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

```
%8:
 %mul.i.i = shl i64 %5, 5
 %cmp218.i = icmp sgt i32 %3, 0, !llvm.access.group !12
 %9 = zext i32 %3 to i64
 br i1 %cmp218.i, label %pregion_for_entry.entry.i.us.preheader, label
... %mvt_kernel2.exit
```

```
pregion_for_entry.entry.i.us.preheader:
 br label %pregion_for_entry.entry.i.us
```

```
pregion_for_entry.entry.i.us:
 %_local_id_x.0.us = phi i64 [ %23, %if.end.r_exit.i.us.1 ], [ 0,
... %pregion_for_entry.entry.i.us.preheader ]
 %add1.i.i.us = add nuw nsw i64 %_local_id_x.0.us, %mul.i.i,
... !llvm.access.group !12
 %conv.i.us = trunc i64 %add1.i.i.us to i32, !llvm.access.group !12
 %cmp.i.us = icmp slt i32 %conv.i.us, %3, !llvm.access.group !12
 br i1 %cmp.i.us, label %for.body.lr.ph.i.us, label %if.end.r_exit.i.us,
... !llvm.access.group !12
```

```
for.body.lr.ph.i.us:
 %sext.i.us = shl i64 %add1.i.i.us, 32, !llvm.access.group !12
 %idxprom7.i.us = ashr exact i64 %sext.i.us, 32, !llvm.access.group !12
 %arrayidx8.i.us = getelementptr inbounds float, float* %1, i64
... %idxprom7.i.us, !llvm.access.group !12
 %.pre.i1.us4 = load float, float* %arrayidx8.i.us, align 4, !tbaa !14,
... !llvm.access.group !12
 br label %for.body.i.us, !llvm.access.group !12
```

```
for.body.i.us:
 %indvars.iv.next.i3.us = phi i64 [ %indvars.iv.next.i.us, %for.body.i.us ],
... [ 0, %for.body.lr.ph.i.us ]
 %10 = phi float [ %15, %for.body.i.us ], [ %.pre.i1.us4,
... %for.body.lr.ph.i.us ]
 %11 = mul nuw nsw i64 %indvars.iv.next.i3.us, %9, !llvm.access.group !12
 %12 = add nsw i64 %11, %idxprom7.i.us, !llvm.access.group !12
 %arrayidx.i.us = getelementptr inbounds float, float* %0, i64 %12,
... !llvm.access.group !12
 %13 = load float, float* %arrayidx.i.us, align 4, !tbaa !14,
... !llvm.access.group !12
 %arrayidx5.i.us = getelementptr inbounds float, float* %2, i64
... %indvars.iv.next.i3.us, !llvm.access.group !12
 %14 = load float, float* %arrayidx5.i.us, align 4, !tbaa !14,
... !llvm.access.group !12
 %15 = tail call float @llvm.fmuladd.f32(float %13, float %14, float %10) #2,
... !llvm.access.group !12
 store float %15, float* %arrayidx8.i.us, align 4, !tbaa !14,
... !llvm.access.group !12
 %indvars.iv.next.i.us = add nuw nsw i64 %indvars.iv.next.i3.us, 1,
... !llvm.access.group !12
 %exitcond.not.i.us = icmp eq i64 %indvars.iv.next.i.us, %9,
... !llvm.access.group !12
 br i1 %exitcond.not.i.us, label %if.end.r_exit.i.us.loopexit, label
... %for.body.i.us, !llvm.loop !18, !llvm.access.group !12
```

```
if.end.r_exit.i.us.loopexit:
 br label %if.end.r_exit.i.us
```

```
if.end.r_exit.i.us:
 %16 = or i64 %_local_id_x.0.us, 1
 %add1.i.i.us.1 = add nuw nsw i64 %16, %mul.i.i, !llvm.access.group !12
 %conv.i.us.1 = trunc i64 %add1.i.i.us.1 to i32, !llvm.access.group !12
 %cmp.i.us.1 = icmp slt i32 %conv.i.us.1, %3, !llvm.access.group !12
 br i1 %cmp.i.us.1, label %for.body.lr.ph.i.us.1, label
... %if.end.r_exit.i.us.1, !llvm.access.group !12
```

```
for.body.lr.ph.i.us.1:
 %sext.i.us.1 = shl i64 %add1.i.i.us.1, 32, !llvm.access.group !12
 %idxprom7.i.us.1 = ashr exact i64 %sext.i.us.1, 32, !llvm.access.group !12
 %arrayidx8.i.us.1 = getelementptr inbounds float, float* %1, i64
... %idxprom7.i.us.1, !llvm.access.group !12
 %.pre.i1.us4.1 = load float, float* %arrayidx8.i.us.1, align 4, !tbaa !14,
... !llvm.access.group !12
 br label %for.body.i.us.1, !llvm.access.group !12
```

```
for.body.i.us.1:
 %indvars.iv.next.i3.us.1 = phi i64 [ %indvars.iv.next.i.us.1,
... %for.body.i.us.1 ], [ 0, %for.body.lr.ph.i.us.1 ]
 %17 = phi float [ %22, %for.body.i.us.1 ], [ %.pre.i1.us4.1,
... %for.body.lr.ph.i.us.1 ]
 %18 = mul nuw nsw i64 %indvars.iv.next.i3.us.1, %9, !llvm.access.group !12
 %19 = add nsw i64 %18, %idxprom7.i.us.1, !llvm.access.group !12
 %arrayidx.i.us.1 = getelementptr inbounds float, float* %0, i64 %19,
... !llvm.access.group !12
 %20 = load float, float* %arrayidx.i.us.1, align 4, !tbaa !14,
... !llvm.access.group !12
 %arrayidx5.i.us.1 = getelementptr inbounds float, float* %2, i64
... %indvars.iv.next.i3.us.1, !llvm.access.group !12
 %21 = load float, float* %arrayidx5.i.us.1, align 4, !tbaa !14,
... !llvm.access.group !12
 %22 = tail call float @llvm.fmuladd.f32(float %20, float %21, float %17) #2,
... !llvm.access.group !12
 store float %22, float* %arrayidx8.i.us.1, align 4, !tbaa !14,
... !llvm.access.group !12
 %indvars.iv.next.i.us.1 = add nuw nsw i64 %indvars.iv.next.i3.us.1, 1,
... !llvm.access.group !12
 %exitcond.not.i.us.1 = icmp eq i64 %indvars.iv.next.i.us.1, %9,
... !llvm.access.group !12
 br i1 %exitcond.not.i.us.1, label %if.end.r_exit.i.us.1.loopexit, label
... %for.body.i.us.1, !llvm.loop !18, !llvm.access.group !12
```

```
if.end.r_exit.i.us.1.loopexit:
 br label %if.end.r_exit.i.us.1
```

```
if.end.r_exit.i.us.1:
 %23 = add nuw nsw i64 %_local_id_x.0.us, 2
 %exitcond.not.1 = icmp eq i64 %23, 32
 br i1 %exitcond.not.1, label %mvt_kernel2.exit.loopexit, label
... %pregion_for_entry.entry.i.us, !llvm.loop !20
```

```
mvt_kernel2.exit.loopexit:
 br label %mvt_kernel2.exit
```

```
mvt_kernel2.exit:
 ret void
```

CFG for '_pocl_kernel_mvt_kernel2' function