

```
%10:  
%11 = sext i32 %4 to i64  
%12 = icmp slt i64 %11, 32  
%13 = select i1 %12, i64 %11, i64 32  
%14 = sext i32 %5 to i64  
%15 = icmp slt i64 %14, 8  
%16 = select i1 %15, i64 %14, i64 8  
%mul.i.i = shl i64 %7, 5  
%mul3.i.i = shl i64 %8, 3  
%17 = tail call float @llvm.sqrt.f32(float %3) #3  
%18 = icmp ugt i64 %13, 1  
%umax = select i1 %18, i64 %13, i64 1  
%19 = icmp ugt i64 %16, 1  
%umax1 = select i1 %19, i64 %16, i64 1  
%20 = add nsw i64 %umax, -1  
%21 = trunc i64 %8 to i32  
%22 = mul i32 %21, %4  
%23 = shl i32 %22, 3  
%24 = zext i32 %23 to i64  
%25 = trunc i64 %7 to i32  
%26 = shl i32 %25, 5  
%27 = zext i32 %26 to i64  
%28 = add nuw nsw i64 %24, %27  
%29 = zext i32 %4 to i64  
%30 = trunc i64 %8 to i32  
%31 = mul i32 %30, %4  
%32 = shl i32 %31, 3  
%33 = zext i32 %32 to i64  
%34 = trunc i64 %7 to i32  
%35 = shl i32 %34, 5  
%36 = sext i32 %35 to i64  
%37 = add nsw i64 %33, %36  
%38 = zext i32 %4 to i64  
%scevgep7 = getelementptr float, float* %2, i64 %umax  
%scevgep10 = getelementptr float, float* %0, i64 %36  
%39 = add nsw i64 %umax, %36  
%scevgep12 = getelementptr float, float* %0, i64 %39  
%scevgep14 = getelementptr float, float* %1, i64 %36  
%scevgep16 = getelementptr float, float* %1, i64 %39  
br label %pregon_for_entry.pregon_for_init.i
```

```
pregon_for_entry.pregon_for_init.i:  
%local_id.y.0 = phi i64 [ 0, %10 ], [ %79, %pregon_for_end.i ]  
%40 = mul i64 %local_id.y.0, %38  
%41 = add i64 %37, %40  
%sext = shl i64 %41, 32  
%42 = ashr exact i64 %sext, 32  
%scevgep = getelementptr float, float* %2, i64 %42  
%scevgep8 = getelementptr float, float* %scevgep7, i64 %42  
%43 = mul i64 %local_id.y.0, %29  
%44 = add i64 %28, %43  
%45 = trunc i64 %44 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.iters.check = icmp ult i64 %umax, 8  
br i1 %min.iters.check, label %pregon_for_entry.entry.i.preheader, label  
... %vector.scevcheck
```

```
vector.scevcheck:  
%46 = trunc i64 %20 to i32  
%47 = add i32 %45, %46  
%48 = icmp slt i32 %47, %45  
%49 = icmp ugt i64 %20, 4294967295  
%50 = or i1 %48, %49  
%51 = trunc i64 %20 to i32  
%52 = add i32 %26, %51  
%53 = icmp slt i32 %52, %26  
%54 = icmp ugt i64 %20, 4294967295  
%55 = or i1 %53, %54  
%56 = or i1 %50, %55  
br i1 %56, label %pregon_for_entry.entry.i.preheader, label %vector.memcheck
```

```
vector.memcheck:  
%bound0 = icmp ult float* %scevgep, %scevgep12  
%bound1 = icmp ult float* %scevgep10, %scevgep8  
%found.conflict = and i1 %bound0, %bound1  
%bound018 = icmp ult float* %scevgep, %scevgep16  
%bound119 = icmp ult float* %scevgep14, %scevgep8  
%found.conflict20 = and i1 %bound018, %bound119  
%conflict.rdx = or i1 %found.conflict, %found.conflict20  
br i1 %conflict.rdx, label %pregon_for_entry.entry.i.preheader, label  
... %vector.ph
```

```
vector.ph:  
%n.vec = and i64 %umax, -8  
%broadcast.splatinsert = insertelement <8 x float> undef, float %17, i32 0  
%broadcast.splat = shufflevector <8 x float> %broadcast.splatinsert, <8 x  
... float> undef, <8 x i32> zeroinitializer  
br label %vector.body
```

```
vector.body:  
%index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]  
%57 = add i64 %index, %mul.i.i  
%58 = trunc i64 %57 to i32  
%59 = shl i64 %57, 32  
%60 = ashr exact i64 %59, 32  
%61 = getelementptr inbounds float, float* %0, i64 %60  
%62 = bitcast float* %61 to <8 x float>*  
%wide.load = load <8 x float>, <8 x float>* %62, align 4, !tbaa !12,  
... !alias.scope !16  
%63 = add nsw i32 %mul.i, %58  
%64 = sext i32 %63 to i64  
%65 = getelementptr inbounds float, float* %2, i64 %64  
%66 = bitcast float* %65 to <8 x float>*  
%wide.load21 = load <8 x float>, <8 x float>* %66, align 4, !tbaa !12,  
... !alias.scope !19, !noalias !21  
%67 = fsub <8 x float> %wide.load21, %wide.load  
%68 = bitcast float* %65 to <8 x float>*  
store <8 x float> %67, <8 x float>* %68, align 4, !tbaa !12, !alias.scope  
... !19, !noalias !21, !llvm.access.group !23  
%69 = getelementptr inbounds float, float* %1, i64 %60  
%70 = bitcast float* %69 to <8 x float>*  
%wide.load22 = load <8 x float>, <8 x float>* %70, align 4, !tbaa !12,  
... !alias.scope !26  
%71 = fmul <8 x float> %broadcast.splat, %wide.load22  
%72 = fdiv <8 x float> %67, %71, !fpmath !27  
%73 = bitcast float* %65 to <8 x float>*  
store <8 x float> %72, <8 x float>* %73, align 4, !tbaa !12, !alias.scope  
... !19, !noalias !21, !llvm.access.group !23  
%index.next = add i64 %index, 8  
%74 = icmp eq i64 %index.next, %n.vec  
br i1 %74, label %middle.block, label %vector.body, !llvm.loop !28
```

```
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.memcheck ], [ 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 [ %78, %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  
%conv.i = trunc i64 %add1.i.i to i32  
%sext.i = shl i64 %add1.i.i, 32  
%idxprom.i = ashr exact i64 %sext.i, 32  
%arrayidx.i = getelementptr inbounds float, float* %0, i64 %idxprom.i  
%75 = load float, float* %arrayidx.i, align 4, !tbaa !12  
%add.i = add nsw i32 %mul.i, %conv.i  
%idxprom6.i = sext i32 %add.i to i64  
%arrayidx7.i = getelementptr inbounds float, float* %2, i64 %idxprom6.i  
%76 = load float, float* %arrayidx7.i, align 4, !tbaa !12  
%sub.i = fsub float %76, %75  
store float %sub.i, float* %arrayidx7.i, align 4, !tbaa !12,  
... !llvm.access.group !23  
%arrayidx10.i = getelementptr inbounds float, float* %1, i64 %idxprom.i  
%77 = load float, float* %arrayidx10.i, align 4, !tbaa !12  
%mul11.i = fmul float %17, %77  
%div.i = fdiv float %sub.i, %mul11.i, !fpmath !27  
store float %div.i, float* %arrayidx7.i, align 4, !tbaa !12,  
... !llvm.access.group !23  
%78 = add nuw i64 %local_id.x.0, 1  
%exitcond.not = icmp eq i64 %78, %umax  
br i1 %exitcond.not, label %pregon_for_end.i.loopexit, label  
... %pregon_for_entry.entry.i, !llvm.loop !31
```

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

```
pregon_for_end.i:  
%79 = add nuw i64 %local_id.y.0, 1  
%exitcond2.not = icmp eq i64 %79, %umax1  
br i1 %exitcond2.not, label %reduce_kernel.exit, label  
... %pregon_for_entry.pregon_for_init.i, !llvm.loop !32
```

```
reduce_kernel.exit:  
ret void
```

CFG for ' pocl\_kernel\_reduce\_kernel' function