

```
%10:
%11 = sext i32 %4 to i64
%12 = icmp slt i64 %11, 32
%13 = select i1 %12, i64 %11, i64 32
%14 = sext i32 %3 to i64
%15 = icmp slt i64 %14, 8
%16 = select i1 %15, i64 %14, i64 8
%mul1.i = shl i64 %7, 5
%mul3.i.i = shl i64 %8, 3
%cmp638.i = icmp sgt i32 %5, 0
%wide.trip.count.i = zext i32 %5 to i64
%17 = icmp ugt i64 %13, 1
%umax = select i1 %17, i64 %13, i64 1
%18 = icmp ugt i64 %16, 1
%umax27 = select i1 %18, i64 %16, i64 1
br i1 %cmp638.i, label %region_for_entry.region_for_init.i.us.preheader,
... label %region_for_entry.region_for_init.i.us.preheader
```

```
region_for_entry.region_for_init.i.us.preheader:
br label %region_for_entry.region_for_init.i.us
```

```
region_for_entry.region_for_init.i.us:
%local_id.y.0.us = phi i64 [ %37, %region_for_end.i.us-lcssa.us.us ], [ 0,
... %region_for_entry.region_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, %4
%mul8.i.us = mul nsw i32 %conv2.i.us, %5
%27 = sext i32 %mul8.i.us to i64
br label %region_for_entry.entry.i.us.us
```

```
region_for_entry.entry.i.us.us:
%local_id.x.0.us.us = phi i64 [ 0, %region_for_entry.region_for_init.i.us
... ], [ %29, %if.end.r_exit.i.loopexit.us.us ]
%add1.i.i.us.us = add i64 %local_id.x.0.us.us, %mul.i.i
%conv1.us.us = trunc i64 %add1.i.i.us.us to i32
%add.i.us.us = add nsw i32 %mul.i.us, %conv1.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
store float 0.000000e+00, 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
%28 = ashr exact i64 %sext.i.us.us, 32
br label %for.body.i.us.us
```

```
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, %region_for_entry.entry.i.us.us ]
%30 = phi float [ %36, %for.body.i.us.us ], [ 0.000000e+00,
... %region_for_entry.entry.i.us.us ]
%31 = add nsw i64 %indvars.iv.next.i2.us.us, %27
%arrayidx11.i.us.us = getelementptr inbounds float, float* %0, i64 %31
%32 = load float, float* %arrayidx11.i.us.us, align 4, !tbaa !12
%33 = mul nsw i64 %indvars.iv.next.i2.us.us, %11
%34 = add nsw i64 %33, %28
%arrayidx15.i.us.us = getelementptr inbounds float, float* %1, i64 %34
%35 = load float, float* %arrayidx15.i.us.us, align 4, !tbaa !12
%36 = tail call float @llvm.fmuladd.f32(float %32, float %35, float %30) #3
store float %36, 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:
%29 = add nuw i64 %local_id.x.0.us.us, 1
%exitcond.not = icmp eq i64 %29, %umax
br i1 %exitcond.not, label %region_for_end.i.us-lcssa.us.us, label
... %region_for_entry.entry.i.us.us, !llvm.loop !19
```

```
region_for_end.i.us-lcssa.us.us:
%37 = add nuw i64 %local_id.y.0.us, 1
%exitcond28.not = icmp eq i64 %37, %umax27
br i1 %exitcond28.not, label %mm3_kernel1.exit.loopexit, label
... %region_for_entry.region_for_init.i.us, !llvm.loop !23
```

```
mm3_kernel1.exit.loopexit:
br label %mm3_kernel1.exit
```

```
mm3_kernel1.exit:
ret void
```

```
region_for_entry.region_for_init.i.preheader:
%19 = add nsw i64 %umax, -1
%20 = trunc i64 %8 to i32
%21 = mul i32 %20, %4
%22 = shl i32 %21, 3
%23 = zext i32 %22 to i64
%24 = shl i64 %7, 5
%25 = add i64 %24, %23
%26 = zext i32 %4 to i64
br label %region_for_entry.region_for_init.i
```

```
region_for_entry.region_for_init.i:
%local_id.y.0 = phi i64 [ %60, %region_for_end.i ], [ 0,
... %region_for_entry.region_for_init.i.preheader ]
%38 = mul i64 %local_id.y.0, %26
%39 = add i64 %25, %38
%40 = trunc i64 %39 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, %4
%min.itors.check = icmp ult i64 %umax, 32
br i1 %min.itors.check, label %region_for_entry.entry.i.preheader, label
... %vector.scevcheck
```

```
vector.scevcheck:
%41 = trunc i64 %19 to i32
%42 = add i32 %40, %41
%43 = icmp slt i32 %42, %40
%44 = icmp ugt i64 %19, 4294967295
%45 = or i1 %43, %44
br i1 %45, label %region_for_entry.entry.i.preheader, label %vector.ph
```

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

```
vector.body:
%index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
%46 = add i64 %index, %mul.i.i
%47 = trunc i64 %46 to i32
%48 = add nsw i32 %mul.i, %47
%49 = sext i32 %48 to i64
%50 = getelementptr inbounds float, float* %2, i64 %49
%51 = bitcast float* %50 to <8 x float>*
store <8 x float> zeroinitializer, <8 x float>* %51, align 4, !tbaa !12,
... !llvm.access.group !16
%52 = getelementptr inbounds float, float* %50, i64 8
%53 = bitcast float* %52 to <8 x float>*
store <8 x float> zeroinitializer, <8 x float>* %53, align 4, !tbaa !12,
... !llvm.access.group !16
%54 = getelementptr inbounds float, float* %50, i64 16
%55 = bitcast float* %54 to <8 x float>*
store <8 x float> zeroinitializer, <8 x float>* %55, align 4, !tbaa !12,
... !llvm.access.group !16
%56 = getelementptr inbounds float, float* %50, i64 24
%57 = bitcast float* %56 to <8 x float>*
store <8 x float> zeroinitializer, <8 x float>* %57, align 4, !tbaa !12,
... !llvm.access.group !16
%index.next = add i64 %index, 32
%58 = icmp eq i64 %index.next, %n.vec
br i1 %58, label %middle.block, label %vector.body, !llvm.loop !25
```

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

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

```
region_for_entry.entry.i:
%local_id.x.0 = phi i64 [ %59, %region_for_entry.entry.i ], [
... %local_id.x.0.ph, %region_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
store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !12,
... !llvm.access.group !16
%59 = add nuw i64 %local_id.x.0, 1
%exitcond30.not = icmp eq i64 %59, %umax
br i1 %exitcond30.not, label %region_for_end.i.loopexit, label
... %region_for_entry.entry.i, !llvm.loop !27
```

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

```
region_for_end.i:
%60 = add nuw i64 %local_id.y.0, 1
%exitcond32.not = icmp eq i64 %60, %umax27
br i1 %exitcond32.not, label %mm3_kernel1.exit.loopexit36, label
... %region_for_entry.region_for_init.i, !llvm.loop !23
```

```
mm3_kernel1.exit.loopexit36:
br label %mm3_kernel1.exit
```

CFG for 'pocl_kernel_mm3_kernel1' function