

```
%9:
%10 = shl i64 %6, 5
%11 = shl i64 %67, 3
%sub.i = add nsw i32 %3, -1
%sub4.i = add nsw i32 %4, -1
%12 = trunc i64 %7 to i32
%13 = mul i32 %12, %4
%14 = shl i32 %13, 3
%15 = trunc i64 %6 to i32
%16 = shl i32 %15, 5
%17 = add i32 %14, %16
%18 = zext i32 %4 to i64
%19 = or i32 %17, 1
%20 = shl i32 %12, 3
%21 = or i32 %20, 1
%22 = mul i32 %21, %4
%23 = add i32 %22, %16
%24 = trunc i64 %7 to i32
%25 = mul i32 %24, %4
%26 = shl i32 %25, 3
%27 = trunc i64 %6 to i32
%28 = shl i32 %27, 5
%29 = add i32 %26, %28
%30 = zext i32 %4 to i64
%scvgevp10 = getelementptr float, float* %2, i64 32
%scvgevp15 = getelementptr float, float* %0, i64 32
%31 = or i32 %29, 1
%32 = zext i32 %31 to i64
%scvgevp20 = getelementptr float, float* %0, i64 32
%scvgevp25 = getelementptr float, float* %1, i64 32
%33 = shl i32 %24, 3
%34 = or i32 %33, 1
%35 = mul i32 %34, %4
%36 = add i32 %35, %28
%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, %1
%found_conflict38 = and i1 %bound036, %bound137
%broadcast.splatinsert = insertelement <8 x i32> undef, i64 %10, i32 0
%broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
... undef, <8 x i32> zeroinitializer
%broadcast.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
%37 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%38 = add <8 x i32> %37, <i32 0, i32 1, i32 2, i32 3, i32 4, i32 5, i32 6,
... i32 7>
%39 = icmp sgt <8 x i32> %broadcast.splat45, %38
%40 = extractelement <8 x i32> %38, i32 0
%41 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%42 = add <8 x i32> %41, <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32
... 14, i32 15>
%43 = icmp sgt <8 x i32> %broadcast.splat45, %42
%44 = extractelement <8 x i32> %42, i32 0
%45 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%46 = add <8 x i32> %45, <i32 16, i32 17, i32 18, i32 19, i32 20, i32 21,
... i32 22, i32 23>
%47 = icmp sgt <8 x i32> %broadcast.splat45, %46
%48 = extractelement <8 x i32> %46, i32 0
%49 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%50 = add <8 x i32> %49, <i32 24, i32 25, i32 26, i32 27, i32 28, i32 29,
... i32 30, i32 31>
%51 = icmp sgt <8 x i32> %broadcast.splat45, %50
%52 = extractelement <8 x i32> %50, i32 0
br i1 %region, for_entry.region, for_init.i
```

```
pre:region for entry.region for init.i:
% local_id y.0 = phi i64 [ 0, %9 ], [ %179, %region_for_end.i ]
%53 = mul i64 % local_id y.0, %30
%54 = trunc i64 %53 to i32
%55 = add i32 %29, %54
%56 = sext i32 %55 to i64
%scvgevp = getelementptr float, float* %2, i64 %56
%scvgevp1 = getelementptr float, float* %scvgevp10, i64 %56
%57 = add i64 %53, %32
%sext = shl i64 %57, 32
%58 = ashr exact i64 %sext, 32
%scvgevp18 = getelementptr float, float* %0, i64 %58
%scvgevp21 = getelementptr float, float* %scvgevp20, i64 %58
%59 = trunc i64 %53 to i32
%60 = add i32 %36, %59
%61 = sext i32 %60 to i64
%scvgevp28 = getelementptr float, float* %1, i64 %61
%scvgevp31 = getelementptr float, float* %scvgevp30, i64 %61
%62 = mul i64 % local_id y.0, %18
%63 = add nsw nsw i64 % local_id y.0, %11
%conv2.i = trunc i64 %63 to i32
%cmp1 = icmp sgt i32 %sub.i, %conv2.i
%mul.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.scevgevp, label %region_for_end.i
```

```
vector.scevcheck:
%64 = trunc i64 %62 to i32
%65 = add i32 %23, %64
%66 = trunc i64 %62 to i32
%67 = add i32 %19, %66
%68 = trunc i64 %62 to i32
%69 = add i32 %17, %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
br i1 %74, 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
```

pre:region for entry.entry.i.us.preheader:  
br label %region\_for\_entry.entry.i.us

```
vector.body:
%75 = add i32 %mul.i, %40
%76 = sext i32 %75 to i64
%77 = getelementptr inbounds float, float* %2, i64 %76
%78 = bitcast float* %77 to <8 x float>*
%wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %78, i32 4, <8 x i1> %39, <8 x float> undef), !tbaa !12, !alias.scope
... !16, !noalias !19
%79 = fpxext <8 x float> %wide.masked.load to <8 x double>
%80 = add i32 %75, 1
%81 = sext i32 %80 to i64
%82 = getelementptr inbounds float, float* %0, i64 %81
%83 = bitcast float* %82 to <8 x float>*
%wide.masked.load46 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %83, i32 4, <8 x i1> %39, <8 x float> undef), !tbaa !12, !alias.scope
... !24
%84 = getelementptr inbounds float, float* %0, i64 %76
%85 = bitcast float* %84 to <8 x float>*
%wide.masked.load47 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %85, i32 4, <8 x i1> %39, <8 x float> undef), !tbaa !12, !alias.scope
... !25
%86 = fsub <8 x float> %wide.masked.load46, %wide.masked.load47
%87 = add nsw i32 %mul19.i, %40
%88 = sext i32 %87 to i64
%89 = getelementptr inbounds float, float* %1, i64 %88
%90 = bitcast float* %89 to <8 x float>*
%wide.masked.load48 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %90, i32 4, <8 x i1> %39, <8 x float> undef), !tbaa !12, !alias.scope
... !26
%91 = fadd <8 x float> %86, %wide.masked.load48
%92 = getelementptr inbounds float, float* %1, i64 %76
%93 = bitcast float* %92 to <8 x float>*
%wide.masked.load49 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %93, i32 4, <8 x i1> %39, <8 x float> undef), !tbaa !12, !alias.scope
... !27
%94 = fsub <8 x float> %91, %wide.masked.load49
%95 = fpxext <8 x float> %94 to <8 x double>
%96 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %95, <8 x double>
... <double 0xBF66666666666666, double 0xBF66666666666666, double
... 0xBF66666666666666, double 0xBF66666666666666, double 0xBF66666666666666,
... double 0xBF66666666666666, double 0xBF66666666666666, double
... 0xBF66666666666666, <8 x double> %79)
%97 = fptrunc <8 x double> %96 to <8 x float>
%98 = bitcast float* %77 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %97, <8 x float>*)
... %98, i32 4, <8 x i1> %39), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !28
%99 = add i32 %mul.i, %44
%100 = sext i32 %99 to i64
%101 = getelementptr inbounds float, float* %2, i64 %100
%102 = bitcast float* %101 to <8 x float>*
%wide.masked.load.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %102, i32 4, <8 x i1> %43, <8 x float> undef), !tbaa !12,
... !alias.scope !16, !noalias !19
%103 = fpxext <8 x float> %wide.masked.load.1 to <8 x double>
%104 = add i32 %99, 1
%105 = sext i32 %104 to i64
%106 = getelementptr inbounds float, float* %0, i64 %105
%107 = bitcast float* %106 to <8 x float>*
%wide.masked.load46.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %107, i32 4, <8 x i1> %43, <8 x float> undef), !tbaa !12,
... !alias.scope !24
%108 = getelementptr inbounds float, float* %0, i64 %100
%109 = bitcast float* %108 to <8 x float>*
%wide.masked.load47.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %109, i32 4, <8 x i1> %43, <8 x float> undef), !tbaa !12,
... !alias.scope !25
%110 = fsub <8 x float> %wide.masked.load46.1, %wide.masked.load47.1
%111 = add nsw i32 %mul19.i, %44
%112 = sext i32 %111 to i64
%113 = getelementptr inbounds float, float* %1, i64 %112
%114 = bitcast float* %113 to <8 x float>*
%wide.masked.load48.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %114, i32 4, <8 x i1> %43, <8 x float> undef), !tbaa !12,
... !alias.scope !26
%115 = fadd <8 x float> %110, %wide.masked.load48.1
%116 = getelementptr inbounds float, float* %1, i64 %100
%117 = bitcast float* %116 to <8 x float>*
%wide.masked.load49.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %117, i32 4, <8 x i1> %43, <8 x float> undef), !tbaa !12,
... !alias.scope !27
%118 = fsub <8 x float> %115, %wide.masked.load49.1
%119 = fpxext <8 x float> %118 to <8 x double>
%120 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %119, <8 x double>
... <double 0xBF66666666666666, double 0xBF66666666666666, double
... 0xBF66666666666666, double 0xBF66666666666666, double 0xBF66666666666666,
... double 0xBF66666666666666, double 0xBF66666666666666, double
... 0xBF66666666666666, <8 x double> %103)
%121 = fptrunc <8 x double> %120 to <8 x float>
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %121, <8 x float>*)
... !122, i32 4, <8 x i1> %43), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !28
%123 = add i32 %mul.i, %48
%124 = sext i32 %123 to i64
%125 = getelementptr inbounds float, float* %2, i64 %124
%126 = bitcast float* %125 to <8 x float>*
%wide.masked.load.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %126, i32 4, <8 x i1> %47, <8 x float> undef), !tbaa !12,
... !alias.scope !26
%127 = fpxext <8 x float> %wide.masked.load.2 to <8 x double>
%128 = add i32 %123, 1
%129 = sext i32 %128 to i64
%130 = getelementptr inbounds float, float* %0, i64 %129
%131 = bitcast float* %130 to <8 x float>*
%wide.masked.load46.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %131, i32 4, <8 x i1> %47, <8 x float> undef), !tbaa !12,
... !alias.scope !24
%132 = getelementptr inbounds float, float* %0, i64 %124
%133 = bitcast float* %132 to <8 x float>*
%wide.masked.load47.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %133, i32 4, <8 x i1> %47, <8 x float> undef), !tbaa !12,
... !alias.scope !25
%134 = fsub <8 x float> %wide.masked.load46.2, %wide.masked.load47.2
%135 = add nsw i32 %mul19.i, %48
%136 = sext i32 %135 to i64
%137 = getelementptr inbounds float, float* %1, i64 %136
%138 = bitcast float* %137 to <8 x float>*
%wide.masked.load48.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %138, i32 4, <8 x i1> %47, <8 x float> undef), !tbaa !12,
... !alias.scope !26
%139 = fadd <8 x float> %134, %wide.masked.load48.2
%140 = getelementptr inbounds float, float* %1, i64 %124
%141 = bitcast float* %140 to <8 x float>*
%wide.masked.load49.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %141, i32 4, <8 x i1> %47, <8 x float> undef), !tbaa !12,
... !alias.scope !27
%142 = fsub <8 x float> %139, %wide.masked.load49.2
%143 = fpxext <8 x float> %142 to <8 x double>
%144 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %143, <8 x double>
... <double 0xBF66666666666666, double 0xBF66666666666666, double
... 0xBF66666666666666, double 0xBF66666666666666, double 0xBF66666666666666,
... double 0xBF66666666666666, double 0xBF66666666666666, double
... 0xBF66666666666666, <8 x double> %127)
%145 = fptrunc <8 x double> %144 to <8 x float>
%146 = bitcast float* %125 to <8 x float>
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %145, <8 x float>*)
... !146, i32 4, <8 x i1> %47), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !28
%147 = add i32 %mul.i, %52
%148 = sext i32 %147 to i64
%149 = getelementptr inbounds float, float* %2, i64 %148
%150 = bitcast float* %149 to <8 x float>*
%wide.masked.load.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %150, i32 4, <8 x i1> %51, <8 x float> undef), !tbaa !12,
... !alias.scope !16, !noalias !19
%151 = fpxext <8 x float> %wide.masked.load.3 to <8 x double>
%152 = add i32 %147, 1
%153 = sext i32 %152 to i64
%154 = getelementptr inbounds float, float* %0, i64 %153
%155 = bitcast float* %154 to <8 x float>*
%wide.masked.load46.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %155, i32 4, <8 x i1> %51, <8 x float> undef), !tbaa !12,
... !alias.scope !25
%156 = getelementptr inbounds float, float* %0, i64 %148
%157 = bitcast float* %156 to <8 x float>*
%wide.masked.load47.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %157, i32 4, <8 x i1> %51, <8 x float> undef), !tbaa !12,
... !alias.scope !26
%158 = fsub <8 x float> %wide.masked.load46.3, %wide.masked.load47.3
%159 = add nsw i32 %mul19.i, %52
%160 = sext i32 %159 to i64
%161 = getelementptr inbounds float, float* %1, i64 %160
%162 = bitcast float* %161 to <8 x float>*
%wide.masked.load48.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %162, i32 4, <8 x i1> %51, <8 x float> undef), !tbaa !12,
... !alias.scope !26
%163 = fadd <8 x float> %158, %wide.masked.load48.3
%164 = getelementptr inbounds float, float* %1, i64 %148
%165 = bitcast float* %164 to <8 x float>*
%wide.masked.load49.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %165, i32 4, <8 x i1> %51, <8 x float> undef), !tbaa !12,
... !alias.scope !27
%166 = fsub <8 x float> %163, %wide.masked.load49.3
%167 = fpxext <8 x float> %166 to <8 x double>
%168 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %167, <8 x double>
... <double 0xBF66666666666666, double 0xBF66666666666666, double
... 0xBF66666666666666, double 0xBF66666666666666, double 0xBF66666666666666,
... double 0xBF66666666666666, double 0xBF66666666666666, double
... 0xBF66666666666666, <8 x double> %151)
%169 = fptrunc <8 x double> %168 to <8 x float>
%170 = bitcast float* %149 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %169, <8 x float>*)
... !170, i32 4, <8 x i1> %51), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !28
br label %region_for_end.i
```

```
if.then.i.us:
%add.i.us = add i32 %mul.i, %conv.i.us
%idxprom.i.us = sext i32 %add.i.us to i64
%arrayidx.i.us = getelementptr inbounds float, float* %2, i64 %idxprom.i.us
%172 = load float, float* %arrayidx.i.us, align 4, !tbaa !12
%conv7.i.us = fpxext float %172 to double
%add10.i.us = add i32 %add.i.us, 1
%idxprom11.i.us = sext i32 %add10.i.us to i64
%arrayidx12.i.us = getelementptr inbounds float, float* %0, i64
... %idxprom11.i.us
%173 = load float, float* %arrayidx12.i.us, align 4, !tbaa !12
%arrayidx16.i.us = getelementptr inbounds float, float* %0, i64 %idxprom.i.us
%174 = load float, float* %arrayidx16.i.us, align 4, !tbaa !12
%sub17.i.us = fsub float %173, %174
%add20.i.us = add nsw i32 %mul19.i, %conv.i.us
%idxprom21.i.us = sext i32 %add20.i.us to i64
%arrayidx22.i.us = getelementptr inbounds float, float* %1, i64
... %idxprom21.i.us
%175 = load float, float* %arrayidx22.i.us, align 4, !tbaa !12
%add23.i.us = fadd float %sub17.i.us, %175
%arrayidx27.i.us = getelementptr inbounds float, float* %1, i64 %idxprom.i.us
%sub28.i.us = fsub float %add23.i.us, %176
%conv29.i.us = fpxext float %sub28.i.us to double
%177 = tail call double @llvm.fmuladd.f64(double %conv29.i.us, double
... 0xBF66666666666666, double %conv7.i.us) #5
%conv31.i.us = fptrunc double %177 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:
%178 = add nsw nsw i64 % local_id x.0.us, 1
%exitcond = icmp eq i64 %178, 32
br i1 %exitcond, label %region_for_end.i.us.preheader, label
... %region_for_entry.entry.i.us, !llvm.loop !31
```

pre:region\_for\_end.i.us.preheader:  
br label %region\_for\_end.i.us

```
pre:region_for_end.i.us:
%179 = add nsw nsw i64 % local_id y.0, 1
%exitcond2 = icmp eq i64 %179, 8
br i1 %exitcond2, label %ftd kernel3.exit, label
... %region_for_entry.region_for_init.i, !llvm.loop !34
```

ftd kernel3.exit:  
ret void