T
F
T
F
T
F
T
F
T
F

```
vector.scevcheck:
 %mul.i.i = shl i64 %7, 8
 %mul2.i = mul nsw i32 %5, %3
 %add3.i = add nsw i32 %mul2.i, %3
 %idxprom4.i = sext i32 %add3.i to i64
 %arrayidx5.i = getelementptr inbounds float, float* %1, i64 %idxprom4.i
 %ident.check = icmp ne i32 %5, 1
 %10 = trunc i64 %7 to i32
 %11 = shl i32 %10, 8
 %12 = add i32 %11, %3
 %13 = icmp sgt i32 %12, 2147483392
 %14 = or i1 %ident.check, %13
 br i1 %14, label %pregion_for_entry.entry.i.preheader, label %vector.memcheck
```

```
vector.memcheck:
 %scevgep = getelementptr float, float* %1, i64 %idxprom4.i
 %scevgep1 = bitcast float* %scevgep to i8*
 %uglygep = getelementptr i8, i8* %scevgep1, i64 1
 %15 = trunc i64 %7 to i32
 %16 = shl i32 %15, 8
 %17 = add i32 %16, %3
 %18 = sext i32 %17 to i64
 %scevgep2 = getelementptr float, float* %2, i64 %18
 %scevgep23 = bitcast float* %scevgep2 to i8*
 %19 = add nsw i64 %18, 256
 %scevgep4 = getelementptr float, float* %2, i64 %19
 %scevgep6 = getelementptr float, float* %0, i64 %18
 %scevgep8 = getelementptr float, float* %0, i64 %19
 %bound0 = icmp ult float* %arrayidx5.i, %scevgep4
 %bound1 = icmp ugt i8* %uglygep, %scevgep23
 %found.conflict = and i1 %bound0, %bound1
 %bound010 = icmp ult float* %scevgep2, %scevgep8
 %bound111 = icmp ult float* %scevgep6, %scevgep4
 %found.conflict12 = and i1 %bound010, %bound111
 %conflict.rdx = or i1 %found.conflict, %found.conflict12
 br i1 %conflict.rdx, label %pregion_for_entry.entry.i.preheader, label
... %vector.ph
```

```
vector.ph:
 %broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
 %broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
... undef, <8 x i32> zeroinitializer
 %broadcast.splatinsert13 = insertelement <8 x i32> undef, i32 %4, i32 0
 %broadcast.splat14 = shufflevector <8 x i32> %broadcast.splatinsert13, <8 x
... i32> undef, <8 x i32> zeroinitializer
 %broadcast.splatinsert15 = insertelement <8 x float*> undef, float*
... %arrayidx5.i, i32 0
 %broadcast.splat16 = shufflevector <8 x float*> %broadcast.splatinsert15, <8
... x float*> undef, <8 x i32> zeroinitializer
 br label %vector.body
```

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

```
vector.body:
 %index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
 %vec.ind = phi <8 x i64> [ <i64 0, i64 1, i64 2, i64 3, i64 4, i64 5, i64 6,
... i64 7>, %vector.ph ], [ %vec.ind.next, %vector.body ]
 %20 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
 %21 = trunc <8 x i64> %20 to <8 x i32>
 %22 = icmp sgt <8 x i32> %broadcast.splat14, %21
 %23 = extractelement <8 x i32> %21, i32 0
 %24 = mul nsw i32 %23, %5
 %25 = add nsw i32 %24, %3
 %26 = sext i32 %25 to i64
 %27 = getelementptr inbounds float, float* %0, i64 %26
 %28 = bitcast float* %27 to <8 x float>*
 %wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %28, i32 4, <8 x i1> %22, <8 x float> undef), !tbaa !12, !alias.scope
... !16
 %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
... x float*> %broadcast.splat16, i32 4, <8 x i1> %22, <8 x float> undef), !tbaa
... !12, !alias.scope !19, !noalias !21
 %29 = fdiv <8 x float> %wide.masked.load, %wide.masked.gather, !fpmath !23
 %30 = getelementptr inbounds float, float* %2, i64 %26
 %31 = bitcast float* %30 to <8 x float>*
 call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %29, <8 x float>*
... %31, i32 4, <8 x i1> %22), !tbaa !12, !alias.scope !21, !noalias !16,
... !llvm.access.group !24
 %index.next = add i64 %index, 8
 %vec.ind.next = add <8 x i64> %vec.ind, <i64 8, i64 8, i64 8, i64 8, i64 8,
... i64 8, i64 8, i64 8>
 %32 = icmp eq i64 %index.next, 256
 br i1 %32, label %gramschmidt_kernel2.exit.loopexit18, label %vector.body,
... !llvm.loop !26
```

```
pregion_for_entry.entry.i:
 %_local_id_x.0 = phi i64 [ %35, %if.end.r_exit.i ], [ 0,
... %pregion_for_entry.entry.i.preheader ]
 %add1.i.i = add nuw nsw i64 %_local_id_x.0, %mul.i.i
 %conv.i = trunc i64 %add1.i.i to i32
 %cmp.i = icmp slt i32 %conv.i, %4
 br i1 %cmp.i, label %if.then.i, label %if.end.r_exit.i
```

```
if.then.i:
 %mul.i = mul nsw i32 %conv.i, %5
 %add.i = add nsw i32 %mul.i, %3
 %idxprom.i = sext i32 %add.i to i64
 %arrayidx.i = getelementptr inbounds float, float* %0, i64 %idxprom.i
 %33 = load float, float* %arrayidx.i, align 4, !tbaa !12
 %34 = load float, float* %arrayidx5.i, align 4, !tbaa !12
 %div.i = fdiv float %33, %34, !fpmath !23
 %arrayidx9.i = getelementptr inbounds float, float* %2, i64 %idxprom.i
 store float %div.i, float* %arrayidx9.i, align 4, !tbaa !12,
... !llvm.access.group !24
 br label %if.end.r_exit.i
```

```
if.end.r_exit.i:
 %35 = add nuw nsw i64 %_local_id_x.0, 1
 %exitcond.not = icmp eq i64 %35, 256
 br i1 %exitcond.not, label %gramschmidt_kernel2.exit.loopexit, label
... %pregion_for_entry.entry.i, !llvm.loop !29
```

```
gramschmidt_kernel2.exit.loopexit18:
 br label %gramschmidt_kernel2.exit
```

```
gramschmidt_kernel2.exit.loopexit:
 br label %gramschmidt_kernel2.exit
```

```
gramschmidt_kernel2.exit:
 ret void
```

CFG for '_pocl_kernel_gramschmidt_kernel2' function