

```

%mul1.i = shl i64 %6, 5
%mul31.i = shl i64 %67, 3
%sub1 = add nsw i32 %3, -1
%sub4.i = add nsw i32 %4, -1
%10 = trunc i64 %7 to i32
%11 = mul i32 %10, %4
%12 = shl i32 %11, 3
%13 = trunc i64 %6 to i32
%14 = shl i32 %13, 5
%15 = add i32 %12, %14
%16 = zext i32 %4 to i64
%17 = or i32 %15, 1
%18 = shl i32 %10, 3
%19 = or i32 %18, 1
%20 = mul i32 %19, %4
%21 = add i32 %20, %14
%22 = trunc i64 %7 to i32
%23 = mul i32 %22, %4
%24 = shl i32 %23, 3
%25 = trunc i64 %6 to i32
%26 = shl i32 %25, 5
%27 = add i32 %24, %26
%28 = zext i32 %4 to i64
%csvegv10 = @getelementptr float, float* %2, i64 32
%csvegv15 = @getelementptr float, float* %60, i64 32
%29 = or i32 %27, 1
%30 = zext i32 %29 to i64
%csvegv20 = @getelementptr float, float* %60, i64 32
%csvegv25 = @getelementptr float, float* %61, i64 32
%31 = shl i32 %21, 3
%32 = or i32 %31, 1
%33 = mul i32 %32, %4
%34 = add i32 %33, %26
%csvegv30 = @getelementptr float, float* %1, i64 32
%bound0 = icmp ugt float* %csvegv15, %2
%bound1 = icmp ugt float* %csvegv10, %0
%found.conflict = and i1 %bound0, %bound1
%bound036 = icmp ugt float* %csvegv25, %2
%bound137 = icmp ugt float* %csvegv10, %1
%found.conflict38 = and i1 %bound036, %bound137
%broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul1.i, i32 0
...
%broadcast.splatinsert44 = insertelement <8 x i32> undef, i32 %sub4.i, i32 0
%broadcast.splatinsert45 = shufflevector <8 x i32> %broadcast.splatinsert44, i32 0, i32 0
...
%broadcast.splatinsert45 = shufflevector <8 x i32> %broadcast.splatinsert44, i32 0, i32 0
%37 = or <8 x i32> %broadcast.splat, to <8 x i32>
%36 = or <8 x i32> %35, <i32 0, i32 1, i32 2, i32 3, i32 4, i32 5, i32 6, i32 7>
%37 = icmp slt <8 x i32> %broadcast.splat45, %36
%38 = extractelement <8 x i32> %36, i32 0
%39 = trunc <8 x i64> %broadcast.splat, to <8 x i32>
%40 = or <8 x i32> %39, <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32 14, i32 15>
%41 = icmp slt <8 x i32> %broadcast.splat45, %40
%42 = extractelement <8 x i32> %40, i32 0
%43 = trunc <8 x i64> %broadcast.splat, to <8 x i32>
%44 = or <8 x i32> %43, <i32 16, i32 17, i32 18, i32 19, i32 20, i32 21, i32 22, i32 23>
%45 = icmp slt <8 x i32> %broadcast.splat45, %44
%46 = extractelement <8 x i32> %44, i32 0
%47 = trunc <8 x i64> %broadcast.splat, to <8 x i32>
%48 = or <8 x i32> %47, <i32 24, i32 25, i32 26, i32 27, i32 28, i32 29, i32 30, i32 31>
%49 = icmp slt <8 x i32> %broadcast.splat45, %48
%50 = extractelement <8 x i32> %48, i32 0
br label %region_for_entry_pregion_for_init.i

```

```

program for entropy (for neg.1)
% local id_y0 = phi h4 10 9, i3, 1 %175, %region for_end.i 1
%51 = mul i64 %local id_y0, %z68
%52 = trunc i64 %51 to i32
%53 = add i32 %52 %27, %52
%54 = sext i32 %53 to i64
%55 = %sexcvtptr float, float* %2, i64 %54
%56 = %sexcvtptr float, float* %sexcvtptr, i64 %54
%55 = add i64 %51, %30
%56 = shl i64 %55, %32
%56 = ashr exact i64 %56, %32
%57 = %sexcvtptr float, float* %56, i64 %56
%58 = %sexcvtptr float, float* %sexcvtptr, i64 %56
%57 = trunc i64 %58 to i32
%58 = add i32 %54, %57
%59 = sext i32 %58 to i64
%sexcvtptr28 = %sexcvtptr float, float* %1, i64 %59
%sexcvtptr31 = %sexcvtptr float, float* %sexcvtptr30, i64 %59
%60 = mul i64 %6, local id_y0, %i6
%add64 = add i64 %60, %local id_y0, %mul3.i.i
%conv2.i = %conv i64 %add6.i.i to i32
%cmp.i = icmp sgt i32 %sub.i, %conv2.i
%mul.i = mul i64 %conv2.i, %mul2.i
%add.i64 = add i64 %mul.i, %mul2.i
%mul.i91 = mul i64 %conv2.i, %add.i64
br i1 %cmp, label %vectorcheck, label %region for_end.i

```

```
vector.sveccheck:
%61 = trunc i64 %60 to i32
%62 = add i32 %21, %61
%63 = trunc i64 %60 to i32
%64 = add i32 %17, %63
%65 = trunc i64 %60 to i32
%66 = add i32 %15, %65
%67 = icmp sgt i32 %66, 2147483616
%68 = icmp sgt i32 %64, 2147483616
%69 = or i1 %67, %68
%70 = icmp sgt i32 %62, 2147483616
%71 = or i1 %69, %70
br i1 %71, label %prejoin for_entry.entry.i.us.preheader, label
```

```
vector.memcheck:
%bound033 = icmp ult float*, %$scevgevp, %$scevgevp21
%bound134 = icmp ult float*, %$scevgevp18, %$scevgevp11
%conflict35 = and i1 %$bound033, %$bound134
%$conflict_rdx = or i1 %$conflict_rdx, %$conflict35
%conflict_rdx39 = or i1 %$conflict_rdx, %$conflict38
%bound040 = icmp ult float*, %$scevgevp, %$scevgevp31
%bound141 = icmp ult float*, %$scevgevp28, %$scevgevp11
%conflict42 = and i1 %$bound040, %$bound141
%conflict_rdx43 = or i1 %$conflict_rdx39, %$conflict42
br i1 %$conflict_rdx43, label %region_for_entry_into_x.preheader, label
%%_vector.body
```

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

```

bodyector:
%762 = add i32 %mul1, %c38
%773 = sext i32 %772 to i64
%774 = getelementptr inbounds float, float* %2, i64 %773
%775 = bitcast float* %774 to <8 x float>*
%wide_masked.load.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... float> %775, i32 4, <8 x i1> %c37, <8 x float> undef), tbaa i12, ialias.scope
... i16, inoalias i19
%776 = fpxext <8 x float> %wide_masked.load.3 to <8 x double>
%777 = add i32 %72, 1
%778 = sext i32 %777 to i64
%779 = getelementptr inbounds float, float* %0, i64 %778
%808 = bitcast float* %779 to <8 x float>*
%wide_masked.load46 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... float> %808, i32 4, <8 x i1> %c37, <8 x float> undef), tbaa i12, ialias.scope
... i124
%810 = getelementptr inbounds float, float* %0, i64 %73
%882 = bitcast float* %810 to <8 x float>*
%883 = bitcast float* %882 to <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... float> %883, i32 4, <8 x i1> %c37, <8 x float> undef), tbaa i12, ialias.scope
... i125
%884 = add nsw i32 %mul19 i1, %c38
%885 = sext i32 %884 to i64
%886 = getelementptr inbounds float, float* %1, i64 %85
%887 = bitcast float* %886 to <8 x float>*
%wide_masked.load48 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... float> %887, i32 4, <8 x i1> %c37, <8 x float> undef), tbaa i12, ialias.scope
... i126
%889 = fadd <8 x float> %83, %wide_masked.load48
%889 = getelementptr inbounds float, float* %1, i64 %73
%909 = bitcast float* %889 to <8 x float>*
%wide_masked.load49 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... float> %909, i32 4, <8 x i1> %c37, <8 x float> undef), tbaa i12, ialias.scope
... i127
%911 = fsub <8 x float> %88, %wide_masked.load49
%929 = fpxext <8 x float> %911 to <8 x double>
%931 = call <8 x double> @llvm.fmuladd.v8f64<8 x double> %92, <8 x double>
<double 0xBF6E666666666666, double 0xBF6E666666666666, double
0xBF6E666666666666, double 0xBF6E666666666666, double 0xBF6E666666666666,
double 0xBF6E666666666666, double 0xBF6E666666666666, double
0xBF6E666666666666>, <8 x double> %976)
%934 = fptrunc <8 x double> %931 to <8 x float>
%935 = bitcast float* %774 to <8 x float>*
%wide_masked.store.v8f32.p0v8f32<8 x float> %934, <8 x float>*
... %935, i32 4, <8 x i1> %c37), tbaa i12, ialias.scope i16, inoalias i19,
... llvm.access.group i28
%936 = add i32 %mul1, %c42
%937 = sext i32 %936 to i64
%938 = getelementptr inbounds float, float* %2, i64 %97
%939 = bitcast float* %938 to <8 x float>*
%wide_masked.load.v8f32.p0v8f32<8 x float> %939, i32 4, <8 x i1> %c41, <8 x float> undef), tbaa i12, ialias.scope
... i116, inoalias i19
%1000 = fpxext <8 x float> %wide_masked.load.1 to <8 x double>
%1001 = add i32 %90, 1
%1002 = sext i32 %1001 to i64
%1003 = getelementptr inbounds float, float* %0, i64 %102
%1004 = bitcast float* %1003 to <8 x float>*
%wide_masked.load103 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... x float> %1004, i32 4, <8 x i1> %c41, <8 x float> undef), tbaa i12,
... ialias.scope i124
%1005 = getelementptr inbounds float, float* %0, i64 %97
%1006 = bitcast float* %1005 to <8 x float>*
%wide_masked.load47.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... x float> %1006, i32 4, <8 x i1> %c41, <8 x float> undef), tbaa i12,
... llvm.access.group i28
%1007 = fsub <8 x float> %wide_masked.load46.1, %wide_masked.load47.1
%1008 = add nsw i32 %mul19 i1, %c42
%1009 = sext i32 %1008 to i64
%1010 = getelementptr inbounds float, float* %1, i64 %109
%1011 = bitcast float* %1010 to <8 x float>*
%wide_masked.load48.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... x float> %1011, i32 4, <8 x i1> %c41, <8 x float> undef), tbaa i12,
... ialias.scope i126
%1012 = fadd <8 x float> %107, %wide_masked.load48.1
%1013 = getelementptr inbounds float, float* %1, i64 %97
%1014 = bitcast float* %1013 to <8 x float>*
%wide_masked.load49.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... x float> %1014, i32 4, <8 x i1> %c41, <8 x float> undef), tbaa i12,
... ialias.scope i127
%1015 = fsub <8 x float> %112, %wide_masked.load49.1
%1016 = fpxext <8 x float> %1015 to <8 x double>
%1017 = call <8 x double> @llvm.fmuladd.v8f64<8 x double> %116, <8 x double>
<double 0xBF6E666666666666, double 0xBF6E666666666666, double
0xBF6E666666666666, double 0xBF6E666666666666, double 0xBF6E666666666666,
double 0xBF6E666666666666, double 0xBF6E666666666666, double
0xBF6E666666666666>, <8 x double> %1000)
%1018 = fptrunc <8 x double> %1017 to <8 x float>
%1019 = bitcast float* %98 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32<8 x float> %1018, <8 x float>*
... %1019, i32 4, <8 x i1> %c41), tbaa i12, ialias.scope i16, inoalias i19,
... llvm.access.group i28
%1020 = add i32 %mul1, %c46
%1021 = sext i32 %1020 to i64
%1022 = getelementptr inbounds float, float* %2, i64 %121
%1023 = bitcast float* %1022 to <8 x float>*
%wide_masked.load.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... float> %1023, i32 4, <8 x i1> %c45, <8 x float> undef), tbaa i12,
... ialias.scope i116, inoalias i19
%1024 = fpxext <8 x float> %wide_masked.load.2 to <8 x double>
%1025 = add i32 %162, 1
%1026 = sext i32 %1025 to i64
%1027 = getelementptr inbounds float, float* %0, i64 %126
%1028 = bitcast float* %1027 to <8 x float>*
%wide_masked.load46.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... x float> %1028, i32 4, <8 x i1> %c45, <8 x float> undef), tbaa i12,
... ialias.scope i124
%1029 = getelementptr inbounds float, float* %0, i64 %121
%1030 = bitcast float* %1029 to <8 x float>*
%wide_masked.load47.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... x float> %1030, i32 4, <8 x i1> %c45, <8 x float> undef), tbaa i12,
... ialias.scope i125
%1031 = fsub <8 x float> %wide_masked.load46.2, %wide_masked.load47.2
%1032 = add nsw i32 %mul19 i1, %c46
%1033 = sext i32 %1032 to i64
%1034 = getelementptr inbounds float, float* %1, i64 %133
%1035 = bitcast float* %1034 to <8 x float>*
%1036 = bitcast float* %1034 to <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... x float> %1035, i32 4, <8 x i1> %c45, <8 x float> undef), tbaa i12,
... ialias.scope i126
%1036 = fadd <8 x float> %131, %wide_masked.load48.2
%1037 = getelementptr inbounds float, float* %1, i64 %121
%1038 = bitcast float* %137 to <8 x float>*
%wide_masked.load49.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... x float> %1038, i32 4, <8 x i1> %c45, <8 x float> undef), tbaa i12,
... ialias.scope i127
%1039 = fsub <8 x float> %136, %wide_masked.load49.2
%1040 = fpxext <8 x float> %139 to <8 x double>
%1041 = call <8 x double> @llvm.fmuladd.v8f64<8 x double> %140, <8 x double>
<double 0xBF6E666666666666, double 0xBF6E666666666666, double
0xBF6E666666666666, double 0xBF6E666666666666, double 0xBF6E666666666666,
double 0xBF6E666666666666, double 0xBF6E666666666666, double
0xBF6E666666666666>, <8 x double> %141)
%1042 = fptrunc <8 x double> %141 to <8 x float>
%1043 = bitcast float* %122 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32<8 x float> %142, <8 x float>*
... %142, <8 x float> %143, i32 4, <8 x i1> %c45), tbaa i12, i
```

```

region for entry.entry.i.us :
% local id x_0.us = phi i64 [%i74, %if.end_r_exit.i.us ], [ 0, ...
%pred1 for entry.entry.i.us.preheader ]
%addl.i.i.us = add nuw nsw i64 %local_id_x_0.us, %mul.i.i
%conv.i.us = trunc i64 %addl.i.i.us to i32
%cmp5.i.us = icmp sgt i32 %sub4.i, %conv.i.us
br i1 %cmp5.i.us, label %if.then.i.us, label %if.end_r_exit.i.us

      T                               F

```

```

t1b0n1.i.us;
%add12.i.us = add i32 %mul1.i, %conv1.us
%idxprom1.i.us = sext i32 %add12.i.us to i64
%getelemntpr1.i.us = getelemntpr inbounds float, float* %2, i64 %idxprom1.i.us
%l16168 = load float, float* %arrayidx22.i.us, align 4, t1b0a !12
%convn71.i.us = fpevt float %l16168 to double
%add1010.i.us = add i32 %add12.i.us, 1
%idxprom11.i.us = sext i32 %add10.i.us to i64
%arrayidx12.i.us = getelemntpr inbounds float, float* %0, i64
... %idxprom11.i.us
%l16169 = load float, float* %arrayidx12.i.us, align 4, t1b0a !12
%getelemntpr16.i.us = getelemntpr inbounds float, float* %0, i64 %idxprom11.i.us
%l170 = load float, float* %arrayidx16.i.us, align 4, t1b0a !12
%fsub171.i.us = fsub float %mul169, %l170
%add2010.i.us = add nsw i32 %mul191, %conv1.us
%idxprom21.i.us = sext i32 %add20.i.us to i64
%arrayidx22.i.us = getelemntpr inbounds float, float* %1, i64
... %idxprom21.i.us
%l171 = load float, float* %arrayidx22.i.us, align 4, t1b0a !12
%add223.i.us = fadd float %fsub171.i.us, %l171
%arrayidx27.i.us = getelemntpr inbounds float, float* %0, i64 %idxprom1.i.us
%l172 = load float, float* %arrayidx27.i.us, align 4, t1b0a !12
%fsub28.i.us = fsub float %mul27, %l172
%convn29.i.us = fpevt float %fsub28.i.us to double
%l173 = tail call double @llvm.fmuladd.f64(double %convn29.i.us, double
... 0xBFDE666666666666, double %conv1.i.us) #5
%convn31.i.us = fptrunc double %l173 to float
store float %convn31.i.us, float* %arrayidx.i.us, align 4, t1b0a !12,
... @llvm.access_group 128

```

```
if.end_r_exit.i.us:
%174 = add nuw nsw i64 % local_id_x.0.us, 1
%exitcond.not = icmp eq i64 %174, 32
br i1 %exitcond.not, label %region_for_end.i.loopexit, label
... %region_for_entry.entry.i.us, !lvm.loop !31
```

```

    if ! %exitcond2.not,
      %pregion_for_entry.

```