

vector.scevcheck:  
%mul.i.i = shl i64 %6, 8  
%sub.i = sub i32 -2, %3  
%sub2.i = add i32 %sub.i, %4  
%mul.i = mul nsw i32 %sub2.i, %4  
%sub3.i = sub i32 -3, %3  
%sub4.i = add i32 %sub3.i, %4  
%mul5.i = mul nsw i32 %sub4.i, %4  
%9 = mul i32 %sub2.i, %4  
%10 = trunc i64 %6 to i32  
%11 = shl i32 %10, 8  
%12 = add i32 %9, %11  
%13 = icmp sgt i32 %12, 2147483392  
%14 = mul i32 %sub4.i, %4  
%15 = add i32 %14, %11  
%16 = icmp sgt i32 %15, 2147483392  
%17 = or i1 %13, %16  
br i1 %17, label %preregion\_for\_entry.entry.i.preheader, label %vector.memcheck

T	F
---	---

vector.memcheck:  
%18 = mul i32 %sub2.i, %4  
%19 = trunc i64 %6 to i32  
%20 = shl i32 %19, 8  
%21 = add i32 %18, %20  
%22 = sext i32 %21 to i64  
%scevgep = getelementptr float, float\* %2, i64 %22  
%23 = add nsw i64 %22, 256  
%scevgep5 = getelementptr float, float\* %2, i64 %23  
%24 = mul i32 %sub4.i, %4  
%25 = add i32 %24, %20  
%26 = sext i32 %25 to i64  
%scevgep7 = getelementptr float, float\* %2, i64 %26  
%27 = add nsw i64 %26, 256  
%scevgep9 = getelementptr float, float\* %2, i64 %27  
%scevgep11 = getelementptr float, float\* %0, i64 %26  
%scevgep13 = getelementptr float, float\* %0, i64 %27  
%scevgep15 = getelementptr float, float\* %1, i64 %22  
%scevgep17 = getelementptr float, float\* %1, i64 %23  
%bound0 = icmp ult float\* %scevgep, %scevgep9  
%bound1 = icmp ult float\* %scevgep7, %scevgep5  
%found.conflict = and i1 %bound0, %bound1  
%bound019 = icmp ult float\* %scevgep, %scevgep13  
%bound120 = icmp ult float\* %scevgep11, %scevgep5  
%found.conflict21 = and i1 %bound019, %bound120  
%conflict.rdx = or i1 %found.conflict, %found.conflict21  
%bound022 = icmp ult float\* %scevgep, %scevgep17  
%bound123 = icmp ult float\* %scevgep15, %scevgep5  
%found.conflict24 = and i1 %bound022, %bound123  
%conflict.rdx25 = or i1 %conflict.rdx, %found.conflict24  
br i1 %conflict.rdx25, label %preregion\_for\_entry.entry.i.preheader, label ... %vector.ph

T	F
---	---

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

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.splatinsert26 = insertelement <8 x i32> undef, i32 %4, i32 0  
%broadcast.splat27 = shufflevector <8 x i32> %broadcast.splatinsert26, <8 x ... i32> undef, <8 x i32> zeroinitializer  
br label %vector.body

preregion\_for\_entry.entry.i:  
% local\_id x.0 = phi i64 [ 0, %preregion\_for\_entry.entry.i.preheader ], [ %60, ... %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, %4  
br i1 %cmp.i, label %if.then.i, label %if.end\_r\_exit.i

T	F
---	---

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 ]  
%28 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat  
%29 = trunc <8 x i64> %28 to <8 x i32>  
%30 = icmp sgt <8 x i32> %broadcast.splat27, %29  
%31 = extractelement <8 x i32> %29, i32 0  
%32 = add nsw i32 %mul.i, %31  
%33 = sext i32 %32 to i64  
%34 = getelementptr inbounds float, float\* %2, i64 %33  
%35 = bitcast float\* %34 to <8 x float>\*  
%wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x ... float>\* %35, i32 4, <8 x i1> %30, <8 x float> undef), !tbaa !12, !alias.scope ... !16, !noalias !19  
%36 = add nsw i32 %mul5.i, %31  
%37 = sext i32 %36 to i64  
%38 = getelementptr inbounds float, float\* %2, i64 %37  
%39 = bitcast float\* %38 to <8 x float>\*  
%wide.masked.load28 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x ... float>\* %39, i32 4, <8 x i1> %30, <8 x float> undef), !tbaa !12, !alias.scope ... !23  
%40 = getelementptr inbounds float, float\* %0, i64 %37  
%41 = bitcast float\* %40 to <8 x float>\*  
%wide.masked.load29 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x ... float>\* %41, i32 4, <8 x i1> %30, <8 x float> undef), !tbaa !12, !alias.scope ... !24  
%42 = fneg <8 x float> %wide.masked.load28  
%43 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %42, <8 x float> ... %wide.masked.load29, <8 x float> %wide.masked.load)  
%44 = getelementptr inbounds float, float\* %1, i64 %33  
%45 = bitcast float\* %44 to <8 x float>\*  
%wide.masked.load30 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x ... float>\* %45, i32 4, <8 x i1> %30, <8 x float> undef), !tbaa !12, !alias.scope ... !25  
%46 = fdiv <8 x float> %43, %wide.masked.load30, !fpmath !26  
%47 = bitcast float\* %34 to <8 x float>\*  
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %46, <8 x float>\* ... %47, i32 4, <8 x i1> %30), !tbaa !12, !alias.scope !16, !noalias !19, ... !llvm.access.group !27  
%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>  
%48 = icmp eq i64 %index.next, 256  
br i1 %48, label %adi\_kernel6.exit.loopexit32, label %vector.body, ... !llvm.loop !29

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  
%49 = load float, float\* %arrayidx.i, align 4, !tbaa !12  
%add6.i = add nsw i32 %mul5.i, %conv.i  
%idxprom7.i = sext i32 %add6.i to i64  
%arrayidx8.i = getelementptr inbounds float, float\* %2, i64 %idxprom7.i  
%50 = load float, float\* %arrayidx8.i, align 4, !tbaa !12  
%arrayidx14.i = getelementptr inbounds float, float\* %0, i64 %idxprom7.i  
%51 = load float, float\* %arrayidx14.i, align 4, !tbaa !12  
%neg.i = fneg float %50  
%52 = tail call float @llvm.fmuladd.f32(float %neg.i, float %51, float %49)  
... #5  
%arrayidx21.i = getelementptr inbounds float, float\* %1, i64 %idxprom.i  
%53 = load float, float\* %arrayidx21.i, align 4, !tbaa !12  
%div.i = fdiv float %52, %53, !fpmath !26  
store float %div.i, float\* %arrayidx.i, align 4, !tbaa !12,  
... !llvm.access.group !27  
br label %if.end\_r\_exit.i

if.end\_r\_exit.i:  
%54 = or i64 % local\_id x.0, 1  
%add1.i.i.1 = add nuw nsw i64 %54, %mul.i.i  
%conv.i.1 = trunc i64 %add1.i.i.1 to i32  
%cmp.i.1 = icmp slt i32 %conv.i.1, %4  
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  
%55 = load float, float\* %arrayidx.i.1, align 4, !tbaa !12  
%add6.i.1 = add nsw i32 %mul5.i, %conv.i.1  
%idxprom7.i.1 = sext i32 %add6.i.1 to i64  
%arrayidx8.i.1 = getelementptr inbounds float, float\* %2, i64 %idxprom7.i.1  
%56 = load float, float\* %arrayidx8.i.1, align 4, !tbaa !12  
%arrayidx14.i.1 = getelementptr inbounds float, float\* %0, i64 %idxprom7.i.1  
%57 = load float, float\* %arrayidx14.i.1, align 4, !tbaa !12  
%neg.i.1 = fneg float %56  
%58 = tail call float @llvm.fmuladd.f32(float %neg.i.1, float %57, float ... %55) #5  
%arrayidx21.i.1 = getelementptr inbounds float, float\* %1, i64 %idxprom.i.1  
%59 = load float, float\* %arrayidx21.i.1, align 4, !tbaa !12  
%div.i.1 = fdiv float %58, %59, !fpmath !26  
store float %div.i.1, float\* %arrayidx.i.1, align 4, !tbaa !12,  
... !llvm.access.group !27  
br label %if.end\_r\_exit.i.1

if.end\_r\_exit.i.1:  
%60 = add nuw nsw i64 % local\_id x.0, 2  
%exitcond.not.1 = icmp eq i64 %60, 256  
br i1 %exitcond.not.1, label %adi\_kernel6.exit.loopexit, label ... %preregion\_for\_entry.entry.i, !llvm.loop !32

T	F
---	---

adi\_kernel6.exit.loopexit:  
br label %adi\_kernel6.exit

adi\_kernel6.exit:  
ret void

CFG for '\_pool\_kernel\_adi\_kernel6' function