

```
%13:
%14 = sext i32 %6 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
%cmp739.i = icmp sgt i32 %4, 0
%wide.trip.count.i = zext i32 %4 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 %cmp739.i, label %pregion_for_entry.pregion_for_init.i.us.preheader,
... label %pregion_for_entry.pregion_for_init.i.preheader
```

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

```
pregion_for_entry.pregion_for_init.i.us:
%local_id.y.0.us = phi i64 [ %41, %pregion_for_end.i.us-lcssa.us.us ], [ 0,
... %pregion_for_entry.pregion_for_init.i.us.preheader ]
%add6.i.i.us = add i64 %local_id.y.0.us, %mul3.i.i
%conv2.i.us = trunc i64 %add6.i.i.us to i32
%mul.i.us = mul nsw i32 %conv2.i.us, %6
%mul9.i.us = mul nsw i32 %conv2.i.us, %4
%30 = sext i32 %mul9.i.us to i64
br label %pregion_for_entry.entry.i.us.us
```

```
pregion_for_entry.pregion_for_init.i.preheader:
%22 = add nsw i64 %umax, -1
%23 = trunc i64 %11 to i32
%24 = mul i32 %23, %6
%25 = shl i32 %24, 3
%26 = zext i32 %25 to i64
%27 = shl i64 %10, 5
%28 = add i64 %27, %26
%29 = zext i32 %26 to i64
br label %pregion_for_entry.pregion_for_init.i
```

```
pregion_for_entry.pregion_for_init.i:
%local_id.y.0 = phi i64 [ %73, %pregion_for_end.i ], [ 0,
... %pregion_for_entry.pregion_for_init.i.preheader ]
%42 = mul i64 %local_id.y.0, %29
%43 = add i64 %28, %42
%44 = trunc i64 %43 to i32
%add6.i.i = add i64 %local_id.y.0, %mul3.i.i
%conv2.i = trunc i64 %add6.i.i to i32
%mul.i = mul nsw i32 %conv2.i, %6
%min.iters.check = icmp ult i64 %umax, 32
br i1 %min.iters.check, label %pregion_for_entry.entry.i.preheader, label
... %vector.scevcheck
```

```
pregion_for_entry.entry.i.us.us:
%local_id.x.0.us.us = phi i64 [ 0, %pregion_for_entry.pregion_for_init.i.us
... ], [ %33, %if.end.r_exit.i.loopexit.us.us ]
%add1.i.i.us.us = add i64 %local_id.x.0.us.us, %mul.i.i
%conv.i.us.us = trunc i64 %add1.i.i.us.us to i32
%add.i.us.us = add nsw i32 %mul.i.us, %conv.i.us.us
%idxprom.i.us.us = sext i32 %add.i.us.us to i64
%arrayidx.i.us.us = getelementptr inbounds float, float* %2, i64
... %idxprom.i.us.us
%31 = load float, float* %arrayidx.i.us.us, align 4, !tbaa !12
%mul6.i.us.us = fmul float %31, %8
store float %mul6.i.us.us, float* %arrayidx.i.us.us, align 4, !tbaa !12,
... !llvm.access.group !16
%sext.i.us.us = shl i64 %add1.i.i.us.us, 32
%32 = ashr exact i64 %sext.i.us.us, 32
br label %for.body.i.us.us
```

```
vector.scevcheck:
%45 = trunc i64 %22 to i32
%46 = add i32 %44, %45
%47 = icmp slt i32 %46, %44
%48 = icmp ugt i64 %22, 4294967295
%49 = or i1 %47, %48
br i1 %49, label %pregion_for_entry.entry.i.preheader, label %vector.ph
```

```
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, %pregion_for_entry.entry.i.us.us ]
%34 = phi float [ %40, %for.body.i.us.us ], [ %mul6.i.us.us,
... %pregion_for_entry.entry.i.us.us ]
%35 = add nsw i64 %indvars.iv.next.i2.us.us, %30
%arrayidx12.i.us.us = getelementptr inbounds float, float* %0, i64 %35
%36 = load float, float* %arrayidx12.i.us.us, align 4, !tbaa !12
%37 = mul nsw i64 %indvars.iv.next.i2.us.us, %14
%38 = add nsw i64 %37, %32
%arrayidx16.i.us.us = getelementptr inbounds float, float* %1, i64 %38
%39 = load float, float* %arrayidx16.i.us.us, align 4, !tbaa !12
%40 = tail call float @llvm.fmuladd.f32(float %36, float %39, float %34) #3
store float %40, float* %arrayidx.i.us.us, align 4, !tbaa !12,
... !llvm.access.group !16
%indvars.iv.next.i.us.us = add nuw nsw i64 %indvars.iv.next.i2.us.us, 1
%exitcond.not.i.us.us = icmp eq i64 %indvars.iv.next.i.us.us,
... %wide.trip.count.i
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
```

```
if.end.r_exit.i.loopexit.us.us:
%33 = add nuw i64 %local_id.x.0.us.us, 1
%exitcond.not = icmp eq i64 %33, %umax
br i1 %exitcond.not, label %pregion_for_end.i.us-lcssa.us.us, label
... %pregion_for_entry.entry.i.us.us, !llvm.loop !19
```

```
pregion_for_end.i.us-lcssa.us.us:
%41 = add nuw i64 %local_id.y.0.us, 1
%exitcond29.not = icmp eq i64 %41, %umax28
br i1 %exitcond29.not, label %mm2_kernel2.exit.loopexit, label
... %pregion_for_entry.pregion_for_init.i.us, !llvm.loop !23
```

```
vector.ph:
%n.vec = and i64 %umax, -32
%broadcast.splatinsert = insertelement <8 x float> undef, float %8, i32 0
%broadcast.splat = shufflevector <8 x float> %broadcast.splatinsert, <8 x
... float> undef, <8 x i32> zeroinitializer
%broadcast.splatinsert39 = insertelement <8 x float> undef, float %8, i32 0
%broadcast.splat40 = shufflevector <8 x float> %broadcast.splatinsert39, <8
... x float> undef, <8 x i32> zeroinitializer
%broadcast.splatinsert41 = insertelement <8 x float> undef, float %8, i32 0
%broadcast.splat42 = shufflevector <8 x float> %broadcast.splatinsert41, <8
... x float> undef, <8 x i32> zeroinitializer
%broadcast.splatinsert43 = insertelement <8 x float> undef, float %8, i32 0
%broadcast.splat44 = shufflevector <8 x float> %broadcast.splatinsert43, <8
... x float> undef, <8 x i32> zeroinitializer
br label %vector.body
```

```
vector.body:
%index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
%50 = add i64 %index, %mul.i.i
%51 = trunc i64 %50 to i32
%52 = add nsw i32 %mul.i, %51
%53 = sext i32 %52 to i64
%54 = getelementptr inbounds float, float* %2, i64 %53
%55 = bitcast float* %54 to <8 x float>*
%wide.load = load <8 x float>, <8 x float>* %55, align 4, !tbaa !12
%56 = getelementptr inbounds float, float* %54, i64 8
%57 = bitcast float* %56 to <8 x float>*
%wide.load36 = load <8 x float>, <8 x float>* %57, align 4, !tbaa !12
%58 = getelementptr inbounds float, float* %54, i64 16
%59 = bitcast float* %58 to <8 x float>*
%wide.load37 = load <8 x float>, <8 x float>* %59, align 4, !tbaa !12
%60 = getelementptr inbounds float, float* %54, i64 24
%61 = bitcast float* %60 to <8 x float>*
%wide.load38 = load <8 x float>, <8 x float>* %61, align 4, !tbaa !12
%62 = fmul <8 x float> %wide.load, %broadcast.splat
%63 = fmul <8 x float> %wide.load36, %broadcast.splat40
%64 = fmul <8 x float> %wide.load37, %broadcast.splat42
%65 = fmul <8 x float> %wide.load38, %broadcast.splat44
%66 = bitcast float* %54 to <8 x float>*
store <8 x float> %62, <8 x float>* %66, align 4, !tbaa !12,
... !llvm.access.group !16
%67 = bitcast float* %56 to <8 x float>*
store <8 x float> %63, <8 x float>* %67, align 4, !tbaa !12,
... !llvm.access.group !16
%68 = bitcast float* %58 to <8 x float>*
store <8 x float> %64, <8 x float>* %68, align 4, !tbaa !12,
... !llvm.access.group !16
%69 = bitcast float* %60 to <8 x float>*
store <8 x float> %65, <8 x float>* %69, align 4, !tbaa !12,
... !llvm.access.group !16
%index.next = add i64 %index, 32
%70 = icmp eq i64 %index.next, %n.vec
br i1 %70, label %middle.block, label %vector.body, !llvm.loop !25
```

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

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

```
pregion_for_entry.entry.i:
%local_id.x.0 = phi i64 [ %72, %pregion_for_entry.entry.i ], [
... %local_id.x.0.ph, %pregion_for_entry.entry.i.preheader ]
%add1.i.i = add i64 %local_id.x.0, %mul.i.i
%conv.i = trunc i64 %add1.i.i to i32
%add.i = add nsw i32 %mul.i, %conv.i
%idxprom.i = sext i32 %add.i to i64
%arrayidx.i = getelementptr inbounds float, float* %2, i64 %idxprom.i
%71 = load float, float* %arrayidx.i, align 4, !tbaa !12
%mul6.i = fmul float %71, %8
store float %mul6.i, float* %arrayidx.i, align 4, !tbaa !12,
... !llvm.access.group !16
%72 = add nuw i64 %local_id.x.0, 1
%exitcond31.not = icmp eq i64 %72, %umax
br i1 %exitcond31.not, label %pregion_for_end.i.loopexit, label
... %pregion_for_entry.entry.i, !llvm.loop !27
```

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

```
pregion_for_end.i:
%73 = add nuw i64 %local_id.y.0, 1
%exitcond33.not = icmp eq i64 %73, %umax28
br i1 %exitcond33.not, label %mm2_kernel2.exit.loopexit46, label
... %pregion_for_entry.pregion_for_init.i, !llvm.loop !23
```

```
mm2_kernel2.exit.loopexit46:
br label %mm2_kernel2.exit
```

```
mm2_kernel2.exit:
ret void
```

CFG for 'pocl_kernel_mm2_kernel2' function