

```
%i3:
%14 = sext i32 %4 to i64
%15 = icmp slt i64 %14, 32
%16 = select i1 %15, i64 %14, i64 32
%17 = sext i32 %3 to i64
%18 = icmp slt i64 %17, 8
%19 = select i1 %18, i64 %17, i64 8
%mul.i.i = shl i64 %10, 5
%mul3.i.i = shl i64 %11, 3
%cmp639.i = icmp sgt i32 %5, 0, !llvm.access.group !12
%wide.trip.count.i = zext i32 %5 to i64
%20 = icmp ugt i64 %16, 1
%umax = select i1 %20, i64 %16, i64 1
%21 = icmp ugt i64 %19, 1
%umax28 = select i1 %21, i64 %19, i64 1
br i1 %cmp639.i, label %pregon_for_entry.pregon_for_init.i.us.preheader,
... label %pregon_for_entry.pregon_for_init.i.preheader_
```

```
pregon_for_entry.pregon_for_init.i.us.preheader:
br label %pregon_for_entry.pregon_for_init.i.us
```

```
pregon_for_entry.pregon_for_init.i.preheader:
%22 = add nsw i64 %umax, -1
%23 = trunc i64 %11 to i32
%24 = mul i32 %23, %4
%25 = shl i32 %24, 3
%26 = zext i32 %25 to i64
%27 = shl i64 %10, 5
%28 = add i64 %27, %26
%29 = zext i32 %4 to i64
br label %pregon_for_entry.pregon_for_init.i
```

```
pregon_for_entry.pregon_for_init.i.us:
% local_id y.0.us = phi i64 [ %40, %pregon_for_end.i.us-icssa.us.us ], [ 0,
... %pregon_for_entry.pregon_for_init.i.us.preheader ]
%add6.i.i.us = add i64 % local_id y.0.us, %mul3.i.i, !llvm.access.group !12
%conv2.i.us = trunc i64 %add6.i.i.us to i32, !llvm.access.group !12
%mul.i.us = mul nsw i32 %conv2.i.us, %4, !llvm.access.group !12
%mul8.i.us = mul nsw i32 %conv2.i.us, %5
%30 = sext i32 %mul8.i.us to i64
br label %pregon_for_entry.entry.i.us.us
```

```
pregon_for_entry.pregon_for_init.i:
% local_id y.0 = phi i64 [ %63, %pregon_for_end.i ], [ 0,
... %pregon_for_entry.pregon_for_init.i.preheader ]
%41 = mul i64 % local_id y.0, %29
%42 = add i64 %28, %41
%43 = trunc i64 %42 to i32
%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
%min.iters.check = icmp ult i64 %umax, 32
br i1 %min.iters.check, label %pregon_for_entry.entry.i.preheader, label
... %vector.scevcheck
```

```
pregon_for_entry.entry.i.us.us:
% local_id x.0.us.us = phi i64 [ 0, %pregon_for_entry.pregon_for_init.i.us
... ], [ %32, %if.end.r_exit.i.loopexit.us.us ]
%add1.i.i.us.us = add i64 % local_id x.0.us.us, %mul.i.i, !llvm.access.group
... !12
%conv.i.us.us = trunc i64 %add1.i.i.us.us to i32, !llvm.access.group !12
%add.i.us.us = add nsw i32 %mul.i.us, %conv.i.us.us, !llvm.access.group !12
%idxprom.i.us.us = sext i32 %add.i.us.us to i64, !llvm.access.group !12
%arrayidx.i.us.us = getelementptr inbounds float, float* %0, i64
... %idxprom.i.us.us, !llvm.access.group !12
store float 0.000000e+00, float* %arrayidx.i.us.us, align 4, !tbaa !15,
... !llvm.access.group !12
%sext.i.us.us = shl i64 %add1.i.i.us.us, 32, !llvm.access.group !12
%31 = ashr exact i64 %sext.i.us.us, 32, !llvm.access.group !12
br label %for.body.i.us.us, !llvm.access.group !12
```

```
vector.scevcheck:
%44 = trunc i64 %22 to i32
%45 = add i32 %43, %44
%46 = icmp slt i32 %45, %43
%47 = icmp ugt i64 %22, 4294967295
%48 = or i1 %46, %47
br i1 %48, label %pregon_for_entry.entry.i.preheader, label %vector.ph
```

```
vector.ph:
%n.vec = and i64 %umax, -32
br label %vector.body
```

```
for.body.i.us.us:
%indvars.iv.next.i2.us.us = phi i64 [ %indvars.iv.next.i.us.us,
... %for.body.i.us.us ], [ 0, %pregon_for_entry.entry.i.us.us ]
%33 = phi float [ %39, %for.body.i.us.us ], [ 0.000000e+00,
... %pregon_for_entry.entry.i.us.us ]
%34 = add nsw i64 %indvars.iv.next.i2.us.us, %30, !llvm.access.group !12
%arrayidx11.i.us.us = getelementptr inbounds float, float* %1, i64 %34,
... !llvm.access.group !12
%35 = load float, float* %arrayidx11.i.us.us, align 4, !tbaa !15,
... !llvm.access.group !12
%mul12.i.us.us = fmul float %35, %7, !llvm.access.group !12
%36 = mul nsw i64 %indvars.iv.next.i2.us.us, %14, !llvm.access.group !12
%37 = add nsw i64 %36, %31, !llvm.access.group !12
%arrayidx16.i.us.us = getelementptr inbounds float, float* %2, i64 %37,
... !llvm.access.group !12
%38 = load float, float* %arrayidx16.i.us.us, align 4, !tbaa !15,
... !llvm.access.group !12
%39 = tail call @llvm.fmuladd.f32(float %mul12.i.us.us, float %38,
... float %33) #3, !llvm.access.group !12
store float %39, float* %arrayidx.i.us.us, align 4, !tbaa !15,
... !llvm.access.group !12
%indvars.iv.next.i.us.us = add nuw nsw i64 %indvars.iv.next.i2.us.us, 1,
... !llvm.access.group !12
%exitcond.not.i.us.us = icmp eq i64 %indvars.iv.next.i.us.us,
... %wide.trip.count.i, !llvm.access.group !12
br i1 %exitcond.not.i.us.us, label %if.end.r_exit.i.loopexit.us.us, label
... %for.body.i.us.us, !llvm.loop !21, !llvm.access.group !12
```

```
if.end.r_exit.i.loopexit.us.us:
%32 = add nuw i64 % local_id x.0.us.us, 1
%exitcond.not = icmp eq i64 %32, %umax
br i1 %exitcond.not, label %pregon_for_end.i.us-icssa.us.us, label
... %pregon_for_entry.entry.i.us.us, !llvm.loop !19
```

```
pregon_for_end.i.us-icssa.us.us:
%40 = add nuw i64 % local_id y.0.us, 1
%exitcond29.not = icmp eq i64 %40, %umax28
br i1 %exitcond29.not, label %mm2_kernel1.exit.loopexit, label
... %pregon_for_entry.pregon_for_init.i.us, !llvm.loop !23
```

```
vector.body:
%index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
%49 = add i64 %index, %mul.i.i, !llvm.access.group !12
%50 = trunc i64 %49 to i32, !llvm.access.group !12
%51 = add nsw i32 %mul.i, %50, !llvm.access.group !12
%52 = sext i32 %51 to i64, !llvm.access.group !12
%53 = getelementptr inbounds float, float* %0, i64 %52, !llvm.access.group
... !12
%54 = bitcast float* %53 to <8 x float>*
store <8 x float> zeroinitializer, <8 x float>* %54, align 4, !tbaa !15,
... !llvm.access.group !12
%55 = getelementptr inbounds float, float* %53, i64 8
%56 = bitcast float* %55 to <8 x float>*
store <8 x float> zeroinitializer, <8 x float>* %56, align 4, !tbaa !15,
... !llvm.access.group !12
%57 = getelementptr inbounds float, float* %53, i64 16
%58 = bitcast float* %57 to <8 x float>*
store <8 x float> zeroinitializer, <8 x float>* %58, align 4, !tbaa !15,
... !llvm.access.group !12
%59 = getelementptr inbounds float, float* %53, i64 24
%60 = bitcast float* %59 to <8 x float>*
store <8 x float> zeroinitializer, <8 x float>* %60, align 4, !tbaa !15,
... !llvm.access.group !12
%index.next = add i64 %index, 32
%61 = icmp eq i64 %index.next, %n.vec
br i1 %61, label %middle.block, label %vector.body, !llvm.loop !25
```

```
middle.block:
%cmp.n = icmp eq i64 %umax, %n.vec
br i1 %cmp.n, label %pregon_for_end.i, label
... %pregon_for_entry.entry.i.preheader
```

```
pregon_for_entry.entry.i.preheader:
% local_id x.0.ph = phi i64 [ 0, %vector.scevcheck ], [ 0,
... %pregon_for_entry.pregon_for_init.i ], [ %n.vec, %middle.block ]
br label %pregon_for_entry.entry.i
```

```
pregon_for_entry.entry.i:
% local_id x.0 = phi i64 [ %62, %pregon_for_entry.entry.i ], [
... % local_id x.0.ph, %pregon_for_entry.entry.i.preheader ]
%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* %0, i64 %idxprom.i,
... !llvm.access.group !12
store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !15,
... !llvm.access.group !12
%62 = add nuw i64 % local_id x.0, 1
%exitcond31.not = icmp eq i64 %62, %umax
br i1 %exitcond31.not, label %pregon_for_end.i.loopexit, label
... %pregon_for_entry.entry.i, !llvm.loop !27
```

```
pregon_for_end.i.loopexit:
br label %pregon_for_end.i
```

```
pregon_for_end.i:
%63 = add nuw i64 % local_id y.0, 1
%exitcond33.not = icmp eq i64 %63, %umax28
br i1 %exitcond33.not, label %mm2_kernel1.exit.loopexit37, label
... %pregon_for_entry.pregon_for_init.i, !llvm.loop !23
```

```
mm2_kernel1.exit.loopexit37:
br label %mm2_kernel1.exit
```

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

CFG for 'pool\_kernel\_mm2\_kernel1' function