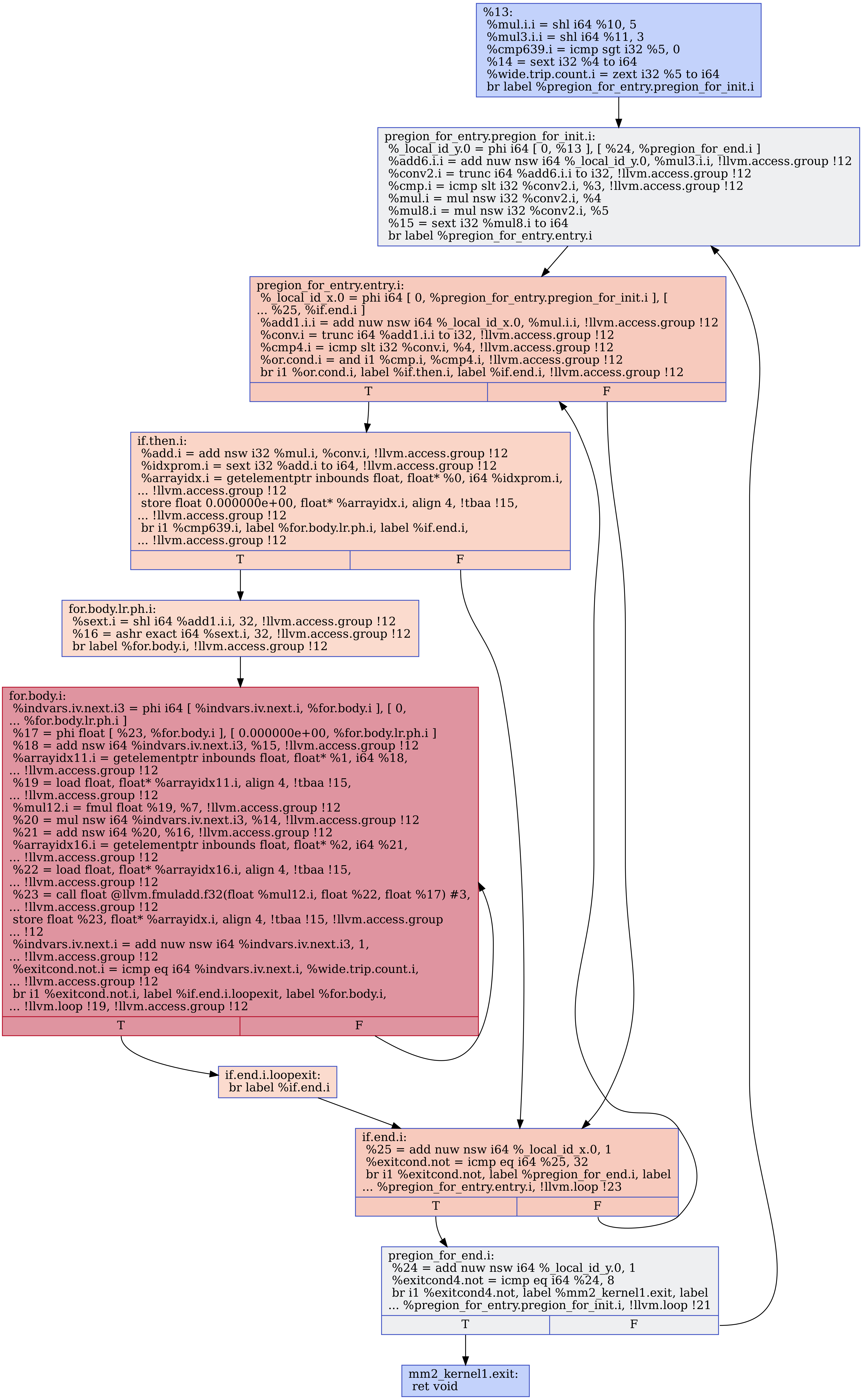

```
%13:
 %mul.i.i = shl i64 %10, 5
 %mul3.i.i = shl i64 %11, 3
 %cmp639.i = icmp sgt i32 %5, 0
 %14 = sext i32 %4 to i64
 %wide.trip.count.i = zext i32 %5 to i64
 br label %pregion_for_entry.pregion_for_init.i
```

```
pregion_for_entry.pregion_for_init.i:
 %_local_id_y.0 = phi i64 [ 0, %13 ], [ %24, %pregion_for_end.i ]
 %add6.i.i = add nuw nsw 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
 %cmp.i = icmp slt i32 %conv2.i, %3, !llvm.access.group !12
 %mul.i = mul nsw i32 %conv2.i, %4
 %mul8.i = mul nsw i32 %conv2.i, %5
 %15 = sext i32 %mul8.i to i64
 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 ], [
... %25, %if.end.i ]
 %add1.i.i = add nuw nsw 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
 %cmp4.i = icmp slt i32 %conv.i, %4, !llvm.access.group !12
 %or.cond.i = and i1 %cmp.i, %cmp4.i, !llvm.access.group !12
 br i1 %or.cond.i, label %if.then.i, label %if.end.i, !llvm.access.group !12
```

```
if.then.i:
 %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* %0, i64 %idxprom.i,
... !llvm.access.group !12
 store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !15,
... !llvm.access.group !12
 br i1 %cmp639.i, label %for.body.lr.ph.i, label %if.end.i,
... !llvm.access.group !12
```

```
for.body.lr.ph.i:
 %sext.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
 %16 = ashr exact i64 %sext.i, 32, !llvm.access.group !12
 br label %for.body.i, !llvm.access.group !12
```

```
for.body.i:
 %indvars.iv.next.i3 = phi i64 [ %indvars.iv.next.i, %for.body.i ], [ 0,
... %for.body.lr.ph.i ]
 %17 = phi float [ %23, %for.body.i ], [ 0.000000e+00, %for.body.lr.ph.i ]
 %18 = add nsw i64 %indvars.iv.next.i3, %15, !llvm.access.group !12
 %arrayidx11.i = getelementptr inbounds float, float* %1, i64 %18,
... !llvm.access.group !12
 %19 = load float, float* %arrayidx11.i, align 4, !tbaa !15,
... !llvm.access.group !12
 %mul12.i = fmul float %19, %7, !llvm.access.group !12
 %20 = mul nsw i64 %indvars.iv.next.i3, %14, !llvm.access.group !12
 %21 = add nsw i64 %20, %16, !llvm.access.group !12
 %arrayidx16.i = getelementptr inbounds float, float* %2, i64 %21,
... !llvm.access.group !12
 %22 = load float, float* %arrayidx16.i, align 4, !tbaa !15,
... !llvm.access.group !12
 %23 = call float @llvm.fmuladd.f32(float %mul12.i, float %22, float %17) #3,
... !llvm.access.group !12
 store float %23, 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.i.loopexit, label %for.body.i,
... !llvm.loop !19, !llvm.access.group !12
```

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

```
if.end.i:
 %25 = add nuw nsw i64 %_local_id_x.0, 1
 %exitcond.not = icmp eq i64 %25, 32
 br i1 %exitcond.not, label %pregion_for_end.i, label
... %pregion_for_entry.entry.i, !llvm.loop !23
```

```
pregion_for_end.i:
 %24 = add nuw nsw i64 %_local_id_y.0, 1
 %exitcond4.not = icmp eq i64 %24, 8
 br i1 %exitcond4.not, label %mm2_kernel1.exit, label
... %pregion_for_entry.pregion_for_init.i, !llvm.loop !21
```

```
mm2_kernel1.exit:
 ret void
```

CFG for '_pocl_kernel_mm2_kernel1' function