

vector.scevcheck:  
%8 = shl i64 %5, 8  
%sub.i = add nsw i32 %3, -1  
%mul.i = mul nsw i32 %sub.i, %3  
%9 = add i32 %3, -1  
%10 = mul i32 %9, %3  
%11 = trunc i64 %5 to i32  
%12 = shl i32 %11, 8  
%13 = add i32 %10, %12  
%14 = icmp sgt i32 %13, 2147483392  
br i1 %14, label %pregon\_for\_entry.entry.i.preheader, label %vector.memcheck

T	F
---	---

vector.memcheck:  
%15 = add i32 %3, -1  
%16 = mul i32 %15, %3  
%17 = trunc i64 %5 to i32  
%18 = shl i32 %17, 8  
%19 = add i32 %16, %18  
%20 = sext i32 %19 to i64  
%scevgep = getelementptr float, float\* %2, i64 %20  
%21 = add nsw i64 %20, 256  
%scevgep2 = getelementptr float, float\* %2, i64 %21  
%scevgep4 = getelementptr float, float\* %1, i64 %20  
%scevgep6 = getelementptr float, float\* %1, i64 %21  
%bound0 = icmp ult float\* %scevgep, %scevgep6  
%bound1 = icmp ult float\* %scevgep4, %scevgep2  
%found.conflict = and i1 %bound0, %bound1  
br i1 %found.conflict, label %pregon\_for\_entry.entry.i.preheader, label ... %vector.ph

T	F
---	---

vector.ph:  
%broadcast.splatinsert = insertelement <8 x i64> undef, i64 %8, 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

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 ]  
%22 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat  
%23 = trunc <8 x i64> %22 to <8 x i32>  
%24 = icmp sgt <8 x i32> %broadcast.splat9, %23  
%25 = extractelement <8 x i32> %23, i32 0  
%26 = add nsw i32 %mul.i, %25  
%27 = sext i32 %26 to i64  
%28 = getelementptr inbounds float, float\* %2, i64 %27  
%29 = bitcast float\* %28 to <8 x float>\*  
%wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x  
... float>\* %29, i32 4, <8 x i1> %24, <8 x float> undef), !tbaa !12, !alias.scope  
... !16, !noalias !19  
%30 = getelementptr inbounds float, float\* %1, i64 %27  
%31 = bitcast float\* %30 to <8 x float>\*  
%wide.masked.load10 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x  
... float>\* %31, i32 4, <8 x i1> %24, <8 x float> undef), !tbaa !12, !alias.scope  
... !19  
%32 = fdiv <8 x float> %wide.masked.load, %wide.masked.load10, !fpmath !21  
%33 = bitcast float\* %28 to <8 x float>\*  
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %32, <8 x float>\*  
... %33, i32 4, <8 x i1> %24), !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, i64 8>  
%34 = add nuw nsw <8 x i64> %vec.ind.next, %broadcast.splat  
%35 = trunc <8 x i64> %34 to <8 x i32>  
%36 = icmp sgt <8 x i32> %broadcast.splat9, %35  
%37 = extractelement <8 x i32> %35, i32 0  
%38 = add nsw i32 %mul.i, %37  
%39 = sext i32 %38 to i64  
%40 = getelementptr inbounds float, float\* %2, i64 %39  
%41 = bitcast float\* %40 to <8 x float>\*  
%wide.masked.load.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x  
... float>\* %41, i32 4, <8 x i1> %36, <8 x float> undef), !tbaa !12, !alias.scope  
... !16, !noalias !19  
%42 = getelementptr inbounds float, float\* %1, i64 %39  
%43 = bitcast float\* %42 to <8 x float>\*  
%wide.masked.load10.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8  
... x float>\* %43, i32 4, <8 x i1> %36, <8 x float> undef), !tbaa !12,  
... !alias.scope !19  
%44 = fdiv <8 x float> %wide.masked.load.1, %wide.masked.load10.1, !fpmath  
... !21  
%45 = bitcast float\* %40 to <8 x float>\*  
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %44, <8 x float>\*  
... %45, i32 4, <8 x i1> %36), !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>  
%46 = icmp eq i64 %index.next.1, 256  
br i1 %46, label %adi\_kernel5.exit.loopexit12, label %vector.body,  
... !llvm.loop !24

T	F
---	---

pregon\_for\_entry.entry.i.preheader:  
br label %pregon\_for\_entry.entry.i

pregon\_for\_entry.entry.i:  
%\_local\_id\_x.0 = phi i64 [ 0, %pregon\_for\_entry.entry.i.preheader ], [ %54,  
... %if.end\_r\_exit.i.1 ]  
%47 = add nuw nsw i64 %\_local\_id\_x.0, %8  
%conv.i = trunc i64 %47 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  
%48 = load float, float\* %arrayidx.i, align 4, !tbaa !12  
%arrayidx6.i = getelementptr inbounds float, float\* %1, i64 %idxprom.i  
%49 = load float, float\* %arrayidx6.i, align 4, !tbaa !12  
%div.i = fdiv float %48, %49, !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:  
%50 = or i64 %\_local\_id\_x.0, 1  
%51 = add nuw nsw i64 %50, %8  
%conv.i.1 = trunc i64 %51 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  
%52 = load float, float\* %arrayidx.i.1, align 4, !tbaa !12  
%arrayidx6.i.1 = getelementptr inbounds float, float\* %1, i64 %idxprom.i.1  
%53 = load float, float\* %arrayidx6.i.1, align 4, !tbaa !12  
%div.i.1 = fdiv float %52, %53, !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:  
%54 = add nuw nsw i64 %\_local\_id\_x.0, 2  
%exitcond.1 = icmp eq i64 %54, 256  
br i1 %exitcond.1, label %adi\_kernel5.exit.loopexit, label  
... %pregon\_for\_entry.entry.i.llvm.loop !27

T	F
---	---

adi\_kernel5.exit.loopexit:  
br label %adi\_kernel5.exit

adi\_kernel5.exit:  
ret void

CFG for ' \_pocl\_kernel\_adi\_kernel5' function