

```
%7:
%mul3.i.i = shl i64 %5, 3
%mul1.i = shl i64 %4, 5
%sub.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 %9, 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
%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
%broadcast.splatinsert16 = insertelement <8 x i32> undef, i32 %sub.i, i32 0
%broadcast.splat17 = shufflevector <8 x i32> %broadcast.splatinsert16, <8 x
i32> undef, <8 x i64> %broadcast.splat to <8 x i32>
%25 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%26 = or <8 x i32> %25, <i32 0, i32 1, i32 2, i32 3, i32 4, i32 5, i32 6,
... i32 7>
%27 = icmp sgt <8 x i32> %26, zeroinitializer
%28 = icmp sgt <8 x i32> %broadcast.splat17, %26
%29 = and <8 x i1> %28, %27
%30 = extractelement <8 x i32> %26, i32 0
%31 = trunc <8 x i32> %broadcast.splat to <8 x i32>
%32 = or <8 x i32> %31, <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32
... 14, i32 15>
%33 = icmp sgt <8 x i32> %32, zeroinitializer
%34 = icmp sgt <8 x i32> %broadcast.splat17, %32
%35 = and <8 x i1> %34, %33
%36 = extractelement <8 x i32> %32, i32 0
%37 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%38 = or <8 x i32> %37, <i32 16, i32 17, i32 18, i32 19, i32 20, i32 21, i32
... 22, i32 23>
%39 = icmp sgt <8 x i32> %38, zeroinitializer
%40 = icmp sgt <8 x i32> %broadcast.splat17, %38
%41 = and <8 x i1> %40, %39
%42 = extractelement <8 x i32> %38, i32 0
%43 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%44 = or <8 x i32> %43, <i32 24, i32 25, i32 26, i32 27, i32 28, i32 29, i32
... 30, i32 31>
%45 = icmp sgt <8 x i32> %44, zeroinitializer
%46 = icmp sgt <8 x i32> %broadcast.splat17, %44
%47 = and <8 x i1> %46, %45
%48 = extractelement <8 x i32> %44, i32 0
br label %region_for_entry.region_for_init.i
```

```
region_for_entry.region_for_init.i:
%local_id.y.0 = phi i64 [0, %7 ], [ %183, %region_for_end.i ]
%49 = mul i64 %local_id.y.0, %14
%add6.i.i = add nuw nsw i64 %local_id.y.0, %mul3.i.i, !llvm.access.group 112
%conv1 = icmp sgt i32 %conv1, 0, !llvm.access.group 112
%mul1.i = mul nsw i32 %conv1, %2
%add25.i = add nuw nsw i32 %conv1, 1
%mul26.i = mul nsw i32 %add25.i, %2
%sub3.i.i = add nsw i32 %conv1, -1
%mul32.i = mul nsw i32 %sub3.i.i, %2
%cmp4.i = icmp sgt i32 %sub.i, %conv1
%orcond = and i1 %cmp1.i, %cmp4.i
br i1 %orcond, label %vector.scevcheck, label %region_for_end.i
```

```
vector.scevcheck:
%50 = trunc i64 %49 to i32
%51 = add i32 %24, %50
%52 = trunc i64 %49 to i32
%53 = add i32 %21, %52
%54 = trunc i64 %49 to i32
%55 = add i32 %17, %54
%56 = trunc i64 %49 to i32
%57 = add i32 %16, %56
%58 = trunc i64 %49 to i32
%59 = add i32 %13, %58
%60 = icmp sgt i32 %59, 2147483616
%61 = icmp sgt i32 %57, 2147483616
%62 = or i1 %60, %61
%63 = icmp sgt i32 %55, 2147483616
%64 = or i1 %62, %63
%65 = icmp sgt i32 %53, 2147483616
%66 = or i1 %64, %65
%67 = icmp sgt i32 %51, 2147483616
%68 = or i1 %66, %67
br i1 %68, label %region_for_entry.entry.i.us.us.preheader, label
... %vector.body
```

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

```
vector.body:
%69 = add i32 %mul.i, %30, !llvm.access.group 112
%70 = sext i32 %69 to i64, !llvm.access.group 112
%71 = getelementptr inbounds float, float* %0, i64 %70, !llvm.access.group
... 112
%72 = bitcast float* %71 to <8 x float>*
%wide.load = load <8 x float>, <8 x float>* %72, align 4, !tbaa !15,
... !llvm.access.group 112
%73 = add i32 %69, -1, !llvm.access.group 112
%74 = sext i32 %73 to i64, !llvm.access.group 112
%75 = getelementptr inbounds float, float* %0, i64 %74, !llvm.access.group
... 112
%76 = bitcast float* %75 to <8 x float>*
%wide.load18 = load <8 x float>, <8 x float>* %76, align 4, !tbaa !15,
... !llvm.access.group 112
%77 = fadd <8 x float> %wide.load, %wide.load18, !llvm.access.group 112
%78 = add i32 %69, 1, !llvm.access.group 112
%79 = sext i32 %78 to i64, !llvm.access.group 112
%80 = getelementptr inbounds float, float* %0, i64 %79, !llvm.access.group
... 112
%81 = bitcast float* %80 to <8 x float>*
%wide.load19 = load <8 x float>, <8 x float>* %81, align 4, !tbaa !15,
... !llvm.access.group 112
%82 = fadd <8 x float> %77, %wide.load19, !llvm.access.group 112
%83 = add nsw i32 %mul26.i, %30, !llvm.access.group 112
%84 = sext i32 %83 to i64, !llvm.access.group 112
%85 = getelementptr inbounds float, float* %0, i64 %84, !llvm.access.group
... 112
%86 = bitcast float* %85 to <8 x float>*
%wide.load20 = load <8 x float>, <8 x float>* %86, align 4, !tbaa !15,
... !llvm.access.group 112
%87 = fadd <8 x float> %82, %wide.load20, !llvm.access.group 112
%88 = add nsw i32 %mul32.i, %30, !llvm.access.group 112
%89 = sext i32 %88 to i64, !llvm.access.group 112
%90 = getelementptr inbounds float, float* %0, i64 %89, !llvm.access.group
... 112
%91 = bitcast float* %90 to <8 x float>*
%wide.load21 = load <8 x float>, <8 x float>* %91, align 4, !tbaa !15,
... !llvm.access.group 112
%92 = fadd <8 x float> %87, %wide.load21, !llvm.access.group 112
%93 = fmul <8 x float> %92, <float 0x3FC99999A0000000, float
... 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
... 0x3FC99999A0000000>, !llvm.access.group 112
%94 = getelementptr inbounds float, float* %1, i64 %70, !llvm.access.group
... 112
%95 = bitcast float* %94 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %93, <8 x float>*
... %95, i32 4, <8 x i1> %29), !tbaa !15, !llvm.access.group 112
%96 = add i32 %mul.i, %36, !llvm.access.group 112
%97 = sext i32 %96 to i64, !llvm.access.group 112
%98 = getelementptr inbounds float, float* %0, i64 %97, !llvm.access.group
... 112
%99 = bitcast float* %98 to <8 x float>*
%wide.load.1 = load <8 x float>, <8 x float>* %99, align 4, !tbaa !15,
... !llvm.access.group 112
%100 = add i32 %96, -1, !llvm.access.group 112
%101 = sext i32 %100 to i64, !llvm.access.group 112
%102 = getelementptr inbounds float, float* %0, i64 %101, !llvm.access.group
... 112
%103 = bitcast float* %102 to <8 x float>*
%wide.load18.1 = load <8 x float>, <8 x float>* %103, align 4, !tbaa !15,
... !llvm.access.group 112
%104 = fadd <8 x float> %wide.load.1, %wide.load18.1, !llvm.access.group 112
%105 = add i32 %96, 1, !llvm.access.group 112
%106 = sext i32 %105 to i64, !llvm.access.group 112
%107 = getelementptr inbounds float, float* %0, i64 %106, !llvm.access.group
... 112
%108 = bitcast float* %107 to <8 x float>*
%wide.load19.1 = load <8 x float>, <8 x float>* %108, align 4, !tbaa !15,
... !llvm.access.group 112
%109 = fadd <8 x float> %104, %wide.load19.1, !llvm.access.group 112
%110 = add nsw i32 %mul26.i, %36, !llvm.access.group 112
%111 = sext i32 %110 to i64, !llvm.access.group 112
%112 = getelementptr inbounds float, float* %0, i64 %111, !llvm.access.group
... 112
%113 = bitcast float* %112 to <8 x float>*
%wide.load20.1 = load <8 x float>, <8 x float>* %113, align 4, !tbaa !15,
... !llvm.access.group 112
%114 = fadd <8 x float> %109, %wide.load20.1, !llvm.access.group 112
%115 = add nsw i32 %mul32.i, %36, !llvm.access.group 112
%116 = sext i32 %115 to i64, !llvm.access.group 112
%117 = getelementptr inbounds float, float* %0, i64 %116, !llvm.access.group
... 112
%118 = bitcast float* %117 to <8 x float>*
%wide.load21.1 = load <8 x float>, <8 x float>* %118, align 4, !tbaa !15,
... !llvm.access.group 112
%119 = fadd <8 x float> %114, %wide.load21.1, !llvm.access.group 112
%120 = fmul <8 x float> %119, <float 0x3FC99999A0000000, float
... 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
... 0x3FC99999A0000000>, !llvm.access.group 112
%121 = getelementptr inbounds float, float* %1, i64 %97, !llvm.access.group
... 112
%122 = bitcast float* %121 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %120, <8 x float>*
... %122, i32 4, <8 x i1> %35), !tbaa !15, !llvm.access.group 112
%123 = add i32 %mul.i, %42, !llvm.access.group 112
%124 = sext i32 %123 to i64, !llvm.access.group 112
%125 = getelementptr inbounds float, float* %0, i64 %124, !llvm.access.group
... 112
%126 = bitcast float* %125 to <8 x float>*
%wide.load.2 = load <8 x float>, <8 x float>* %126, align 4, !tbaa !15,
... !llvm.access.group 112
%127 = add i32 %123, -1, !llvm.access.group 112
%128 = sext i32 %127 to i64, !llvm.access.group 112
%129 = getelementptr inbounds float, float* %0, i64 %128, !llvm.access.group
... 112
%130 = bitcast float* %129 to <8 x float>*
%wide.load18.2 = load <8 x float>, <8 x float>* %130, align 4, !tbaa !15,
... !llvm.access.group 112
%131 = fadd <8 x float> %wide.load.2, %wide.load18.2, !llvm.access.group 112
%132 = add i32 %123, 1, !llvm.access.group 112
%133 = sext i32 %132 to i64, !llvm.access.group 112
%134 = getelementptr inbounds float, float* %0, i64 %133, !llvm.access.group
... 112
%135 = bitcast float* %134 to <8 x float>*
%wide.load19.2 = load <8 x float>, <8 x float>* %135, align 4, !tbaa !15,
... !llvm.access.group 112
%136 = fadd <8 x float> %131, %wide.load19.2, !llvm.access.group 112
%137 = add nsw i32 %mul26.i, %42, !llvm.access.group 112
%138 = sext i32 %137 to i64, !llvm.access.group 112
%139 = getelementptr inbounds float, float* %0, i64 %138, !llvm.access.group
... 112
%140 = bitcast float* %139 to <8 x float>*
%wide.load20.2 = load <8 x float>, <8 x float>* %140, align 4, !tbaa !15,
... !llvm.access.group 112
%141 = fadd <8 x float> %136, %wide.load20.2, !llvm.access.group 112
%142 = add nsw i32 %mul32.i, %42, !llvm.access.group 112
%143 = sext i32 %142 to i64, !llvm.access.group 112
%144 = getelementptr inbounds float, float* %0, i64 %143, !llvm.access.group
... 112
%145 = bitcast float* %144 to <8 x float>*
%wide.load21.2 = load <8 x float>, <8 x float>* %145, align 4, !tbaa !15,
... !llvm.access.group 112
%146 = fadd <8 x float> %141, %wide.load21.2, !llvm.access.group 112
%147 = fmul <8 x float> %146, <float 0x3FC99999A0000000, float
... 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
... 0x3FC99999A0000000>, !llvm.access.group 112
%148 = getelementptr inbounds float, float* %1, i64 %124, !llvm.access.group
... 112
%149 = bitcast float* %148 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %147, <8 x float>*
... %149, i32 4, <8 x i1> %41), !tbaa !15, !llvm.access.group 112
%150 = add i32 %mul.i, %48, !llvm.access.group 112
%151 = sext i32 %150 to i64, !llvm.access.group 112
%152 = getelementptr inbounds float, float* %0, i64 %151, !llvm.access.group
... 112
%153 = bitcast float* %152 to <8 x float>*
%wide.load.3 = load <8 x float>, <8 x float>* %153, align 4, !tbaa !15,
... !llvm.access.group 112
%154 = add i32 %150, -1, !llvm.access.group 112
%155 = sext i32 %154 to i64, !llvm.access.group 112
%156 = getelementptr inbounds float, float* %0, i64 %155, !llvm.access.group
... 112
%157 = bitcast float* %156 to <8 x float>*
%wide.load18.3 = load <8 x float>, <8 x float>* %157, align 4, !tbaa !15,
... !llvm.access.group 112
%158 = fadd <8 x float> %wide.load.3, %wide.load18.3, !llvm.access.group 112
%159 = add i32 %150, 1, !llvm.access.group 112
%160 = sext i32 %159 to i64, !llvm.access.group 112
%161 = getelementptr inbounds float, float* %0, i64 %160, !llvm.access.group
... 112
%162 = bitcast float* %161 to <8 x float>*
%wide.load19.3 = load <8 x float>, <8 x float>* %162, align 4, !tbaa !15,
... !llvm.access.group 112
%163 = fadd <8 x float> %158, %wide.load19.3, !llvm.access.group 112
%164 = add nsw i32 %mul26.i, %48, !llvm.access.group 112
%165 = sext i32 %164 to i64, !llvm.access.group 112
%166 = getelementptr inbounds float, float* %0, i64 %165, !llvm.access.group
... 112
%167 = bitcast float* %166 to <8 x float>*
%wide.load20.3 = load <8 x float>, <8 x float>* %167, align 4, !tbaa !15,
... !llvm.access.group 112
%168 = fadd <8 x float> %163, %wide.load20.3, !llvm.access.group 112
%169 = add nsw i32 %mul32.i, %48, !llvm.access.group 112
%170 = sext i32 %169 to i64, !llvm.access.group 112
%171 = getelementptr inbounds float, float* %0, i64 %170, !llvm.access.group
... 112
%172 = bitcast float* %171 to <8 x float>*
%wide.load21.3 = load <8 x float>, <8 x float>* %172, align 4, !tbaa !15,
... !llvm.access.group 112
%173 = fadd <8 x float> %168, %wide.load21.3, !llvm.access.group 112
%174 = fmul <8 x float> %173, <float 0x3FC99999A0000000, float
... 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
... 0x3FC99999A0000000>, !llvm.access.group 112
%175 = getelementptr inbounds float, float* %1, i64 %151, !llvm.access.group
... 112
%176 = bitcast float* %175 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %174, <8 x float>*
... %176, i32 4, <8 x i1> %47), !tbaa !15, !llvm.access.group 112
br label %region_for_end.i
```

```
if then.i.us.us:
%add1.i.us.us = add i32 %mul.i, %conv2.i.us.us, !llvm.access.group 112
%idxprom.i.us.us = sext i32 %add1.i.us.us to i64, !llvm.access.group 112
%arrayidx.i.us.us = getelementptr inbounds float, float* %0, i64
... %idxprom.i.us.us, !llvm.access.group 112
%177 = load float, float* %arrayidx.i.us.us, align 4, !tbaa !15,
... !llvm.access.group 112
%add15.i.us.us = add i32 %add1.i.us.us, -1, !llvm.access.group 112
%idxprom16.i.us.us = sext i32 %add15.i.us.us to i64, !llvm.access.group 112
%arrayidx17.i.us.us = getelementptr inbounds float, float* %0, i64
... %idxprom16.i.us.us, !llvm.access.group 112
%178 = load float, float* %arrayidx17.i.us.us, align 4, !tbaa !15,
... !llvm.access.group 112
%add18.i.us.us = fadd float %177, %178, !llvm.access.group 112
%add21.i.us.us = add i32 %add1.i.us.us, 1, !llvm.access.group 112
%idxprom22.i.us.us = sext i32 %add21.i.us.us to i64, !llvm.access.group 112
%arrayidx23.i.us.us = getelementptr inbounds float, float* %0, i64
... %idxprom22.i.us.us, !llvm.access.group 112
%179 = load float, float* %arrayidx23.i.us.us, align 4, !tbaa !15,
... !llvm.access.group 112
%add24.i.us.us = fadd float %add18.i.us.us, %179, !llvm.access.group 112
%add27.i.us.us = add nsw i32 %mul26.i, %conv2.i.us.us, !llvm.access.group 112
%idxprom28.i.us.us = sext i32 %add27.i.us.us to i64, !llvm.access.group 112
%arrayidx29.i.us.us = getelementptr inbounds float, float* %0, i64
... %idxprom28.i.us.us, !llvm.access.group 112
%180 = load float, float* %arrayidx29.i.us.us, align 4, !tbaa !15,
... !llvm.access.group 112
%add30.i.us.us = fadd float %add24.i.us.us, %180, !llvm.access.group 112
%add33.i.us.us = add nsw i32 %mul32.i, %conv2.i.us.us, !llvm.access.group 112
%idxprom34.i.us.us = sext i32 %add33.i.us.us to i64, !llvm.access.group 112
%arrayidx35.i.us.us = getelementptr inbounds float, float* %0, i64
... %idxprom34.i.us.us, !llvm.access.group 112
%181 = load float, float* %arrayidx35.i.us.us, align 4, !tbaa !15,
... !llvm.access.group 112
%add36.i.us.us = fadd float %add30.i.us.us, %181, !llvm.access.group 112
%add39.i.us.us = fmul float %add36.i.us.us, 0x3FC99999A0000000,
... !llvm.access.group 112
%arrayidx41.i.us.us = getelementptr inbounds float, float* %0, i64
... %idxprom.i.us.us, !llvm.access.group 112
store float %mul37.i.us.us, float* %arrayidx41.i.us.us, align 4, !tbaa !15,
... !llvm.access.group 112
br label %if_end.i.us.us, !llvm.access.group 112
```

```
if_end.i.us.us:
%182 = add nuw nsw i64 %local_id.x.0.us.us, 1
%exitcond.not = icmp eq i64 %182, 32
br i1 %exitcond.not, label %region_for_end.i.loopexit, label
... %region_for_entry.entry.i.us.us, !llvm.loop 19
```

region_for_end.i.loopexit:
br label %region_for_end.i

```
region_for_end.i:
%183 = add nuw nsw i64 %local_id.y.0, 1
%exitcond3.not = icmp eq i64 %183, 8
br i1 %exitcond3.not, label %runjacobi2D_kernel1.exit, label
... %region_for_entry.region_for_init.i, !llvm.loop 122
```

runjacobi2D_kernel1.exit:
ret void