

```
%11:
%12 = sext i32 %2 to i64
%13 = icmp slt i64 %12, 32
%14 = select i1 %13, i64 %12, i64 32
%15 = sext i32 %1 to i64
%16 = icmp slt i64 %15, 8
%17 = select i1 %16, i64 %15, i64 8
%mul.i.i = shl i64 %8, 5
%mul3.i.i = shl i64 %9, 3
%mul6.i = mul i32 %6, %1, !llvm.access.group !12
%18 = icmp ugt i64 %14, 1
%umax = select i1 %18, i64 %14, i64 1
%19 = icmp ugt i64 %17, 1
%umax1 = select i1 %19, i64 %17, 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
```

T

F

```
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 %mul6.i, i32 0
%broadcast.splat4 = shufflevector <8 x i32> %broadcast.splatinsert3, <8 x
... i32> undef, <8 x i32> zeroinitializer
%broadcast.splatinsert5 = insertelement <8 x i32> undef, i32 %2, i32 0
%broadcast.splat6 = shufflevector <8 x i32> %broadcast.splatinsert5, <8 x
... i32> undef, <8 x i32> zeroinitializer
%broadcast.splatinsert8 = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
%broadcast.splat9 = shufflevector <8 x i64> %broadcast.splatinsert8, <8 x
... i64> undef, <8 x i32> zeroinitializer
%broadcast.splatinsert10 = insertelement <8 x i64> undef, i64 %umax, i32 0
%broadcast.splat11 = shufflevector <8 x i64> %broadcast.splatinsert10, <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.i12 ]
%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.i12 ]
%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 = add <8 x i32> %broadcast.splat4, %21, !llvm.access.group !12
%23 = mul <8 x i32> %22, %broadcast.splat6, !llvm.access.group !12
br label %pregion_for_entry.entry.i7
```

```
pregion_for_entry.entry.i7:
%vec.phi = phi <8 x i64> [ zeroinitializer, %vector.body ], [ %30,
... %pregion_for_entry.entry.i7 ]
%24 = add <8 x i64> %vec.phi, %broadcast.splat9, !llvm.access.group !12
%25 = trunc <8 x i64> %24 to <8 x i32>, !llvm.access.group !12
%26 = add <8 x i32> %23, %25, !llvm.access.group !12
%27 = sext <8 x i32> %26 to <8 x i64>, !llvm.access.group !12
%28 = getelementptr inbounds float, float* %5, <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 = getelementptr inbounds float, float* %3, <8 x i64> %27,
... !llvm.access.group !12
call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float>
... %wide.masked.gather, <8 x float*> %29, i32 4, <8 x i1> <i1 true, i1 true, i1
... true, i1 true, i1 true, i1 true, i1 true>), !tbaa !15,
... !llvm.access.group !12
%30 = add nuw <8 x i64> %vec.phi, <i64 1, i64 1, i64 1, i64 1, i64 1, i64 1,
... i64 1, i64 1>
%31 = icmp eq <8 x i64> %30, %broadcast.splat11
%32 = extractelement <8 x i1> %31, i32 0
br i1 %32, label %pregion_for_end.i12, label %pregion_for_entry.entry.i7
```

T

F

```
pregion_for_end.i12:
%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>
%33 = icmp eq i64 %index.next, %n.vec
br i1 %33, label %middle.block, label %vector.body, !llvm.loop !19
```

T

F

```
middle.block:
%cmp.n = icmp eq i64 %umax1, %n.vec
br i1 %cmp.n, label %doitgen_kernel2.exit, label
... %pregion_for_entry.pregion_for_init.i.preheader
```

T

F

```
pregion_for_entry.pregion_for_init.i.preheader:
% local_id_y.0.ph = phi i64 [ 0, %11 ], [ %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 [%36, %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
%reass.add.i = add i32 %mul6.i, %conv2.i, !llvm.access.group !12
%reass.mul.i = mul i32 %reass.add.i, %2, !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 ], [
... %35, %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
%add8.i = add i32 %reass.mul.i, %conv.i, !llvm.access.group !12
%idxprom.i = sext i32 %add8.i to i64, !llvm.access.group !12
%arrayidx.i = getelementptr inbounds float, float* %5, i64 %idxprom.i,
... !llvm.access.group !12
%34 = load float, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
... !12
%arrayidx15.i = getelementptr inbounds float, float* %3, i64 %idxprom.i,
... !llvm.access.group !12
store float %34, float* %arrayidx15.i, align 4, !tbaa !15,
... !llvm.access.group !12
%35 = add nuw i64 % local_id_x.0, 1
%exitcond.not = icmp eq i64 %35, %umax
br i1 %exitcond.not, label %pregion_for_end.i, label
... %pregion_for_entry.entry.i, !llvm.loop !22
```

T

F

```
pregion_for_end.i:
%36 = add nuw i64 % local_id_y.0, 1
%exitcond2.not = icmp eq i64 %36, %umax1
br i1 %exitcond2.not, label %doitgen_kernel2.exit.loopexit, label
... %pregion_for_entry.pregion_for_init.i, !llvm.loop !25
```

T

F

```
doitgen_kernel2.exit.loopexit:
br label %doitgen_kernel2.exit
```

```
doitgen_kernel2.exit:
ret void
```

CFG for ' pocl_kernel_doitgen_kernel2' function