

```
%8 = bitcast float* %0 to i8*
%mul.i.i = shl i64 %4, 5
%add.i = add nsw i32 %1, 1
%mul3.i.i = shl i64 %5, 3
%mul16.i = mul nsw i32 %2, %1
%9 = trunc i64 %4 to i32
%10 = shl i32 %9, 5
%11 = trunc i64 %5 to i32
%12 = shl i32 %11, 3
%13 = add i32 %10, %12
%14 = shl i32 %1, 1
%15 = add i32 %13, %14
%16 = add i32 %15, 2
%17 = add i32 %10, %14
%18 = or i32 %17, 1
%19 = trunc i64 %4 to i32
%20 = shl i32 %19, 5
%21 = trunc i64 %5 to i32
%22 = shl i32 %21, 3
%23 = add i32 %20, %22
%24 = shl i32 %1, 1
%25 = add i32 %23, %24
%26 = add i32 %25, 2
%scevgep7 = getelementptr float, float* %0, i64 32
%uglygep = getelementptr i8, i8* %8, i64 1
%27 = add i32 %22, %1
%28 = add i32 %27, 1
%29 = mul i32 %28, %2
%30 = add i32 %29, %1
%31 = add i32 %20, %24
%32 = sext i32 %31 to i64
%33 = or i64 %32, 1
%scevgep11 = getelementptr float, float* %0, i64 %33
%34 = add nsw i64 %32, 33
%scevgep13 = getelementptr float, float* %0, i64 %34
br label %region for_entry.region for_init.i
```

```

    pre local_for_entry.region for init.i:
    % local_id y_0 = phi i64 [0, %7 ], [ %73, %region_for_end.i ]
    %35 = trunc i64 % local_id y_0 to i32
    %36 = add i32 %26, %35
    %37 = sext i32 %36 to i64
    %scevgep = getelementptr float, float* %0, i64 %37
    %scevgep6 = bitcast float* %scevgep to i8*
    %scevgep8 = getelementptr float, float* %scevgep7, i64 %37
    %38 = trunc i64 % local_id y_0 to i32
    %39 = mul i32 %38, %2
    %40 = add i32 %39, %30
    %41 = sext i32 %40 to i64
    %42 = shl nsw i64 %41, 2
    %scevgep10 = getelementptr i8, i8* %uglygep, i64 %42
    %add6.i.i = add nuw nsw i64 % local_id y_0, %mul3.i.i
    %43 = trunc i64 %add6.i.i to i32
    %conv7.i = add i32 %add.i, %43
    %cmp.i = icmp slt i32 %conv7.i, %2
    %mul.i = mul nsw i32 %conv7.i, %2
    %add13.i = add nsw i32 %mul.i, %1
    %idxprom14.i = sext i32 %add13.i to i64
    %arrayidx15.i = getelementptr @inbounds float, float* %0, i64 %idxprom14.i
    br 1 %cmp.i, label %vector.scevcheck, label %region_for_end.i

```

```
vector.scevcheck:
%44 = trunc i64 % local_id_y, to i32
%45 = add i32 %16, %44
%ident.check = icmp ne i32 %2, 1
%46 = icmp sgt i32 %45, 2147483616
%47 = or i1 %ident.check, %46
%48 = icmp sgt i32 %18, 2147483616
%49 = or i1 %47, %48
br i1 %49, label %region_for_entry.entry.i.us.preheader, label
... %vector.memcheck
```

```
vector.memcheck
%bound0 = icmp ugt i8* %scevgep10, %scevgep6
%bound1 = icmp ult float* %arrayidx15.1, %scevgep13
%found.conflict = and i1 %bound0, %bound1
%bound015 = icmp ult float* %scevgep, %scevgep13
%bound116 = icmp ult float* %scevgep11, %scevgep8
%found.conflict17 = and i1 %bound015, %bound116
%conflict.rdx = or i1 %found.conflict, %found.conflict17
br i1 %conflict.rdx, label %region_for_entry.entry.i.us.preheader, label
... %vector.ph
```

```
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.splatinsert18 = insertelement <8 x i32> undef, i32 %add.i, i32 0
%broadcast.splat19 = shufflevector <8 x i32> %broadcast.splatinsert18, <8 x
... i32> undef, <8 x i32> zeroinitializer
%broadcast.splatinsert20 = insertelement <8 x float*> undef, float*
..., arrayidx15.i, i32 0
%broadcast.splat21 = shufflevector <8 x float*> %broadcast.splatinsert20, <8
... x float*> undef, <8 x i32> zeroinitializer
br label %vector.body
```

```
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 ]
%50 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
%51 = trunc <8 x i64> %50 to <8 x i32>
%52 = add <8 x i32> %broadcast.splat19, %51
%53 = icmp slt <8 x i32> %52, <i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1>
... 1, i32 1>
%54 = extractelement <8 x i32> %52, i32 0
%55 = add nsw i32 %54, %mul.i
%56 = sext i32 %55 to i64
%57 = getelementptr inbounds float, float* %0, i64 %56
%58 = bitcast float* %57 to <8 x float>*
%wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %58, i32 4, <8 x i1> %53, <8 x float> undef), !tbaa !12, !alias.scope
... !16, !noalias !19
%wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8 x
... x float*> %broadcast.splat21, i32 4, <8 x i1> %53, <8 x float> undef), !tbaa
... !12, !alias.scope !22
%59 = add nsw i32 %54, %mul16.i
%60 = sext i32 %59 to i64
%61 = getelementptr inbounds float, float* %0, i64 %60
%62 = bitcast float* %61 to <8 x float>*
%wide.masked.load22 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %62, i32 4, <8 x i1> %53, <8 x float> undef), !tbaa !12, !alias.scope
... !23
%63 = fneg <8 x float> %wide.masked.gather
%64 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %63, <8 x float>
... %wide.masked.load22, <8 x float> %wide.masked.load)
%65 = bitcast float* %57 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %64, <8 x float*>
... %65, i32 4, <8 x i1> %53), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !24
%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>
%66 = icmp eq i64 %index.next, 32
br i1 %66, label %region_for_end.i.loopexit24, label %vector.body,
... !llvm.loop !27
```

```

pre: region for entry.entri.i.us:
% local_id x.0.us = phi i64 [%72, %if.end.r_exit.i.us], [0,
... %region for entry.entri.i.us.preheader ]
%add1.i.us = add nuw nsw i64 %local_id_x.0.us, %mul.i.i
%67 = trunc i64 %add1.i.i.us to i32
%conv2.i.us = add i32 %add.i.i, %67
%cmp9.i.us = icmp slt i32 %conv2.i.us, %2
br i1 %cmp9.i.us, label %if.then.i.us, label %if.end.r_exit.i.us

```

```

if.then.i.us:
%add11.i.us = add nsw i32 %conv2.i.us, %mul.i
%idxprom.i.us = sext i32 %add11.i.us to i64
%arrayidx.i.us = getelementptr inbounds float, float* %0, i64 %idxprom.i.us
%68 = load float, float* %arrayidx.i.us, align 4, !tbaa !12
%69 = load float, float* %arrayidx15.i, align 4, !tbaa !12
%add17.i.us = add nsw i32 %conv2.i.us, %mul16.i
%idxprom18.i.us = sext i32 %add17.i.us to i64
%arrayidx19.i.us = getelementptr inbounds float, float* %0, i64
... %idxprom18.i.us
%70 = load float, float* %arrayidx19.i.us, align 4, !tbaa !12
%neg.i.us = fneg float %69
%71 = tail call float @llvm.fmuladd.f32(float %neg.i.us, float %70, float
... %68) #6
store float %71, float* %arrayidx.i.us, align 4, !tbaa !12,
... !llvm.access.group !24
br label %if.end.r_exit.i.us

```

```
if.end.r_exit.i.us:  
%72 = add nuw nsw i64 % local_id_x0.us, 1  
%exitcond.not = icmp eq i64 %72, 32  
br i1 %exitcond.not, label %prejoin_for_end.i.loopexit, label  
... %prejoin_for_entry.entry.i.us, !llvm.loop !30
```

```

region_for_end.i.loopexit:
    label_%region_for_end.i:

```

```

pregon for end.i:
%73 = add nuw nsw i64 %local_id.y0, 1
%exitcond2.not = icmp eq i64 %73, 8
br i1 %exitcond2.not, label %lu_kernel2.exit, label
... %pregon_for_entry.pregon_for_init.i,!llvm.loop !31

```

```
lu_kernel2.exit:
    ret void
```