

```
%7:
    %mul3.i.i = shl i64 %5, 3
    %mul.i.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 %6, 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
    %scevgep17 = getelementptr float, float* %1, i64 32
    %32 = shl i32 %25, 3
    %33 = add i32 %32, -1
    %34 = mul i32 %33, %2
    %35 = add i32 %34, %29
    %scevgep22 = getelementptr float, float* %0, i64 32
    %36 = or i32 %32, 1
    %37 = mul i32 %36, %2
    %38 = add i32 %37, %29
    %scevgep27 = getelementptr float, float* %0, i64 32
    %39 = or i32 %30, 1
    %40 = zext i32 %39 to i64
    %scevgep32 = getelementptr float, float* %0, i64 32
    %41 = add i32 %27, %29
    %42 = add i32 %41, -8
    %43 = or i32 %42, 7
    %scevgep37 = getelementptr float, float* %0, i64 32
    %scevgep42 = getelementptr float, float* %0, i64 32
    %bound056 = icmp ugt float* %scevgep42, %1
    %bound157 = icmp ugt float* %scevgep17, %0
    %found.conflict58 = and i1 %bound056, %bound157
    %broadcast.splatinert = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
    %broadcast.splat = shufflevector <8 x i64> %broadcast.splatinert60, <8 x i64>
    ..., undef, <8 x i32> zeroinitializer
    %broadcast.splatinert60 = insertelement <8 x i32> undef, i64 %mul.i.i, i32 0
    %broadcast.splat61 = shufflevector <8 x i32> %broadcast.splatinert60, <8 x i32>
    ..., undef, <8 x i32> zeroinitializer
    %44 = trunc <8 x i64> %broadcast.splat to <8 x i32>
    %45 = or <8 x i32> %44, <i32 0, i32 1, i32 2, i32 3, i32 4, i32 5, i32 6,
    ..., i32 7>
    %46 = icmp sgt <8 x i32> %45, zeroinitializer
    %47 = icmp sgt <8 x i32> %broadcast.splat61, %45
    %48 = and <8 x i1> %47, %46
    %49 = extractelement <8 x i32> %45, i32 0
    %50 = trunc <8 x i64> %broadcast.splat to <8 x i32>
    %51 = or <8 x i32> %50, <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32
    ..., 14, i32 15>
    %52 = icmp sgt <8 x i32> %51, zeroinitializer
    %53 = icmp sgt <8 x i32> %broadcast.splat61, %51
    %54 = and <8 x i1> %53, %52
    %55 = extractelement <8 x i32> %51, i32 0
    %56 = trunc <8 x i64> %broadcast.splat to <8 x i32>
    %57 = or <8 x i32> %56, <i32 16, i32 17, i32 18, i32 19, i32 20, i32 21, i32
    ..., 22, i32 23>
    %58 = icmp sgt <8 x i32> %57, zeroinitializer
    %59 = icmp sgt <8 x i32> %broadcast.splat61, %57
    %60 = and <8 x i1> %59, %58
    %61 = extractelement <8 x i32> %57, i32 0
    %62 = trunc <8 x i64> %broadcast.splat to <8 x i32>
    %63 = or <8 x i32> %62, <i32 24, i32 25, i32 26, i32 27, i32 28, i32 29, i32
    ..., 30, i32 31>
    %64 = icmp sgt <8 x i32> %63, zeroinitializer
    %65 = icmp sgt <8 x i32> %broadcast.splat61, %63
    %66 = and <8 x i1> %65, %64
    %67 = extractelement <8 x i32> %63, i32 0
    br label %pregion_for_entry.pregion_for_init.i
```

```
pregion_for_entry.pregion_for_init.i:
    % local_id y0 = phi i64 [ 0, %7 ], [ %217, %pregion_for_end.i ]
    %68 = mul i64 % local_id y0, %31
    %69 = trunc i64 %68 to i32
    %70 = add i32 %30, %69
    %71 = sext i32 %70 to i64
    %scevgep = getelementptr float, float* %1, i64 %71
    %scevgep18 = getelementptr float, float* %scevgep17, i64 %71
    %72 = trunc i64 %68 to i32
    %73 = add i32 %35, %72
    %74 = sext i32 %73 to i64
    %scevgep20 = getelementptr float, float* %0, i64 %74
    %scevgep23 = getelementptr float, float* %scevgep22, i64 %74
    %75 = trunc i64 %68 to i32
    %76 = add i32 %38, %75
    %77 = sext i32 %76 to i64
    %scevgep25 = getelementptr float, float* %0, i64 %77
    %scevgep28 = getelementptr float, float* %scevgep27, i64 %77
    %78 = add i64 %68, %40
    %sext = shl i64 %78, 32
    %79 = ashr exact i64 %sext, 32
    %scevgep30 = getelementptr float, float* %0, i64 %79
    %scevgep33 = getelementptr float, float* %scevgep32, i64 %79
    %80 = trunc i64 %68 to i32
    %81 = add i32 %43, %80
    %82 = sext i32 %81 to i64
    %scevgep35 = getelementptr float, float* %0, i64 %82
    %scevgep38 = getelementptr float, float* %scevgep37, i64 %82
    %83 = mul i64 % local_id y0, %14
    %add6.i.i = add nsw nsw i64 % local_id y0, %mul3.i.i
    %conv.i = trunc i64 %add6.i.i to i32
    %cmp.i = icmp sgt i32 %conv.i, 0
    %mul.i = mul nsw i32 %conv.i, %2
    %add25.i = add nsw nsw i32 %conv.i, 1
    %mul26.i = mul nsw i32 %add25.i, %2
    %sub31.i = add nsw i32 %conv.i, -1
    %mul32.i = mul nsw i32 %sub31.i, %2
    %cmp4.i = icmp sgt i32 %sub.i, %conv.i
    %or.cond = and i1 %cmp.i, %cmp4.i
    br i1 %or.cond, label %vector.scevcheck, label %pregion_for_end.i
```

```
vector.scevcheck:
    %84 = trunc i64 %83 to i32
    %85 = add i32 %24, %84
    %86 = trunc i64 %83 to i32
    %87 = add i32 %21, %86
    %88 = trunc i64 %83 to i32
    %89 = add i32 %17, %88
    %90 = trunc i64 %83 to i32
    %91 = add i32 %16, %90
    %92 = trunc i64 %83 to i32
    %93 = add i32 %13, %92
    %94 = icmp sgt i32 %93, -1
    %95 = icmp sgt i32 %91, 2147483616
    %96 = or i1 %94, %95
    %97 = icmp sgt i32 %89, 2147483616
    %98 = or i1 %96, %97
    %99 = icmp sgt i32 %87, 2147483616
    %100 = or i1 %98, %99
    %101 = icmp sgt i32 %85, 2147483616
    %102 = or i1 %100, %101
    br i1 %102, label %pregion_for_entry.entry.i.us.us.preheader, label
    ... %vector.memcheck
```

```
vector.memcheck:
    %bound0 = icmp ult float* %scevgep, %scevgep23
    %bound1 = icmp ult float* %scevgep20, %scevgep18
    %found.conflict = and i1 %bound0, %bound1
    %bound045 = icmp ult float* %scevgep, %scevgep28
    %bound146 = icmp ult float* %scevgep25, %scevgep18
    %found.conflict47 = and i1 %bound045, %bound146
    %conflict.rdx = or i1 %found.conflict, %found.conflict47
    %bound048 = icmp ult float* %scevgep, %scevgep33
    %bound149 = icmp ult float* %scevgep30, %scevgep18
    %found.conflict50 = and i1 %bound048, %bound149
    %conflict.rdx51 = or i1 %conflict.rdx, %found.conflict50
    %bound052 = icmp ult float* %scevgep, %scevgep38
    %bound153 = icmp ult float* %scevgep35, %scevgep18
    %found.conflict54 = and i1 %bound052, %bound153
    %conflict.rdx55 = or i1 %conflict.rdx51, %found.conflict54
    %conflict.rdx59 = or i1 %conflict.rdx55, %found.conflict58
    br i1 %conflict.rdx59, label %pregion_for_entry.entry.i.us.us.preheader,
    ... label %vector.body
```

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

```
vector.body:
    %103 = add i32 %mul.i, %49
    %104 = sext i32 %103 to i64
    %105 = getelementptr inbounds float, float* %0, i64 %104
    %106 = bitcast float* %105 to <8 x float>*
    %wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
    ... float>* %106, i32 4, <8 x i1> %48, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !16
    %107 = add i32 %103, -1
    %108 = sext i32 %107 to i64
    %109 = getelementptr inbounds float, float* %0, i64 %108
    %110 = bitcast float* %109 to <8 x float>*
    %111 = bitcast float* %110 to <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
    ... float>* %110, i32 4, <8 x i1> %48, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !19
    %111 = fadd <8 x float> %wide.masked.load, %wide.masked.load62
    %112 = add i32 %103, 1
    %113 = sext i32 %112 to i64
    %114 = getelementptr inbounds float, float* %0, i64 %113
    %115 = bitcast float* %114 to <8 x float>*
    %wide.masked.load63 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
    ... float>* %115, i32 4, <8 x i1> %48, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !21
    %116 = fadd <8 x float> %111, %wide.masked.load63
    %117 = add nsw i32 %mul26.i, %49
    %118 = sext i32 %117 to i64
    %119 = getelementptr inbounds float, float* %0, i64 %118
    %120 = bitcast float* %119 to <8 x float>*
    %wide.masked.load64 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
    ... float>* %120, i32 4, <8 x i1> %48, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !23
    %121 = fadd <8 x float> %116, %wide.masked.load64
    %122 = add nsw i32 %mul32.i, %49
    %123 = sext i32 %122 to i64
    %124 = getelementptr inbounds float, float* %0, i64 %123
    %125 = bitcast float* %124 to <8 x float>*
    %wide.masked.load65 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
    ... float>* %125, i32 4, <8 x i1> %48, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !25
    %126 = fadd <8 x float> %121, %wide.masked.load65
    %127 = fmul <8 x float> %126, <float 0x3FC99999A0000000, float
    ..., 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
    ..., 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
    ..., 0x3FC99999A0000000>
    %128 = getelementptr inbounds float, float* %1, i64 %104
    %129 = bitcast float* %128 to <8 x float>*
    call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %127, <8 x float>*
    ..., !tbaa !12, i32 4, <8 x i1> %48), !tbaa !12, !alias.scope !27, !noalias !29,
    ..., !llvm.access.group !30
    %130 = add i32 %mul.i, %55
    %131 = sext i32 %130 to i64
    %132 = getelementptr inbounds float, float* %0, i64 %131
    %133 = bitcast float* %132 to <8 x float>*
    %wide.masked.load.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
    ... float>* %133, i32 4, <8 x i1> %54, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !16
    %134 = add i32 %130, -1
    %135 = sext i32 %134 to i64
    %136 = getelementptr inbounds float, float* %0, i64 %135
    %137 = bitcast float* %136 to <8 x float>*
    %wide.masked.load62.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
    ... x float>* %137, i32 4, <8 x i1> %54, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !19
    %138 = fadd <8 x float> %wide.masked.load.1, %wide.masked.load62.1
    %139 = add i32 %130, 1
    %140 = sext i32 %139 to i64
    %141 = getelementptr inbounds float, float* %0, i64 %140
    %142 = bitcast float* %141 to <8 x float>*
    %143 = bitcast float* %142 to <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
    ... x float>* %142, i32 4, <8 x i1> %54, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !21
    %143 = fadd <8 x float> %138, %wide.masked.load63.1
    %144 = add nsw i32 %mul26.i, %55
    %145 = sext i32 %144 to i64
    %146 = getelementptr inbounds float, float* %0, i64 %145
    %147 = bitcast float* %146 to <8 x float>*
    %wide.masked.load64.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
    ... x float>* %147, i32 4, <8 x i1> %54, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !23
    %148 = fadd <8 x float> %143, %wide.masked.load64.1
    %149 = add nsw i32 %mul32.i, %55
    %150 = sext i32 %149 to i64
    %151 = getelementptr inbounds float, float* %0, i64 %150
    %152 = bitcast float* %151 to <8 x float>*
    %wide.masked.load65.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
    ... x float>* %152, i32 4, <8 x i1> %54, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !25
    %153 = fadd <8 x float> %148, %wide.masked.load65.1
    %154 = fmul <8 x float> %153, <float 0x3FC99999A0000000, float
    ..., 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
    ..., 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
    ..., 0x3FC99999A0000000>
    %155 = getelementptr inbounds float, float* %1, i64 %131
    %156 = bitcast float* %155 to <8 x float>*
    call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %154, <8 x float>*
    ..., !tbaa !12, i32 4, <8 x i1> %54), !tbaa !12, !alias.scope !27, !noalias !29,
    ..., !llvm.access.group !30
    %157 = add i32 %mul.i, %61
    %158 = sext i32 %157 to i64
    %159 = getelementptr inbounds float, float* %0, i64 %158
    %160 = bitcast float* %159 to <8 x float>*
    %wide.masked.load.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
    ... float>* %160, i32 4, <8 x i1> %60, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !16
    %161 = add i32 %157, -1
    %162 = sext i32 %161 to i64
    %163 = getelementptr inbounds float, float* %0, i64 %162
    %164 = bitcast float* %163 to <8 x float>*
    %wide.masked.load62.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
    ... x float>* %164, i32 4, <8 x i1> %60, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !19
    %165 = fadd <8 x float> %wide.masked.load.2, %wide.masked.load62.2
    %166 = add i32 %157, 1
    %167 = sext i32 %166 to i64
    %168 = getelementptr inbounds float, float* %0, i64 %167
    %169 = bitcast float* %168 to <8 x float>*
    %wide.masked.load64.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
    ... x float>* %169, i32 4, <8 x i1> %60, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !21
    %170 = fadd <8 x float> %165, %wide.masked.load63.2
    %171 = add nsw i32 %mul26.i, %61
    %172 = sext i32 %171 to i64
    %173 = getelementptr inbounds float, float* %0, i64 %172
    %174 = bitcast float* %173 to <8 x float>*
    %wide.masked.load64.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
    ... x float>* %174, i32 4, <8 x i1> %60, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !23
    %175 = fadd <8 x float> %170, %wide.masked.load64.2
    %176 = add nsw i32 %mul32.i, %61
    %177 = sext i32 %176 to i64
    %178 = getelementptr inbounds float, float* %0, i64 %177
    %179 = bitcast float* %178 to <8 x float>*
    %wide.masked.load65.2 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
    ... x float>* %179, i32 4, <8 x i1> %60, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !25
    %180 = fadd <8 x float> %175, %wide.masked.load65.2
    %181 = fmul <8 x float> %180, <float 0x3FC99999A0000000, float
    ..., 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
    ..., 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
    ..., 0x3FC99999A0000000>
    %182 = getelementptr inbounds float, float* %1, i64 %158
    %183 = bitcast float* %182 to <8 x float>*
    call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %181, <8 x float>*
    ..., !tbaa !12, i32 4, <8 x i1> %60), !tbaa !12, !alias.scope !27, !noalias !29,
    ..., !llvm.access.group !30
    %184 = add i32 %mul.i, %67
    %185 = sext i32 %184 to i64
    %186 = getelementptr inbounds float, float* %0, i64 %185
    %187 = bitcast float* %186 to <8 x float>*
    %wide.masked.load.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
    ... float>* %187, i32 4, <8 x i1> %66, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !16
    %188 = add i32 %184, -1
    %189 = sext i32 %188 to i64
    %190 = getelementptr inbounds float, float* %0, i64 %189
    %191 = bitcast float* %190 to <8 x float>*
    %wide.masked.load62.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
    ... x float>* %191, i32 4, <8 x i1> %66, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !19
    %192 = fadd <8 x float> %wide.masked.load.3, %wide.masked.load62.3
    %193 = add i32 %184, 1
    %194 = sext i32 %193 to i64
    %195 = getelementptr inbounds float, float* %0, i64 %194
    %196 = bitcast float* %195 to <8 x float>*
    %wide.masked.load63.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
    ... x float>* %196, i32 4, <8 x i1> %66, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !21
    %197 = fadd <8 x float> %192, %wide.masked.load63.3
    %198 = add nsw i32 %mul26.i, %67
    %199 = sext i32 %198 to i64
    %200 = getelementptr inbounds float, float* %0, i64 %199
    %201 = bitcast float* %200 to <8 x float>*
    %wide.masked.load64.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
    ... x float>* %201, i32 4, <8 x i1> %66, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !23
    %202 = fadd <8 x float> %197, %wide.masked.load64.3
    %203 = add nsw i32 %mul32.i, %67
    %204 = sext i32 %203 to i64
    %205 = getelementptr inbounds float, float* %0, i64 %204
    %206 = bitcast float* %205 to <8 x float>*
    %wide.masked.load65.3 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
    ... x float>* %206, i32 4, <8 x i1> %66, <8 x float> undef), !tbaa !12,
    ..., !alias.scope !25
    %207 = fadd <8 x float> %202, %wide.masked.load65.3
    %208 = fmul <8 x float> %207, <float 0x3FC99999A0000000, float
    ..., 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
    ..., 0x3FC99999A0000000, float 0x3FC99999A0000000, float 0x3FC99999A0000000, float
    ..., 0x3FC99999A0000000>
    %209 = getelementptr inbounds float, float* %1, i64 %185
    %210 = bitcast float* %209 to <8 x float>*
    call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %208, <8 x float>*
    ..., !tbaa !12, i32 4, <8 x i1> %66), !tbaa !12, !alias.scope !27, !noalias !29,
    ..., !llvm.access.group !30
    br label %pregion_for_end.i
```

```
if.then.i.us.us:
    %add.i.us.us = add i32 %mul.i, %conv2.i.us.us
    %idxprom.i.us.us = sext i32 %add.i.us.us to i64
    %arrayidx.i.us.us = getelementptr inbounds float, float* %0, i64
    ..., %idxprom.i.us.us
    %211 = load float, float* %arrayidx.i.us.us, align 4, !tbaa !12
    %add15.i.us.us = add i32 %add.i.us.us, -1
    %idxprom16.i.us.us = sext i32 %add15.i.us.us to i64
    %arrayidx17.i.us.us = getelementptr inbounds float, float* %0, i64
    ..., %idxprom16.i.us.us
    %212 = load float, float* %arrayidx17.i.us.us, align 4, !tbaa !12
    %add30.i.us.us = fadd float %add24.i.us.us, %214
    %add33.i.us.us = add nsw i32 %mul32.i, %conv2.i.us.us
    %add21.i.us.us = add i32 %add.i.us.us, 1
    %arrayidx22.i.us.us = sext i32 %add21.i.us.us to i64
    %arrayidx23.i.us.us = getelementptr inbounds float, float* %0, i64
    ..., %idxprom22.i.us.us
    %213 = load float, float* %arrayidx23.i.us.us, align 4, !tbaa !12
    %add24.i.us.us = fadd float %add18.i.us.us, %213
    %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* %0, i64
    ..., %idxprom28.i.us.us
    %214 = load float, float* %arrayidx29.i.us.us, align 4, !tbaa !12
    %add30.i.us.us = fadd float %add24.i.us.us, %214
    %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* %0, i64
    ..., %idxprom34.i.us.us
    %215 = load float, float* %arrayidx35.i.us.us, align 4, !tbaa !12
    %add36.i.us.us = fadd float %add30.i.us.us, %215
    %add39.i.us.us = fmul float %add36.i.us.us, 0x3FC99999A0000000
    %arrayidx41.i.us.us = getelementptr inbounds float, float* %1, i64
    ..., %idxprom.i.us.us
    store float %mul37.i.us.us, float* %arrayidx41.i.us.us, align 4, !tbaa !12,
    ..., !llvm.access.group !30
    br label %if.end.i.us.us

if.end.i.us.us:
    %216 = add nsw nsw i64 % local_id x0.o.us.us, 1
    %exitcond.not = icmp eqq i64 %216, 32
    br i1 %exitcond.not, label %pregion_for_end.i.loopexit, label
    ... %pregion_for_entry.entry.i.us.us, !llvm.loop !33
```

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

```
pregion_for_end.i:
    %217 = add nsw nsw i64 % local_id y0, 1
    %exitcond3.not = icmp eqq i64 %217, 8
    br i1 %exitcond3.not, label %runJacobi2D_kernel1.exit, label
    ... %pregion_for_entry.pregion_for_init.i.us.us, !llvm.loop !36
```

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

CFG for '_pool_kernel_runJacobi2D_kernel1' function