

```
%9:
%mul1.i = shl i64 %6, 5
%mul3.i.i = shl i64 %7, 3
%sub1 = add nsw i32 %3, -1
%sub4.i = add nsw i32 %4, -1
%10 = trunc i64 %7 to i32
%11 = add i32 %12, %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
%scvgevp10 = getelementptr float, float* %2, i64 32
%scvgevp15 = getelementptr float, float* %0, i64 32
%29 = or i32 %27, 1
%30 = zext i32 %29 to i64
%scvgevp20 = getelementptr float, float* %0, i64 32
%scvgevp25 = getelementptr float, float* %1, i64 32
%31 = shl i32 %22, 3
%32 = or i32 %31, 1
%33 = mul i32 %32, %4
%34 = add i32 %33, %26
%scvgevp30 = getelementptr float, float* %1, i64 32
%bound0 = icmp ugt float* %scvgevp15, %2
%bound1 = (icmp ugt float* %scvgevp10, %0
%found_conflict = and i1 %bound0, %bound1
%bound036 = icmp ugt float* %scvgevp25, %2
%bound137 = icmp ugt float* %scvgevp10, %0
%found_conflict38 = and i1 %bound036, %bound137
%broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul1.i, i32 0
%broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
... undef, <8 x i32> zeroinitializer
%brnroadcast.splatinsert4 = insertelement <8 x i32> undef, i32 %sub4.i, i32 0
%broadcast.splat45 = shufflevector <8 x i32> %broadcast.splatinsert44, <8 x
... i32> undef, <8 x i32> zeroinitializer
%35 = trunc <8 x i64> %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 sgt <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 <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 sgt <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 sgt <8 x i32> %broadcast.splat45, %48
%50 = extractelement <8 x i32> %48, i32 0
br i1 %cmp.i, label %region_for_entry.region_for_init.i
```

```
region_for_entry.region_for_init.i:
% local_id.y0 = phi i64 [ 0, %9 ], [ %175, %region_for_end.i ]
%51 = mul i64 % local_id.y0, %28
%52 = trunc i64 %51 to i32
%53 = add i32 %27, %52
%54 = sext i32 %53 to i64
%scvgevp = getelementptr float, float* %2, i64 %54
%scvgevp1 = getelementptr float, float* %scvgevp10, i64 %54
%55 = add i64 %51, %30
%sext = shl i64 %55, 32
%56 = ashr exact i64 %sext, 32
%scvgevp18 = getelementptr float, float* %0, i64 %56
%scvgevp21 = getelementptr float, float* %scvgevp20, i64 %56
%57 = trunc i64 %51 to i32
%58 = add i32 %34, %57
%59 = sext i32 %58 to i64
%scvgevp28 = getelementptr float, float* %1, i64 %59
%scvgevp31 = getelementptr float, float* %scvgevp30, i64 %59
%60 = mul i64 % local_id.y0, %16
%add6.i.i = add nuw nsw i64 % local_id.y0, %mul3.i.i
%conv2.i = trunc i64 %add6.i.i to i32
%cmp.i = icmp sgt i32 %sub.i, %conv2.i
%mul1.i = mul nsw i32 %conv2.i, %4
%add18.i = add nsw i32 %conv2.i, 1
%mul19.i = mul nsw i32 %add18.i, %4
br i1 %cmp.i, label %vector.scevcheck, label %region_for_end.i
```

```
vector.scevcheck:
%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, %66
%70 = icmp sgt i32 %62, 2147483616
%71 = or i1 %69, %70
br i1 %71, label %region_for_entry.entry.i.us.preheader, label
... %vector.memcheck
```

```
vector.memcheck:
%bound033 = icmp ult float* %scvgevp, %scvgevp21
%bound134 = icmp ult float* %scvgevp18, %scvgevp11
%found_conflict35 = and i1 %bound033, %bound134
%conflict_rdx = or i1 %found_conflict, %found_conflict35
%conflict_rdx39 = or i1 %conflict_rdx, %found_conflict38
%bound040 = icmp ult float* %scvgevp, %scvgevp31
%bound141 = icmp ult float* %scvgevp28, %scvgevp11
%found_conflict42 = and i1 %bound040, %bound141
%conflict_rdx43 = or i1 %conflict_rdx39, %found_conflict42
br i1 %conflict_rdx43, label %region_for_entry.entry.i.us.preheader, label
... %vector.body
```

region_for_entry.entry.i.us.preheader:
br label %region_for_entry.entry.i.us

```
vector.body:
%72 = add i32 %mul1, %38
%73 = sext i32 %72 to i64
%74 = getelementptr inbounds float, float* %2, i64 %73
%75 = bitcast float* %74 to <8 x float>*
%wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %75, i32 4, <8 x i1> %37, <8 x float> undef), !tbaa !12, !alias.scope
... !16, !noalias !19
%76 = fpxext <8 x float> %wide.masked.load to <8 x double>
%77 = add i32 %72, 1
%78 = sext i32 %77 to i64
%79 = getelementptr inbounds float, float* %0, i64 %78
%80 = bitcast float* %79 to <8 x float>*
%wide.masked.load46 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %80, i32 4, <8 x i1> %37, <8 x float> undef), !tbaa !12, !alias.scope
... !24
%81 = getelementptr inbounds float, float* %0, i64 %73
%82 = bitcast float* %81 to <8 x float>*
%wide.masked.load47 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %82, i32 4, <8 x i1> %37, <8 x float> undef), !tbaa !12, !alias.scope
... !25
%83 = fsub <8 x float> %wide.masked.load46, %wide.masked.load47
%84 = add nsw i32 %mul19.i, %38
%85 = sext i32 %84 to i64
%86 = getelementptr inbounds float, float* %1, i64 %85
%87 = bitcast float* %86 to <8 x float>*
%wide.masked.load48 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %87, i32 4, <8 x i1> %37, <8 x float> undef), !tbaa !12, !alias.scope
... !26
%88 = fadd <8 x float> %83, %wide.masked.load48
%89 = getelementptr inbounds float, float* %1, i64 %73
%90 = bitcast float* %89 to <8 x float>*
%wide.masked.load49 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %90, i32 4, <8 x i1> %37, <8 x float> undef), !tbaa !12, !alias.scope
... !27
%91 = fsub <8 x float> %88, %wide.masked.load49
%92 = fpxext <8 x float> %91 to <8 x double>
%93 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %92, <8 x double>
... <double 0xBFEE666666666666, double 0xBFEE666666666666, double
... 0xBFEE666666666666, double 0xBFEE666666666666, double 0xBFEE666666666666,
... double 0xBFEE666666666666, double 0xBFEE666666666666, double
... 0xBFEE666666666666>, <8 x double> %76)
%94 = fptrunc <8 x double> %93 to <8 x float>
%95 = bitcast float* %74 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %94, <8 x float>*
... %95, i32 4, <8 x i1> %37), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !28
%96 = add i32 %mul1, %42
%97 = sext i32 %96 to i64
%98 = getelementptr inbounds float, float* %2, i64 %97
%99 = bitcast float* %98 to <8 x float>*
%wide.masked.load.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %99, i32 4, <8 x i1> %41, <8 x float> undef), !tbaa !12, !alias.scope
... !16, !noalias !19
%100 = fpxext <8 x float> %wide.masked.load.1 to <8 x double>
%101 = add i32 %96, %42
%102 = sext i32 %101 to i64
%103 = getelementptr inbounds float, float* %0, i64 %102
%104 = bitcast float* %103 to <8 x float>*
%wide.masked.load46.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %104, i32 4, <8 x i1> %41, <8 x float> undef), !tbaa !12,
... !alias.scope !24
%105 = getelementptr inbounds float, float* %0, i64 %97
%106 = bitcast float* %105 to <8 x float>*
%wide.masked.load47.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %106, i32 4, <8 x i1> %41, <8 x float> undef), !tbaa !12,
... !alias.scope !25
%107 = fsub <8 x float> %wide.masked.load46.1, %wide.masked.load47.1
%108 = add nsw i32 %mul19.i, %42
%109 = sext i32 %108 to i64
%110 = getelementptr inbounds float, float* %1, i64 %109
%111 = bitcast float* %110 to <8 x float>*
%wide.masked.load48.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %111, i32 4, <8 x i1> %41, <8 x float> undef), !tbaa !12,
... !alias.scope !26
%112 = fadd <8 x float> %107, %wide.masked.load48.1
%113 = getelementptr inbounds float, float* %1, i64 %97
%114 = bitcast float* %113 to <8 x float>*
%wide.masked.load49.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %114, i32 4, <8 x i1> %41, <8 x float> undef), !tbaa !12,
... !alias.scope !27
%115 = fsub <8 x float> %112, %wide.masked.load49.1
%116 = fpxext <8 x float> %115 to <8 x double>
%117 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %116, <8 x double>
... <double 0xBFEE666666666666, double 0xBFEE666666666666, double
... 0xBFEE666666666666, double 0xBFEE666666666666, double 0xBFEE666666666666,
... double 0xBFEE666666666666, double 0xBFEE666666666666, double
... 0xBFEE666666666666>, <8 x double> %100)
%118 = fptrunc <8 x double> %117 to <8 x float>
%119 = bitcast float* %98 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %118, <8 x float>*
... %119, i32 4, <8 x i1> %41), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !28
%120 = add i32 %mul1, %46
%121 = sext i32 %120 to i64
%122 = getelementptr inbounds float, float* %2, i64 %121
%123 = bitcast float* %122 to <8 x float>*
%wide.masked.load47.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %123, i32 4, <8 x i1> %45, <8 x float> undef), !tbaa !12,
... !alias.scope !16, !noalias !19
%124 = fpxext <8 x float> %wide.masked.load.2 to <8 x double>
%125 = add i32 %120, 1
%126 = sext i32 %125 to i64
%127 = getelementptr inbounds float, float* %0, i64 %126
%128 = bitcast float* %127 to <8 x float>*
%wide.masked.load46.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %128, i32 4, <8 x i1> %45, <8 x float> undef), !tbaa !12,
... !alias.scope !24
%129 = getelementptr inbounds float, float* %0, i64 %121
%130 = bitcast float* %129 to <8 x float>*
%wide.masked.load47.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %130, i32 4, <8 x i1> %45, <8 x float> undef), !tbaa !12,
... !alias.scope !25
%131 = fsub <8 x float> %wide.masked.load46.2, %wide.masked.load47.2
%132 = add nsw i32 %mul19.i, %46
%133 = sext i32 %132 to i64
%134 = getelementptr inbounds float, float* %1, i64 %133
%135 = bitcast float* %134 to <8 x float>*
%wide.masked.load48.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %135, i32 4, <8 x i1> %45, <8 x float> undef), !tbaa !12,
... !alias.scope !26
%136 = fadd <8 x float> %131, %wide.masked.load48.2
%137 = getelementptr inbounds float, float* %1, i64 %121
%138 = bitcast float* %137 to <8 x float>*
%wide.masked.load49.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %138, i32 4, <8 x i1> %45, <8 x float> undef), !tbaa !12,
... !alias.scope !27
%139 = fsub <8 x float> %136, %wide.masked.load49.2
%140 = fpxext <8 x float> %139 to <8 x double>
%141 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %140, <8 x double>
... <double 0xBFEE666666666666, double 0xBFEE666666666666, double
... 0xBFEE666666666666, double 0xBFEE666666666666, double 0xBFEE666666666666,
... double 0xBFEE666666666666, double 0xBFEE666666666666, double
... 0xBFEE666666666666>, <8 x double> %124)
%142 = fptrunc <8 x double> %141 to <8 x float>
%143 = bitcast float* %122 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %142, <8 x float>*
... %143, i32 4, <8 x i1> %45), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !28
%144 = add i32 %mul1, %50
%145 = sext i32 %144 to i64
%146 = getelementptr inbounds float, float* %2, i64 %145
%147 = bitcast float* %146 to <8 x float>*
%wide.masked.load.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %147, i32 4, <8 x i1> %49, <8 x float> undef), !tbaa !12,
... !alias.scope !16, !noalias !19
%148 = fpxext <8 x float> %wide.masked.load.3 to <8 x double>
%149 = add i32 %144, 1
%150 = sext i32 %149 to i64
%151 = getelementptr inbounds float, float* %0, i64 %150
%152 = bitcast float* %151 to <8 x float>*
%wide.masked.load46.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %152, i32 4, <8 x i1> %49, <8 x float> undef), !tbaa !12,
... !alias.scope !25
%153 = getelementptr inbounds float, float* %0, i64 %145
%154 = bitcast float* %153 to <8 x float>*
%wide.masked.load47.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %154, i32 4, <8 x i1> %49, <8 x float> undef), !tbaa !12,
... !alias.scope !26
%155 = fsub <8 x float> %wide.masked.load46.3, %wide.masked.load47.3
%156 = add nsw i32 %mul19.i, %50
%157 = sext i32 %156 to i64
%158 = getelementptr inbounds float, float* %1, i64 %157
%159 = bitcast float* %158 to <8 x float>*
%wide.masked.load48.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %159, i32 4, <8 x i1> %49, <8 x float> undef), !tbaa !12,
... !alias.scope !26
%160 = fadd <8 x float> %155, %wide.masked.load48.3
%161 = getelementptr inbounds float, float* %1, i64 %145
%162 = bitcast float* %161 to <8 x float>*
%wide.masked.load49.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %162, i32 4, <8 x i1> %49, <8 x float> undef), !tbaa !12,
... !alias.scope !27
%163 = fsub <8 x float> %160, %wide.masked.load49.3
%164 = fpxext <8 x float> %163 to <8 x double>
%165 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %164, <8 x double>
... <double 0xBFEE666666666666, double 0xBFEE666666666666, double
... 0xBFEE666666666666, double 0xBFEE666666666666, double 0xBFEE666666666666,
... double 0xBFEE666666666666, double 0xBFEE666666666666, double
... 0xBFEE666666666666>, <8 x double> %148)
%166 = fptrunc <8 x double> %165 to <8 x float>
%167 = bitcast float* %146 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %166, <8 x float>*
... %167, i32 4, <8 x i1> %49), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !28
br label %region_for_end.i
```

```
if.then.i.us:
%add1.i.us = add i32 %mul1, %conv1.us
%idxprom1.i.us = sext i32 %add1.i.us to i64
%arrayidx.i.us = getelementptr inbounds float, float* %2, i64 %idxprom1.i.us
%168 = load float, float* %arrayidx.i.us, align 4, !tbaa !12
%conv7.i.us = fpxext float %168 to double
%add20.i.us = add i32 %add1.i.us, %conv1.us
%idxprom11.i.us = sext i32 %add20.i.us to i64
%arrayidx12.i.us = getelementptr inbounds float, float* %0, i64
... %idxprom11.i.us
%169 = load float, float* %arrayidx12.i.us, align 4, !tbaa !12
%arrayidx16.i.us = getelementptr inbounds float, float* %0, i64 %idxprom1.i.us
%170 = load float, float* %arrayidx16.i.us, align 4, !tbaa !12
%sub17.i.us = fsub float %169, %170
%add201.us = add nsw i32 %mul19.i, %conv1.us
%idxprom21.i.us = sext i32 %add201.us to i64
%arrayidx22.i.us = getelementptr inbounds float, float* %1, i64
... %idxprom21.i.us
%171 = load float, float* %arrayidx22.i.us, align 4, !tbaa !12
%add23.i.us = fadd float %sub17.i.us, %171
%arrayidx27.i.us = getelementptr inbounds float, float* %1, i64 %idxprom1.i.us
%172 = load float, float* %arrayidx27.i.us, align 4, !tbaa !12
%sub28.i.us = fsub float %add23.i.us, %172
%conv29.i.us = fpxext float %sub28.i.us to double
%173 = tail call double @llvm.fmuladd.f64(double %conv29.i.us, double
... 0xBFEE666666666666, double %conv7.i.us) #5
%conv31.i.us = fptrunc double %173 to float
store float %conv31.i.us, float* %arrayidx.i.us, align 4, !tbaa !12,
... !llvm.access.group !28
br label %if.end.r.exit.i.us
```

```
if.end.r.exit.i.us:
%174 = add nuw nsw i64 % local_id.x0.us, 1
%exitcond.not = icmp eq i64 %174, 32
br i1 %exitcond.not, label %region_for_end.i.us, !llvm.loop !31
... %region_for_entry.entry.i.us, !llvm.loop !31
```

region_for_end.i.us:
br label %region_for_end.i.us

```
region_for_end.i:
%175 = add nuw nsw i64 % local_id.y0, 1
%exitcond2.not = icmp eq i64 %175, 8
br i1 %exitcond2.not, label %region_for_end.i.us, !llvm.loop !34
... %region_for_entry.region_for_init.i, !llvm.loop !34
```

ftrd kernel3.exit:
ret void