

```
%9:
%mul1.i = shl i64 %6, 5
%mul3.i.i = shl i64 %7, 3
%sub1.i = add nsw i32 %3, -1, !llvm.access.group !12
%sub4.i = add nsw i32 %4, -1, !llvm.access.group !12
%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
%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.splatinsert9 = insertelement <8 x i32> undef, i32 %sub4.i, i32 0
%broadcast.splat10 = shufflevector <8 x i32> %broadcast.splatinsert9, <8 x
... i32> undef, <8 x i32> zeroinitializer
%22 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%23 = or <8 x i32> %22, <i32 0, i32 1, i32 2, i32 3, i32 4, i32 5, i32 6,
... i32 7>
%24 = icmp sgt <8 x i32> %broadcast.splat10, %23
%25 = extractelement <8 x i32> %23, i32 0
%26 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%27 = or <8 x i32> %26, <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32
... 14, i32 15>
%28 = icmp sgt <8 x i32> %broadcast.splat10, %27
%29 = extractelement <8 x i32> %27, i32 0
%30 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%31 = or <8 x i32> %30, <i32 16, i32 17, i32 18, i32 19, i32 20, i32 21, i32
... 22, i32 23>
%32 = icmp sgt <8 x i32> %broadcast.splat10, %31
%33 = extractelement <8 x i32> %31, i32 0
%34 = trunc <8 x i64> %broadcast.splat to <8 x i32>
%35 = or <8 x i32> %34, <i32 24, i32 25, i32 26, i32 27, i32 28, i32 29, i32
... 30, i32 31>
%36 = icmp sgt <8 x i32> %broadcast.splat10, %35
%37 = extractelement <8 x i32> %35, 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, %9 ], [ %153, %region_for_end.i ]
%38 = mul i64 %local_id.y.0, %16
%add6.i.i = add nuw nsw i64 %local_id.y.0, %mul3.i.i, !llvm.access.group !12
%conv2.i = trunc i64 %add6.i.i to i32, !llvm.access.group !12
%cmp1.i = icmp sgt i32 %sub4.i, %conv2.i, !llvm.access.group !12
%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 %cmp1.i, label %vector.scevcheck, label %region_for_end.i
```

```
vector.scevcheck:
%39 = trunc i64 %38 to i32
%40 = add i32 %21, %39
%41 = trunc i64 %38 to i32
%42 = add i32 %17, %41
%43 = trunc i64 %38 to i32
%44 = add i32 %15, %43
%45 = icmp sgt i32 %44, 2147483616
%46 = icmp sgt i32 %44, 2147483616
%47 = or i1 %45, %46
%48 = icmp sgt i32 %40, 2147483616
%49 = or i1 %47, %48
br i1 %49, 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:
%50 = add i32 %mul.i, %25, !llvm.access.group !12
%51 = sext i32 %50 to i64, !llvm.access.group !12
%52 = getelementptr inbounds float, float* %2, i64 %51, !llvm.access.group
... !12
%53 = bitcast float* %52 to <8 x float>*
%wide.load = load <8 x float>, <8 x float>* %53, align 4, !tbaa !15,
... !llvm.access.group !12
%54 = fpxext <8 x float> %wide.load to <8 x double>, !llvm.access.group !12
%55 = add i32 %50, 1, !llvm.access.group !12
%56 = sext i32 %55 to i64, !llvm.access.group !12
%57 = getelementptr inbounds float, float* %0, i64 %56, !llvm.access.group
... !12
%58 = bitcast float* %57 to <8 x float>*
%wide.load11 = load <8 x float>, <8 x float>* %58, align 4, !tbaa !15,
... !llvm.access.group !12
%59 = getelementptr inbounds float, float* %0, i64 %51, !llvm.access.group
... !12
%60 = bitcast float* %59 to <8 x float>*
%wide.load12 = load <8 x float>, <8 x float>* %60, align 4, !tbaa !15,
... !llvm.access.group !12
%61 = fsub <8 x float> %wide.load11, %wide.load12, !llvm.access.group !12
%62 = add nsw i32 %mul19.i, %25, !llvm.access.group !12
%63 = sext i32 %62 to i64, !llvm.access.group !12
%64 = getelementptr inbounds float, float* %1, i64 %63, !llvm.access.group
... !12
%65 = bitcast float* %64 to <8 x float>*
%wide.load13 = load <8 x float>, <8 x float>* %65, align 4, !tbaa !15,
... !llvm.access.group !12
%66 = fadd <8 x float> %61, %wide.load13, !llvm.access.group !12
%67 = getelementptr inbounds float, float* %1, i64 %51, !llvm.access.group
... !12
%68 = bitcast float* %67 to <8 x float>*
%wide.load14 = load <8 x float>, <8 x float>* %68, align 4, !tbaa !15,
... !llvm.access.group !12
%69 = fsub <8 x float> %66, %wide.load14, !llvm.access.group !12
%70 = fpxext <8 x float> %69 to <8 x double>, !llvm.access.group !12
%71 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %70, <8 x double>
... <double 0xBF6E666666666666, double 0xBF6E666666666666, double
... double 0xBF6E666666666666, double 0xBF6E666666666666, double
... 0xBF6E666666666666>, <8 x double> %54), !llvm.access.group !12
%72 = fptrunc <8 x double> %71 to <8 x float>, !llvm.access.group !12
%73 = bitcast float* %52 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %72, <8 x float>*
... %73, i32 4, <8 x i1> %24), !tbaa !15, !llvm.access.group !12
%74 = add i32 %mul.i, %29, !llvm.access.group !12
%75 = sext i32 %74 to i64, !llvm.access.group !12
%76 = getelementptr inbounds float, float* %2, i64 %75, !llvm.access.group
... !12
%77 = bitcast float* %76 to <8 x float>*
%wide.load.1 = load <8 x float>, <8 x float>* %77, align 4, !tbaa !15,
... !llvm.access.group !12
%78 = fpxext <8 x float> %wide.load.1 to <8 x double>, !llvm.access.group !12
%79 = add i32 %74, 1, !llvm.access.group !12
%80 = sext i32 %79 to i64, !llvm.access.group !12
%81 = getelementptr inbounds float, float* %0, i64 %80, !llvm.access.group
... !12
%82 = bitcast float* %81 to <8 x float>*
%wide.load11.1 = load <8 x float>, <8 x float>* %82, align 4, !tbaa !15,
... !llvm.access.group !12
%83 = getelementptr inbounds float, float* %0, i64 %75, !llvm.access.group
... !12
%84 = bitcast float* %83 to <8 x float>*
%wide.load12.1 = load <8 x float>, <8 x float>* %84, align 4, !tbaa !15,
... !llvm.access.group !12
%85 = fsub <8 x float> %wide.load11.1, %wide.load12.1, !llvm.access.group !12
%86 = add nsw i32 %mul19.i, %29, !llvm.access.group !12
%87 = sext i32 %86 to i64, !llvm.access.group !12
%88 = getelementptr inbounds float, float* %1, i64 %87, !llvm.access.group
... !12
%89 = bitcast float* %88 to <8 x float>*
%wide.load13.1 = load <8 x float>, <8 x float>* %89, align 4, !tbaa !15,
... !llvm.access.group !12
%90 = fadd <8 x float> %85, %wide.load13.1, !llvm.access.group !12
%91 = getelementptr inbounds float, float* %1, i64 %75, !llvm.access.group
... !12
%92 = bitcast float* %91 to <8 x float>*
%wide.load14.1 = load <8 x float>, <8 x float>* %92, align 4, !tbaa !15,
... !llvm.access.group !12
%93 = fsub <8 x float> %90, %wide.load14.1, !llvm.access.group !12
%94 = fpxext <8 x float> %93 to <8 x double>, !llvm.access.group !12
%95 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %94, <8 x double>
... <double 0xBF6E666666666666, double 0xBF6E666666666666, double
... double 0xBF6E666666666666, double 0xBF6E666666666666, double
... 0xBF6E666666666666>, <8 x double> %54), !llvm.access.group !12
%96 = fptrunc <8 x double> %95 to <8 x float>, !llvm.access.group !12
%97 = bitcast float* %76 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %96, <8 x float>*
... %97, i32 4, <8 x i1> %28), !tbaa !15, !llvm.access.group !12
%98 = add i32 %mul.i, %33, !llvm.access.group !12
%99 = sext i32 %98 to i64, !llvm.access.group !12
%100 = getelementptr inbounds float, float* %2, i64 %99, !llvm.access.group
... !12
%101 = bitcast float* %100 to <8 x float>*
%wide.load.2 = load <8 x float>, <8 x float>* %101, align 4, !tbaa !15,
... !llvm.access.group !12
%102 = fpxext <8 x float> %wide.load.2 to <8 x double>, !llvm.access.group !12
%103 = add i32 %98, 1, !llvm.access.group !12
%104 = sext i32 %103 to i64, !llvm.access.group !12
%105 = getelementptr inbounds float, float* %0, i64 %104, !llvm.access.group
... !12
%106 = bitcast float* %105 to <8 x float>*
%wide.load11.2 = load <8 x float>, <8 x float>* %106, align 4, !tbaa !15,
... !llvm.access.group !12
%107 = getelementptr inbounds float, float* %0, i64 %99, !llvm.access.group
... !12
%108 = bitcast float* %107 to <8 x float>*
%wide.load12.2 = load <8 x float>, <8 x float>* %108, align 4, !tbaa !15,
... !llvm.access.group !12
%109 = fsub <8 x float> %wide.load11.2, %wide.load12.2, !llvm.access.group
... !12
%110 = add nsw i32 %mul19.i, %33, !llvm.access.group !12
%111 = sext i32 %110 to i64, !llvm.access.group !12
%112 = getelementptr inbounds float, float* %1, i64 %111, !llvm.access.group
... !12
%113 = bitcast float* %112 to <8 x float>*
%wide.load13.2 = load <8 x float>, <8 x float>* %113, align 4, !tbaa !15,
... !llvm.access.group !12
%114 = fadd <8 x float> %109, %wide.load13.2, !llvm.access.group !12
%115 = getelementptr inbounds float, float* %1, i64 %99, !llvm.access.group
... !12
%116 = bitcast float* %115 to <8 x float>*
%wide.load14.2 = load <8 x float>, <8 x float>* %116, align 4, !tbaa !15,
... !llvm.access.group !12
%117 = fsub <8 x float> %114, %wide.load14.2, !llvm.access.group !12
%118 = fpxext <8 x float> %117 to <8 x double>, !llvm.access.group !12
%119 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %118, <8 x double>
... <double 0xBF6E666666666666, double 0xBF6E666666666666, double
... double 0xBF6E666666666666, double 0xBF6E666666666666, double
... 0xBF6E666666666666>, <8 x double> %102), !llvm.access.group !12
%120 = fptrunc <8 x double> %119 to <8 x float>, !llvm.access.group !12
%121 = bitcast float* %100 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %120, <8 x float>*
... %121, i32 4, <8 x i1> %32), !tbaa !15, !llvm.access.group !12
%122 = add i7, !llvm.access.group !12
%123 = sext i32 %122 to i64, !llvm.access.group !12
%124 = getelementptr inbounds float, float* %2, i64 %123, !llvm.access.group
... !12
%125 = bitcast float* %124 to <8 x float>*
%wide.load.3 = load <8 x float>, <8 x float>* %125, align 4, !tbaa !15,
... !llvm.access.group !12
%126 = fpxext <8 x float> %wide.load.3 to <8 x double>, !llvm.access.group !12
%127 = add i32 %122, 1, !llvm.access.group !12
%128 = sext i32 %127 to i64, !llvm.access.group !12
%129 = getelementptr inbounds float, float* %0, i64 %128, !llvm.access.group
... !12
%130 = bitcast float* %129 to <8 x float>*
%wide.load11.3 = load <8 x float>, <8 x float>* %130, align 4, !tbaa !15,
... !llvm.access.group !12
%131 = getelementptr inbounds float, float* %0, i64 %123, !llvm.access.group
... !12
%132 = bitcast float* %131 to <8 x float>*
%wide.load12.3 = load <8 x float>, <8 x float>* %132, align 4, !tbaa !15,
... !llvm.access.group !12
%133 = fsub <8 x float> %wide.load11.3, %wide.load12.3, !llvm.access.group
... !12
%134 = add nsw i32 %mul19.i, %37, !llvm.access.group !12
%135 = sext i32 %134 to i64, !llvm.access.group !12
%136 = getelementptr inbounds float, float* %1, i64 %135, !llvm.access.group
... !12
%137 = bitcast float* %136 to <8 x float>*
%wide.load13.3 = load <8 x float>, <8 x float>* %137, align 4, !tbaa !15,
... !llvm.access.group !12
%138 = fadd <8 x float> %133, %wide.load13.3, !llvm.access.group !12
%139 = getelementptr inbounds float, float* %1, i64 %123, !llvm.access.group
... !12
%140 = bitcast float* %139 to <8 x float>*
%wide.load14.3 = load <8 x float>, <8 x float>* %140, align 4, !tbaa !15,
... !llvm.access.group !12
%141 = fsub <8 x float> %138, %wide.load14.3, !llvm.access.group !12
%142 = fpxext <8 x float> %141 to <8 x double>, !llvm.access.group !12
%143 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %142, <8 x double>
... <double 0xBF6E666666666666, double 0xBF6E666666666666, double
... double 0xBF6E666666666666, double 0xBF6E666666666666, double
... 0xBF6E666666666666>, <8 x double> %126), !llvm.access.group !12
%144 = fptrunc <8 x double> %143 to <8 x float>, !llvm.access.group !12
%145 = bitcast float* %124 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %144, <8 x float>*
... %145, i32 4, <8 x i1> %36), !tbaa !15, !llvm.access.group !12
br label %region_for_end.i
```

```
if.then.i.us:
%add.i.us = add i32 %mul.i, %conv1.us, !llvm.access.group !12
%idxprom1.us = sext i32 %add.i.us to i64, !llvm.access.group !12
%arrayidx1.us = getelementptr inbounds float, float* %2, i64 %idxprom1.us,
... !llvm.access.group !12
%146 = load float, float* %arrayidx1.us, align 4, !tbaa !15,
... !llvm.access.group !12
%conv71.us = fpxext float %146 to double, !llvm.access.group !12
%add101.us = add i32 %add.i.us, 1, !llvm.access.group !12
%idxprom11.us = sext i32 %add101.us to i64, !llvm.access.group !12
%arrayidx12.us = getelementptr inbounds float, float* %0, i64
... %idxprom11.us, !llvm.access.group !12
%147 = load float, float* %arrayidx12.us, align 4, !tbaa !15,
... !llvm.access.group !12
%arrayidx16.us = getelementptr inbounds float, float* %0, i64
... %idxprom11.us, !llvm.access.group !12
%148 = load float, float* %arrayidx16.us, align 4, !tbaa !15,
... !llvm.access.group !12
%sub17.us = fsub float %147, %148, !llvm.access.group !12
%add201.us = add nsw i32 %mul19.i, %conv1.us, !llvm.access.group !12
%idxprom21.us = sext i32 %add201.us to i64, !llvm.access.group !12
%arrayidx22.us = getelementptr inbounds float, float* %1, i64
... %idxprom21.us, !llvm.access.group !12
%149 = load float, float* %arrayidx22.us, align 4, !tbaa !15,
... !llvm.access.group !12
%add23.us = fadd float %sub17.us, %149, !llvm.access.group !12
%arrayidx27.us = getelementptr inbounds float, float* %1, i64
... %idxprom11.us, !llvm.access.group !12
%150 = load float, float* %arrayidx27.us, align 4, !tbaa !15,
... !llvm.access.group !12
%sub28.us = fsub float %add23.us, %150, !llvm.access.group !12
%conv29.us = fpxext float %sub28.us to double, !llvm.access.group !12
%151 = tail call double @llvm.fmuladd.f64(double %conv29.us, double
... 0xBF6E666666666666, double %conv71.us) #4, !llvm.access.group !12
store float %conv31.us, float* %arrayidx1.us, align 4, !tbaa !15,
... !llvm.access.group !12
br label %if.end.r_exit.i.us, !llvm.access.group !12
```

```
if.end.r_exit.i.us:
%152 = add nuw nsw i64 %local_id.x.0.us, 1
%exitcond.not = icmp eq i64 %152, 32
br i1 %exitcond.not, label %region_for_end.i.loopexit, label
... %region_for_entry.entry.i.us, !llvm.loop !19
```

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

```
region_for_end.i:
%153 = add nuw nsw i64 %local_id.y.0, 1
%exitcond2.not = icmp eq i64 %153, 8
br i1 %exitcond2.not, label %fddt_kernel3_exit, label
... %region_for_entry.region_for_init.i, !llvm.loop !22
```

```
fddt_kernel3_exit:
ret void
```