```
entry:
                                                        tail call void asm sideeffect "NOP
                                                        %scevgep = getelementptr i32, i32* %a, i64 1000
                                                        %scevgep40 = getelementptr i32, i32* %b, i64 1000
                                                        %bound0 = icmp ugt i32* %scevgep40, %a
                                                        %bound1 = icmp ugt i32* %scevgep, %b
                                                        %found.conflict = and i1 %bound1, %bound0
                                                        \%0 = bitcast i32* \%a to <4 x i32>*
                                                        %1 = getelementptr inbounds i32, i32* %a, i64 4
                                                        \%2 = bitcast i32*\%1 to <4 x i32>*
                                                        %3 = bitcast i32* %b to <4 x i32>*
                                                        %4 = getelementptr inbounds i32, i32* %b, i64 4
                                                        \%5 = bitcast i32*\%4 to <4 x i32>*
                                                        br label %vector.memcheck
                                                   vector.memcheck:
                                                    %j.032 = phi i32 [ 0, %entry ], [ %inc15, %for.cond.cleanup3 ]
                                                    br i1 %found.conflict, label %for.body4.preheader, label
                                                   ... %vector.body.preheader
                                                                                                  F
                                                                                         vector.body.preheader:
                                                                                         store <4 x i32> <i32 0, i32 2, i32 4, i32 6>, <4 x i32>* %0, align 4, !tbaa
                                                                                         ... !4, !alias.scope !8, !noalias !11
                                                                                         store <4 x i32> <i32 8, i32 10, i32 12, i32 14>, <4 x i32>* %2, align 4,
                                         for.body4.preheader:
                                                                                         ...!tbaa!4,!alias.scope!8,!noalias!11
                                                                                         store <4 x i32> <i32 10, i32 11, i32 12, i32 13>, <4 x i32>* %3, align 4,
                                          br label %for.body4
                                                                                         ...!tbaa!4,!alias.scope!11
                                                                                         store <4 x i32> <i32 14, i32 15, i32 16, i32 17>, <4 x i32>* %5, align 4,
                                                                                         ...!tbaa!4,!alias.scope!11
                                                                                         br label %vector.body.1
                                                                                        vector.body.1:
                                                                                        %index.next5 = phi i64 [ 8, %vector.body.preheader ], [ %index.next,
                                                                                        ... %vector.body.1 ]
                                                                                        \%6 = phi < 4 \times i32 > [ < i32 0, i32 2, i32 4, i32 6 >, %vector.body.preheader ],
                                                                                        ... [ %20, %vector.body.1 ]
                                                                                        %vec.ind474 = phi <4 \times i32> [ <i32.0, i32.1, i32.2, i32.3>,
                                                                                        ... %vector.body.preheader ], [ %vec.ind.next50.1, %vector.body.1 ]
                                                                                        %vec.ind433 = phi <4 \times i32> [<i32.0, i32.1, i32.2, i32.3>,
                                                                                        ... %vector.body.preheader ], [ %vec.ind.next46.1, %vector.body.1 ]
                                                                                        %index2 = phi i64 [ 0, %vector.body.preheader ], [ %index.next.1,
                                                                                        ... %vector.body.1 ]
                                                                                        %7 = getelementptr inbounds i32, i32* %a, i64 %index.next5
                                                                                        %8 = add <4 x i32> %6, <i32 16, i32 16, i32 16, i32 16>
                                                                                        \%9 = \text{add} < 4 \times i32 > \%6, < i32\ 24, i32\ 24, i32\ 24, i32\ 24 > \%6
                                                                                        %10 = bitcast i32* \%7 to <4 x i32>*
                                                                                        store <4 x i32> %8, <4 x i32>* %10, align 4, !tbaa !4, !alias.scope !8,
                                                                                        ... !noalias !11
                                                                                        %11 = \text{getelementptr inbounds } i32, i32*\%7, i64 4
for.body4:
%indvars.iv = phi i64 [ %indvars.iv.next.1, %for.body4 ], [ 0,
                                                                                        %12 = bitcast i32*\%11 to <4 x i32>*
.. %for.body4.preheader ]
                                                                                        store <4 x i32> %9, <4 x i32>* %12, align 4, !tbaa !4, !alias.scope !8,
%arrayidx = getelementptr inbounds i32, i32* %a, i64 %indvars.iv
                                                                                        ... !noalias !11
%indvars.iv.tr = trunc i64 %indvars.iv to i32
                                                                                        %13 = getelementptr inbounds i32, i32* %b, i64 %index.next5
                                                                                        %14 = add <4 x i32> %vec.ind474, <i32 18, i32 18, i32 18, i32 18>
%32 = shl i32 %indvars.iv.tr, 1
store i32 %32, i32* %arrayidx, align 4, !tbaa !4
                                                                                         %15 = add <4 x i32> %vec.ind474, <i32 22, i32 22, i32 22, i32 22>
%arrayidx6 = getelementptr inbounds i32, i32* %b, i64 %indvars.iv
                                                                                        %16 = bitcast i32* %13 to <4 x i32>*
%33 = add nuw nsw i32 %indvars.iv.tr, 10
                                                                                        store <4 x i32> %14, <4 x i32>* %16, align 4, !tbaa !4, !alias.scope !11
store i32 %33, i32* %arrayidx6, align 4, !tbaa !4
                                                                                         \%17 = \text{getelementptr inbounds } i32, i32*\%13, i64 4
%indvars.iv.next = or i64 %indvars.iv, 1
                                                                                         %18 = bitcast i32* \%17 to < 4 x i32>*
%arrayidx.1 = getelementptr inbounds i32, i32* %a, i64 %indvars.iv.next
                                                                                        store <4 x i32> %15, <4 x i32>* %18, align 4, !tbaa !4, !alias.scope !11
%indvars.iv.tr.1 = trunc i64 %indvars.iv.next to i32
                                                                                         %index.next.1 = add nuw nsw i64 %index\overline{2}, 16
%34 = shl i32 %indvars.iv.tr.1, 1
                                                                                        %vec.ind.next46.1 = add <4 x i32> %vec.ind433, <i32 16, i32 16, i32 16, i32
store i32 %34, i32* %arrayidx.1, align 4, !tbaa !4
                                                                                        ... 16>
%arrayidx6.1 = getelementptr inbounds i32, i32* %b, i64 %indvars.iv.next
                                                                                        %vec.ind.next50.1 = add <4 x i32> %vec.ind474, <i32 16, i32 16, i32 16, i32
                                                                                        ... 16>
%35 = add nuw nsw i32 %indvars.iv.tr.1, 10
store i32 %35, i32* %arrayidx6.1, align 4, !tbaa !4
                                                                                        %19 = getelementptr inbounds i32, i32* %a, i64 %index.next.1
%indvars.iv.next.1 = add nuw nsw i64 %indvars.iv, 2
                                                                                        %20 = shl <4 x i32> %vec.ind.next46.1, <i32 1, i32 1, i32 1>
%exitcond.not.1 = icmp eq i64 %indvars.iv.next.1, 1000
                                                                                        %21 = add <4 x i32> %20, <i32 8, i32 8, i32 8, i32 8>
                                                                                        %22 = bitcast i32* %19 to <4 x i32>*
br i1 %exitcond.not.1, label %for.cond.cleanup3.loopexit, label %for.body4,
...!llvm.loop!17
                                                                                        store <4 x i32> %20, <4 x i32>* %22, align 4, !tbaa !4, !alias.scope !8,
                                                                                        ... !noalias !11
                                                        F
                                                                                        %23 = getelementptr inbounds i32, i32* %19, i64 4
                                                                                        %24 = bitcast i32* \%23 to < 4 x i32>*
                                                                                        store <4 x i32> %21, <4 x i32>* %24, align 4, !tbaa !4, !alias.scope !8,
                                                                                        %25 = getelementptr inbounds i32, i32* %b, i64 %index.next.1
                                                                                        %26 = add <4 x i32> %vec.ind474, <i32 26, i32 26, i32 26, i32 26>
                                                                                        %27 = add <4 x i32> %vec.ind474, <i32 30, i32 30, i32 30, i32 30>
                                                                                        %28 = bitcast i32* %25 to <4 x i32>*
                                                                                        store <4 x i32> %26, <4 x i32>* %28, align 4, !tbaa !4, !alias.scope !11
                                                                                        %29 = \text{getelementptr inbounds } i32, i32* \%25, i64 4
                                                                                        %30 = bitcast i32*\%29 to <4 x i32>*
                                                                                        store <4 x i32> %27, <4 x i32>* %30, align 4, !tbaa !4, !alias.scope !11
                                                                                         %index.next = or i64 %index.next.1, 8
                                                                                        %31 = icmp eq i64 %index.next, 1000
                                                                                        br i1 %31, label %for.cond.cleanup3.loopexit10, label %vector.body.1,
                                                                                        ... !llvm.loop !13
                                                                                                                                                  F
                                                                                          for.cond.cleanup3.loopexit10:
                          for.cond.cleanup3.loopexit:
                                                                                          br label %for.cond.cleanup3
                           br label %for.cond.cleanup3
                           for.cond.cleanup3:
                           %inc15 = add nuw nsw i32 %j.032, 1
                           %exitcond36.not = icmp eq i32 %inc15, 1000
                           br i1 %exitcond36.not, label %for.cond.cleanup, label %vector.memcheck,
                           ...!llvm.loop!16
                                           Τ
                                                                                  F
                                   for.cond.cleanup:
                                    ret void
```