 x i32>, !llvm.access.group !12
 %22 = mul nsw <8 x i32> %broadcast.splat4, %21, !llvm.access.group !12
 br label %pregion_for_entry.entry.i5
```

```
pregion_for_entry.entry.i5:
 %vec.phi = phi <8 x i64> [ zeroinitializer, %vector.body ], [ %35,
... %pregion_for_entry.entry.i5 ]
 %23 = add <8 x i64> %vec.phi, %broadcast.splat7, !llvm.access.group !12
 %24 = trunc <8 x i64> %23 to <8 x i32>, !llvm.access.group !12
 %25 = shl <8 x i64> %23, <i64 32, i64 32, i64 32, i64 32, i64 32, i64 32,
... i64 32, i64 32>, !llvm.access.group !12
 %26 = ashr exact <8 x i64> %25, <i64 32, i64 32, i64 32, i64 32, i64 32, i64
... 32, i64 32, i64 32>, !llvm.access.group !12
 %27 = getelementptr inbounds float, float* %0, <8 x i64> %26,
... !llvm.access.group !12
 %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
... x float*> %27, 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
 %28 = add nsw <8 x i32> %22, %24, !llvm.access.group !12
 %29 = sext <8 x i32> %28 to <8 x i64>, !llvm.access.group !12
 %30 = getelementptr inbounds float, float* %2, <8 x i64> %29,
... !llvm.access.group !12
 %wide.masked.gather8 = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
... x float*> %30, 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
 %31 = fsub <8 x float> %wide.masked.gather8, %wide.masked.gather,
... !llvm.access.group !12
 call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %31, <8 x float*>
... %30, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1 true,
... i1 true, i1 true>), !tbaa !15, !llvm.access.group !12
 %32 = getelementptr inbounds float, float* %1, <8 x i64> %26,
... !llvm.access.group !12
 %wide.masked.gather9 = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
... x float*> %32, 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
 %33 = fmul <8 x float> %broadcast.splat11, %wide.masked.gather9,
... !llvm.access.group !12
 %34 = fdiv <8 x float> %31, %33, !fpmath !19, !llvm.access.group !12
 call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %34, <8 x float*>
... %30, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1 true,
... i1 true, i1 true>), !tbaa !15, !llvm.access.group !12
 %35 = add nuw <8 x i64> %vec.phi, <i64 1, i64 1, i64 1, i64 1, i64 1, i64 1,
... i64 1, i64 1>
 %36 = icmp eq <8 x i64> %35, %broadcast.splat13
 %37 = extractelement <8 x i1> %36, i32 0
 br i1 %37, label %pregion_for_end.i14, label %pregion_for_entry.entry.i5
```

```
pregion_for_end.i14:
 %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>
 %38 = icmp eq i64 %index.next, %n.vec
 br i1 %38, label %middle.block, label %vector.body, !llvm.loop !20
```

```
middle.block:
 %cmp.n = icmp eq i64 %umax1, %n.vec
 br i1 %cmp.n, label %reduce_kernel.exit, label
... %pregion_for_entry.pregion_for_init.i.preheader
```

```
pregion_for_entry.pregion_for_init.i.preheader:
 %_local_id_y.0.ph = phi i64 [ 0, %10 ], [ %n.vec, %middle.block ]
 br label %pregion_for_entry.pregion_for_init.i
```

```
pregion_for_entry.pregion_for_init.i:
 %_local_id_y.0 = phi i64 [ %43, %pregion_for_end.i ], [ %_local_id_y.0.ph,
... %pregion_for_entry.pregion_for_init.i.preheader ]
 %add6.i.i = add i64 %_local_id_y.0, %mul3.i.i, !llvm.access.group !12
 %conv2.i = trunc i64 %add6.i.i to i32, !llvm.access.group !12
 %mul.i = mul nsw i32 %conv2.i, %4, !llvm.access.group !12
 br label %pregion_for_entry.entry.i
```

```
pregion_for_entry.entry.i:
 %_local_id_x.0 = phi i64 [ 0, %pregion_for_entry.pregion_for_init.i ], [
... %42, %pregion_for_entry.entry.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
 %sext.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
 %idxprom.i = ashr exact i64 %sext.i, 32, !llvm.access.group !12
 %arrayidx.i = getelementptr inbounds float, float* %0, i64 %idxprom.i,
... !llvm.access.group !12
 %39 = load float, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
... !12
 %add.i = add nsw i32 %mul.i, %conv.i, !llvm.access.group !12
 %idxprom6.i = sext i32 %add.i to i64, !llvm.access.group !12
 %arrayidx7.i = getelementptr inbounds float, float* %2, i64 %idxprom6.i,
... !llvm.access.group !12
 %40 = load float, float* %arrayidx7.i, align 4, !tbaa !15,
... !llvm.access.group !12
 %sub.i = fsub float %40, %39, !llvm.access.group !12
 store float %sub.i, float* %arrayidx7.i, align 4, !tbaa !15,
... !llvm.access.group !12
 %arrayidx10.i = getelementptr inbounds float, float* %1, i64 %idxprom.i,
... !llvm.access.group !12
 %41 = load float, float* %arrayidx10.i, align 4, !tbaa !15,
... !llvm.access.group !12
 %mul11.i = fmul float %17, %41, !llvm.access.group !12
 %div.i = fdiv float %sub.i, %mul11.i, !fpmath !19, !llvm.access.group !12
 store float %div.i, float* %arrayidx7.i, align 4, !tbaa !15,
... !llvm.access.group !12
 %42 = add nuw i64 %_local_id_x.0, 1
 %exitcond.not = icmp eq i64 %42, %umax
 br i1 %exitcond.not, label %pregion_for_end.i, label
... %pregion_for_entry.entry.i, !llvm.loop !23
```

```
pregion_for_end.i:
 %43 = add nuw i64 %_local_id_y.0, 1
 %exitcond2.not = icmp eq i64 %43, %umax1
 br i1 %exitcond2.not, label %reduce_kernel.exit.loopexit, label
... %pregion_for_entry.pregion_for_init.i, !llvm.loop !26
```

```
reduce_kernel.exit.loopexit:
 br label %reduce_kernel.exit
```

```
reduce_kernel.exit:
 ret void
```

CFG for '_pocl_kernel_reduce_kernel' function