

```
%:7:
%mul3.1.i = shl i64 %5, 3
%mul.i.i = shl i64 %4, 5
%sub.i.i = add nsw i32 %2, -1
%8 = trunc i64 %5 to i32
%9 = mul i32 %8, %2
%10 = shl i32 %9, 3
%11 = trunc i64 %4 to i32
%12 = shl i32 %11, 5
%13 = add i32 %10, %12
%14 = zext i32 %2 to i64
%15 = add i32 %13, -8
%16 = or i32 %15, 7
%17 = or i32 %13, 1
%18 = shl i32 %8, 3
%19 = or i32 %18, 1
%20 = mul i32 %19, %2
%21 = add i32 %20, %12
%22 = add i32 %18, -1
%23 = mul i32 %22, %2
%24 = add i32 %23, %12
%25 = trunc i64 %5 to i32
%26 = mul i32 %25, %2
%27 = shl i32 %26, 3
%28 = trunc i64 %4 to i32
%29 = shl i32 %28, 5
%30 = add i32 %27, %29
%31 = zext i32 %2 to i64
%ceveqep17 = getelementptr float, float* %1, i64 32
%32 = shl i32 %25, 3
%33 = add i32 %32, -1
%34 = mul i32 %33, %2
%35 = add i32 %35, %29
%ceveqep22 = getelementptr float, float* %0, i64 32
%36 = or i32 %32, 1
%37 = mul i32 %36, %2
%38 = add i32 %37, %29
%ceveqep32 = getelementptr float, float* %0, i64 32
%39 = or i32 %31, 1
%40 = zext i32 %39 to i64
%ceveqep32 = getelementptr float, float* %0, i64 32
%41 = add i32 %27, %29
%42 = add i32 %41, -8
%43 = or i32 %42, 7
%ceveqep37 = getelementptr float, float* %0, i64 32
%44 = add i32 %43, %29
%ceveqep37 = getelementptr float, float* %0, i64 32
br label %region for entry region for init.i
```

```

program for-entropy-pregion for-init.i
% local id_y,0 = phi i64 [ 0, %7, [%119,%pregion_for_end.i]
%44 = mul i64 % local id_y,0,%31
%45 = trunc i64 %44 to i32
%46 = add i32 %30,%45
%47 = sext i32 %46 to i64
%$cveqep = getelementptr float,float* %i,i64 %47
%$cveqep18 = getelementptr float,float* %$cveqep17,i64 %47
%48 = trunc i64 %44 to i32
%49 = add i32 %35,%48
%50 = sext i32 %49 to i64
%$cveqep20 = getelementptr float,float* %i,i64 %50
%$cveqep23 = getelementptr float,float* %$cveqep22,i64 %50
%51 = trunc i64 %44 to i32
%52 = add i32 %38,%51
%53 = sext i32 %52 to i64
%$cveqep25 = getelementptr float,float* %i,i64 %53
%$cveqep28 = getelementptr float,float* %$cveqep27,i64 %53
%54 = add i64 %44,%40
%$ext = shl i64 %54,%32
%55 = ashr exact i64 %$ext,%32
%$cveqep30 = getelementptr float,float* %i,i64 %55
%$cveqep33 = getelementptr float,float* %$cveqep32,i64 %55
%56 = trunc i64 %44 to i32
%57 = add i32 %43,%56
%58 = sext i32 %57 to i64
%$cveqep35 = getelementptr float,float* %i,i64 %58
%$cveqep38 = getelementptr float,float* %$cveqep37,i64 %58
%59 = mul i64 % local id_y,%14
%$add6.i.i = add nsw nsw i64 % local id_y,%mul3.i.i
%$conv1 = trunc i64 %$add6.i.i to i32
%$cmp.i.i = icmp sgt i32 %$conv1,%i
%$mul1.i = mul nsw i32 %$conv1,%i
%$add25.i.i = add nsw nsw i32 %$conv1,%mul25.i.i
%$mul26.i.i = mul nsw i32 %$add25.i.i,%2
%$sub31.i.i = add nsw i32 %$conv1,-i
%$mul32.i.i = mul nsw i32 %$sub31.i.i,%2
%$cmp4.i.i = icmp sgt i32 %$sub1.i,%$conv1
%$or.cond = and i1 %$cmp1.i,%$cmp4.i.i
br i1 %or.cond,cond,label %vector.$cveqcheck,label %pregion_for_end.i

```

```
vector<vecvec>
%60 = trunc i64 %59 to i32
%61 = add i32 %24, %60
%62 = trunc i64 %59 to i32
%63 = add i32 %21, %62
%64 = trunc i64 %59 to i32
%65 = add i32 %17, %64
%66 = trunc i64 %59 to i32
%67 = add i32 %16, %66
%68 = trunc i64 %59 to i32
%69 = add i32 %13, %68
%70 = icmp sgt i32 %69, 2147483616
%71 = icmp sgt i32 %67, 2147483616
%72 = or i1 %70, %71
%73 = icmp sgt i32 %65, 2147483616
%74 = or i1 %72, %73
%75 = icmp sgt i32 %63, 2147483616
%76 = or i1 %74, %75
%77 = icmp sgt i32 %61, 2147483616
%78 = or i1 %76, %77
br i1 %78, label %region_for_entry.entry.i.us.us.preheader, label
```

```
vector memcheck:
%b0und0 = icmp ult float* %scevgevp, %scevgevp23
%b0und1 = icmp ult float* %scevgevp20, %scevgevp18
%found1.conflict = and i1 %b0und0, %b0und1
%b0und405 = icmp ult float* %scevgevp, %scevgevp28
%b0und146 = icmp ult float* %scevgevp25, %scevgevp18
%found.conflict147 = and i1 %b0und405, %b0und146
%conflict.rdx1 = or i1 %found.conflict, %found.conflict47
%b0und408 = icmp ult float* %scevgevp, %scevgevp33
%b0und148 = icmp ult float* %scevgevp30, %scevgevp18
%found1.conflict50 = and i1 %b0und408, %b0und148
%conflict.rdx51 = or i1 %conflict.rdx1, %found.conflict50
%b0und052 = icmp ult float* %scevgevp, %scevgevp38
%b0und153 = icmp ult float* %scevgevp35, %scevgevp18
%found.conflict54 = and i1 %b0und052, %b0und153
%conflict.rdx55 = or i1 %conflict.rdx51, %found.conflict54
%b0und056 = icmp ult float* %scevgevp42, %b0und157
%b0und157 = icmp ult float* %scevgevp41, %b0und157
%conflict.rdx59 = and i1 %found.conflict55, %b0und157
%found.conflict58 = or i1 %conflict.rdx59, %found.conflict58
br i1 %conflict.rdx59, label %regionfor_entryentry.us.us
... label %vector.ph
```

```
preregion_for_entry.entry.i.us.us.preheader:
br label %preregion_for_entry.entry.i.us.us
```

```
vector.ph:
%broadcast.splatinser = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
%broadcast.splat = shufflevector <8 x i64> %broadcast.splatinser, <8 x i64>
... undef, <8 x i32> zeroinitializer
%broadcast.splatinser60 = insertelement <8 x i32> undef, i32 %sub.i.i, i32 0
%broadcast.splat61 = shufflevector <8 x i32> %broadcast.splatinser60, <8 x
... i32> 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 ]
%79 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
%80 = trunc <8 x i64> %79 to <8 x i32>
%81 = icmp sgt <8 x i32> %80, zeroinitializer
%82 = icmp sgt <8 x i32> %80, broadcast.splat61, %80
%83 = add <8 x i1> %82, %81
%84 = extractelement <8 x i32> %80, i32 0
%85 = add i32 %mul1, %84
%86 = sext i32 %85 to i64
%87 = getelementptr inbounds float, float* %0, i64 %86
%88 = bitcast float* %87 to <8 x float>*
%wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %88, i32 4, <8 x i1> %83, <8 x float> undef), !tbaa !12, !alias.scope
... !16
%89 = add i32 %85, -1
%90 = sext i32 %89 to i64
%91 = getelementptr inbounds float, float* %0, i64 %90
%92 = bitcast float* %91 to <8 x float>*
%wide.masked.load62 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %92, i32 4, <8 x i1> %83, <8 x float> undef), !tbaa !12, !alias.scope
... !19
%93 = fadd <8 x float> %wide.masked.load, %wide.masked.load62
%94 = add i32 %85, -1
%95 = sext i32 %94 to i64
%96 = getelementptr inbounds float, float* %0, i64 %95
%97 = bitcast float* %96 to <8 x float>*
%wide.masked.load63 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %97, i32 4, <8 x i1> %83, <8 x float> undef), !tbaa !12, !alias.scope
... !22
%98 = fadd <8 x float> %93, %wide.masked.load63
%99 = add nsw i32 %mul26.i, %84
%100 = sext i32 %99 to i64
%101 = getelementptr inbounds float, float* %0, i64 %100
%102 = bitcast float* %101 to <8 x float>*
%wide.masked.load64 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %102, i32 4, <8 x i1> %83, <8 x float> undef), !tbaa !12,
... !alias.scope !23
%103 = fadd <8 x float> %98, %wide.masked.load64
%104 = add nsw i32 %mul32.i, %84
%105 = sext i32 %104 to i64
%106 = getelementptr inbounds float, float* %0, i64 %105
%107 = bitcast float* %106 to <8 x float>*
%wide.masked.load65 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %107, i32 4, <8 x i1> %83, <8 x float> undef), !tbaa !12,
... !alias.scope !25
%108 = fadd <8 x float> %103, %wide.masked.load65
%109 = fmul <8 x float> %108, <float 0x3FC99999A0000000, float
... 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
... 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
... 0x3FC99999A0000000>
%110 = getelementptr inbounds float, float* %1, i64 %86
%111 = bitcast float* %110 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %109, <8 x float>*
... %111, i32 4, <8 x i1> %83), !tbaa !12, !alias.scope !27, !noalias !29,
... !llvm.access.group !30
%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>
%112 = icmp eq i64 %index.next, 32
br i1 %112, label %region_for_end.i.loopexit67, label %vector.body,
... !llvm.loop !33

```

```

pregon for entry.entry.i.us.us:
  % local id x.0.us.us = phi i64 [%i18,%if.end.i.us.us ], [ 0,
... %region for entry.entry.i.us.us.preheader
  %add1.i.us.us = add nuw nsw i64 % local id x.0.us.us, %mul.i.i
  %conv2.i.us.us = trunc i64 %add1.i.us.us to i32
  %cmp7.i.us.us = icmp sgt i32 %conv2.i.us.us, 0
  %cmp11.i.us.us = icmp sgt i32 %sub.i.i, %conv2.i.us.us
  %or.cond70.i.us.us = and i1 %cmp11.i.us.us, %cmp7.i.us.us
br i1 %or.cond70.i.us.us, label %if.then.i.us.us, label %if.end.i.us.us

```

```

if then i.us.us:
  %add1.i.us.us = add i32 %mul.i, %conv2.i.us.us
  %idxprom1.i.us.us = sext i32 %add1.i.us.us to i64
  %arrayidx1.i.us.us = getelementptr inbounds float, float* %, i64
  ... %idxprom1.i.us.us
  %i113 = load float, float* %arrayidx1.i.us.us, align 4, !tbaa !i2
  %add15.i.us.us = add i32 %add1.i.us.us, -1
  %idxprom16.i.us.us = sext i32 %add15.i.us.us to i64
  %arrayidx17.i.us.us = getelementptr inbounds float, float* %, i64
  ... %idxprom16.i.us.us
  %i114 = load float, float* %arrayidx17.i.us.us, align 4, !tbaa !i2
  %add18.i.us.us = fadd float %i113, %i114
  %add21.i.us.us = add i32 %add.i.us.us, 1
  %idxprom22.i.us.us = sext i32 %add21.i.us.us to i64
  %arrayidx23.i.us.us = getelementptr inbounds float, float* %, i64
  ... %idxprom22.i.us.us
  %add24.i.us.us = fadd float %arrayidx23.i.us.us, align 4, !tbaa !i2
  %add27.i.us.us = add nsw i32 %mul26.i, %conv2.i.us.us
  %idxprom28.i.us.us = sext i32 %add27.i.us.us to i64
  %arrayidx29.i.us.us = getelementptr inbounds float, float* %, i64
  ... %idxprom28.i.us.us
  %i116 = load float, float* %arrayidx29.i.us.us, align 4, !tbaa !i2
  %add30.i.us.us = fadd float %add24.i.us.us, %i116
  %add33.i.us.us = add nsw i32 %mul32.i, %conv2.i.us.us
  %idxprom34.i.us.us = sext i32 %add33.i.us.us to i64
  %arrayidx35.i.us.us = getelementptr inbounds float, float* %, i64
  ... %idxprom34.i.us.us
  %i117 = load float, float* %arrayidx35.i.us.us, align 4, !tbaa !i2
  %add36.i.us.us = fadd float %add30.i.us.us, %i117
  %mul37.i.us.us = fmul float %add36.i.us.us, 0x3F9C9999A0000000
  %arrayidx41.i.us.us = getelementptr inbounds float, float* %, i64
  ... %idxprom.i.us.us
  store float %mul.i.us.us, float* %arrayidx41.i.us.us, align 4, !tbaa !i2,
    !llvm.access_group 130
  br label %if_end.i.us.us

```

```
if.end.i.us.us:
    %118 = add nuw nsw i64 % local_id.x0.us.us, 1
    %exitcond.not = icmp eq i64 %118, 32
    br i1 %exitcond.not, label %pregion_for_end.i.loopexit, label
    ... %pregion_for_end.i.loopexit.us.us, !llvm.loop !36
```

```
pregion_for_end.i.loopexit67:  
br label %pregion_for_end.i
```

```
region for_end.i.loopexit:  
    br label %region_for_end.i
```

```

    pregon for end.i:
      %t19 = add nuw nsw i64 %local_id.y0, 1
      %exitcond3.not = icmp eq i64 %t19, 8
      br i1 %exitcond3.not, label %runJacobi2D_kernel1_exit, label
      ... %pregon_for_entry.pregon_for_init.i, !llvm.loop!l37

```

```
runJacobi2D_kernel1.exit:  
ret void
```