

vector.scevcheck:
%mul.i.i = shl i64 %5, 8
%sub.i = add nsw i32 %3, -1
%mul.i = mul nsw i32 %sub.i, %3
%8 = add i32 %3, -1
%9 = mul i32 %8, %3
%10 = trunc i64 %5 to i32
%11 = shl i32 %10, 8
%12 = add i32 %9, %11
%13 = icmp sgt i32 %12, 2147483392
br i1 %13, label %preion_for_entry.entry.i.preheader, label %vector.memcheck

T	F
---	---

vector.memcheck:
%14 = add i32 %3, -1
%15 = mul i32 %14, %3
%16 = trunc i64 %5 to i32
%17 = shl i32 %16, 8
%18 = add i32 %15, %17
%19 = sext i32 %18 to i64
%scevgep = getelementptr float, float* %2, i64 %19
%20 = add nsw i64 %19, 256
%scevgep2 = getelementptr float, float* %2, i64 %20
%scevgep4 = getelementptr float, float* %1, i64 %19
%scevgep6 = getelementptr float, float* %1, i64 %20
%bound0 = icmp ult float* %scevgep, %scevgep6
%bound1 = icmp ult float* %scevgep4, %scevgep2
%found.conflict = and i1 %bound0, %bound1
br i1 %found.conflict, label %preion_for_entry.entry.i.preheader, label ... %vector.ph

T	F
---	---

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.splatinsert8 = insertelement <8 x i32> undef, i32 %3, i32 0
%broadcast.splat9 = shufflevector <8 x i32> %broadcast.splatinsert8, <8 x
... i32> undef, <8 x i32> zeroinitializer
br label %vector.body

preion_for_entry.entry.i.preheader:
br label %preion_for_entry.entry.i

vector.body:
%index = phi i64 [0, %vector.ph], [%index.next.1, %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.1, %vector.body]
%21 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
%22 = trunc <8 x i64> %21 to <8 x i32>
%23 = icmp sgt <8 x i32> %broadcast.splat9, %22
%24 = extractelement <8 x i32> %22, i32 0
%25 = add nsw i32 %mul.i, %24
%26 = sext i32 %25 to i64
%27 = getelementptr inbounds float, float* %2, 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> %23, <8 x float> undef), !tbaa !12, !alias.scope
... !16, !noalias !19
%29 = getelementptr inbounds float, float* %1, i64 %26
%30 = bitcast float* %29 to <8 x float>*
%wide.masked.load10 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %30, i32 4, <8 x i1> %23, <8 x float> undef), !tbaa !12, !alias.scope
... !19
%31 = fdiv <8 x float> %wide.masked.load, %wide.masked.load10, !fpmath !21
%32 = bitcast float* %27 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %31, <8 x float>*
... %32, i32 4, <8 x i1> %23), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !22
%vec.ind.next = add <8 x i64> %vec.ind, <i64 8, i64 8, i64 8, i64 8,
... i64 8, i64 8, i64 8>
%33 = add nuw nsw <8 x i64> %vec.ind.next, %broadcast.splat
%34 = trunc <8 x i64> %33 to <8 x i32>
%35 = icmp sgt <8 x i32> %broadcast.splat9, %34
%36 = extractelement <8 x i32> %34, i32 0
%37 = add nsw i32 %mul.i, %36
%38 = sext i32 %37 to i64
%39 = getelementptr inbounds float, float* %2, i64 %38
%40 = bitcast float* %39 to <8 x float>*
%wide.masked.load.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %40, i32 4, <8 x i1> %35, <8 x float> undef), !tbaa !12, !alias.scope
... !16, !noalias !19
%41 = getelementptr inbounds float, float* %1, i64 %38
%42 = bitcast float* %41 to <8 x float>*
%wide.masked.load10.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %42, i32 4, <8 x i1> %35, <8 x float> undef), !tbaa !12,
... !alias.scope !19
%43 = fdiv <8 x float> %wide.masked.load.1, %wide.masked.load10.1, !fpmath
... !21
%44 = bitcast float* %39 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %43, <8 x float>*
... %44, i32 4, <8 x i1> %35), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !22
%index.next.1 = add nuw nsw i64 %index, 16
%vec.ind.next.1 = add <8 x i64> %vec.ind, <i64 16, i64 16, i64 16, i64 16,
... i64 16, i64 16, i64 16, i64 16>
%45 = icmp eq i64 %index.next.1, 256
br i1 %45, label %adi_kernel5.exit.loopexit12, label %vector.body,
... !llvm.loop !24

T	F
---	---

preion_for_entry.entry.i:
% local_id_x.0 = phi i64 [0, %preion_for_entry.entry.i.preheader], [%51,
... %if.end.r_exit.i.1]
%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, %3
br i1 %cmp.i, label %if.then.i, label %if.end.r_exit.i

T	F
---	---

if.then.i:
%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
%46 = load float, float* %arrayidx.i, align 4, !tbaa !12
%arrayidx6.i = getelementptr inbounds float, float* %1, i64 %idxprom.i
%47 = load float, float* %arrayidx6.i, align 4, !tbaa !12
%div.i = fdiv float %46, %47, !fpmath !21
store float %div.i, float* %arrayidx.i, align 4, !tbaa !12,
... !llvm.access.group !22
br label %if.end.r_exit.i

if.end.r_exit.i:
%48 = or i64 % local_id_x.0, 1
%add1.i.i.1 = add nuw nsw i64 %48, %mul.i.i
%conv.i.1 = trunc i64 %add1.i.i.1 to i32
%cmp.i.1 = icmp slt i32 %conv.i.1, %3
br i1 %cmp.i.1, label %if.then.i.1, label %if.end.r_exit.i.1

T	F
---	---

if.then.i.1:
%add.i.1 = add nsw i32 %mul.i, %conv.i.1
%idxprom.i.1 = sext i32 %add.i.1 to i64
%arrayidx.i.1 = getelementptr inbounds float, float* %2, i64 %idxprom.i.1
%49 = load float, float* %arrayidx.i.1, align 4, !tbaa !12
%arrayidx6.i.1 = getelementptr inbounds float, float* %1, i64 %idxprom.i.1
%50 = load float, float* %arrayidx6.i.1, align 4, !tbaa !12
%div.i.1 = fdiv float %49, %50, !fpmath !21
store float %div.i.1, float* %arrayidx.i.1, align 4, !tbaa !12,
... !llvm.access.group !22
br label %if.end.r_exit.i.1

if.end.r_exit.i.1:
%51 = add nuw nsw i64 % local_id_x.0, 2
%exitcond.not.1 = icmp eq i64 %51, 256
br i1 %exitcond.not.1, label %adi_kernel5.exit.loopexit, label
... %preion_for_entry.entry.i, !llvm.loop !27

T	F
---	---

adi_kernel5.exit.loopexit:
br label %adi_kernel5.exit

adi_kernel5.exit:
ref void

CFG for '_pocl_kernel_adi_kernel5' function