

```
%7:
%8 = bitcast float* %0 to i8*
%9 = shl i64 %4, 5
%add.i = add nsw i32 %1, 1
%10 = shl i64 %5, 3
%mul16.i = mul nsw i32 %2, %1
%11 = trunc i64 %4 to i32
%12 = shl i32 %11, 5
%13 = trunc i64 %5 to i32
%14 = shl i32 %13, 3
%15 = add i32 %12, %14
%16 = shl i32 %1, 1
%17 = add i32 %15, %16
%18 = add i32 %17, 2
%19 = add i32 %12, %16
%20 = or i32 %19, 1
%21 = trunc i64 %4 to i32
%22 = shl i32 %21, 5
%23 = trunc i64 %5 to i32
%24 = shl i32 %23, 3
%25 = add i32 %22, %24
%26 = shl i32 %1, 1
%27 = add i32 %25, %26
%28 = add i32 %27, 2
%sevegep7 = getelementptr float, float* %0, i64 32
%uglygep = getelementptr i8, i8* %8, i64 1
%29 = add i32 %24, %1
%30 = add i32 %29, 1
%31 = mul i32 %30, %2
%32 = add i32 %31, %1
%33 = add i32 %22, %26
%34 = sext i32 %33 to i64
%35 = or i64 %34, 1
%sevegep11 = getelementptr float, float* %0, i64 %
%36 = add nsw i64 %34, 33
%sevegep13 = getelementptr float, float* %0, i64 %
br label %region for_entry.region for_init.i
```

```

    pregon for entry.pregon for init.i
% local_id y_0 = phi i64 [0, %7 ], [ %77,%region_for_end.i ]
%37 = trunc i64 % local_id_y_0 to i32
%38 = add i32 %28, %37
%39 = sext i32 %38 to i64
%scevgep = getelementptr float, float* %, i64 %39
%scevgep6 = bitcast float* %scevgep to i8*
%scevgep8 = getelementptr float, float* %scevgep7, i64 %39
%40 = trunc i64 % local_id_y_0 to i32
%41 = mul i32 %40, %2
%42 = add i32 %32, %41
%43 = sext i32 %42 to i64
%44 = shl nsw i64 %43, 2
%scevgep10 = getelementptr i8, i8* %uglygep, i64 %44
%45 = add nuw nsw i64 %local_id_y_0, %10
%46 = trunc i64 %45 to i32
%conv7.i = add i32 %add.i, %46
%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
%arrayid15.i = getelementptr @inbounds float, float* %, i64 %idxprom14.i
br i1 %cmp.i, label %vector.scevcheck, label %region_for_end.i

```

```
vector.scevcheck:
%47 = trunc i64 % local_id_y0 to i32
%48 = add i32 %18, %47
%ident.check = icmp ne i32 %2, 1
%49 = icmp sgt i32 %48, 2147483616
%50 = or i1 %ident.check, %49
%51 = icmp sgt i32 %20, 2147483616
%52 = or i1 %50, %51
br i1 %52, label %region_for_entry.entry.us.preheader, label
... %vector.memcheck
```

```
vector.memcheck:
%bound0 = icmp ugt i8* %scevgep10, %scevgep6
%bound1 = icmp ult float* %arrayidx15.i, %scevgep8
%found.conflict = and i1 %bound0, %bound1
%bound015 = icmp ult float* %scevgep, %scevgep3
%bound116 = icmp ult float* %scevgep11, %scevgep18
%found.conflict17 = and i1 %bound015, %bound116
%conflict.rdx = or i1 %found.conflict, %found.conflict17
br i1 %conflict.rdx, label %pregion_for_entry.entry.i.us.preheader, label
%vector.ph
```

```
vector:ph:
%broadcast.splatinsert = insertelement <8 x i64> undef, i64 %9, 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 ]
%53 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
%54 = trunc <8 x i64> %53 to <8 x i32>
%55 = add <8 x i32> %broadcast.splat19, %54
%56 = icmp slt <8 x i32> %55, <i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1>
... 1, i32 1>
%57 = extractelement <8 x i32> %55, i32 0
%58 = add nsw i32 %57, %mul.i
%59 = sext i32 %58 to i64
%60 = getelementptr inbounds float, float* %, i64 %59
%61 = bitcast float* %60 to <8 x float>*
%wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %61, i32 4, <8 x i1> %56, <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> %56, <8 x float> undef), !tbaa
... !12, !alias.scope !22
%62 = add nsw i32 %57, %mul16.i
%63 = sext i32 %62 to i64
%64 = getelementptr inbounds float, float* %, i64 %63
%65 = bitcast float* %64 to <8 x float>*
%wide.masked.load22 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %65, i32 4, <8 x i1> %56, <8 x float> undef), !tbaa !12, !alias.scope
... !23
%66 = fneg <8 x float> %wide.masked.gather
%67 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %66, <8 x float>
... %wide.masked.load22, <8 x float> %wide.masked.load)
%68 = bitcast float* %60 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %67, <8 x float>*
... %68, i32 4, <8 x i1> %56), !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>
%69 = icmp eq i64 %index.next, 32
br i1 %69, label %region_for_end.i.loopexit24, label %vector.body,
... !llvm.loop !27
```

```

preparation for entry.entry.i.us:
% local_id x.0.us = phi i64 [ %76, %if.end.r.exit.i.us ], [ 0,
... %preparation for entry.entry.i.us.preheader ]
%70 = add nuw nsw i64 % local_id x.0.us, %9
%71 = trunc i64 %70 to i32
%conv2.i.us = add i32 %add.i. %71
%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* %, i64 %idxprom.i.us
%72 = load float, float* %arrayidx.i.us, align 4, !tbaa !12
%73 = load float, float* %arrayidx.i.us, 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* %, i64
... %idxprom18.i.us
%74 = load float, float* %arrayidx19.i.us, align 4, !tbaa !12
%neg.i.us = fneg float %73
%75 = tail call @llvm.fmuladd.f32(float %neg.i.us, float %74, float
... %72) #6
store float %75, 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:
%76 = add nuw nsw i64 % local_id_x.0.us, 1
%exitcond = icmp eq i64 %76, 32
br i1 %exitcond, label %pregion_for_end.i.loopexit, label
... %pregion_for_entry.entry.i.us, !llvm.loop !30
```

```
preregion for_end.i.loopexit:
br label %preregion for_end.i
```

```
preregion for_end.i.loopexit24:  
br label %preregion for_end.i
```

```
pregion_for_end.i:
%77 = add nuw nsw i64 % local_id_y0, 1
%exitcond2 = icmp eq i64 %77, 8
br i1 %exitcond2, label %lu_kernel2.exit, label
... %pregion_for_entry.pregion_for_init.i, !llvm.loop !31
```

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