

```
%9:
%mul.i.i = shl i64 %6, 5
%mul3.i.i = shl i64 %7, 3
%10 = trunc i64 %6 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 = add i32 %15, -8
%18 = or i32 %17, 7
%19 = trunc i64 %6 to i32
%20 = mul i32 %19, %4
%21 = shl i32 %20, 3
%22 = trunc i64 %6 to i32
%23 = shl i32 %22, 5
%24 = add i32 %21, %23
%25 = zext i32 %4 to i64
%scevgep7 = getelementptr float, float* %0, i64 32
%26 = add i32 %24, -8
%27 = or i32 %26, 7
%scevgep12 = getelementptr float, float* %2, i64 32
%scevgep17 = getelementptr float, float* %2, i64 32
br label %region for_entry.region for_init.i
```

```

preparation for entry.pregion for init.i:
% local id y_0 = phi i64 [%0, %9 ], [%71, %preregion_for_end.i ]
%28 = mul i64 % local id y_0, %25
%29 = trunc i64 %28 to i32
%30 = add i32 %24, %29
%31 = sext i32 %30 to i64
%scevgep = getelementptr float, float* %0, i64 %31
%scevgep8 = getelementptr float, float* %scevgep7, i64 %31
%32 = trunc i64 %28 to i32
%33 = add i32 %27, %32
%34 = sext i32 %33 to i64
%scevgep10 = getelementptr float, float* %2, i64 %34
%scevgep13 = getelementptr float, float* %scevgep12, i64 %34
%35 = mul i64 % local id y_0, %16
%add6.i.1 = add nuw nsw i64 % local id y_0, %mul3.i.i
%conv2.i.1 = trunc i64 %add6.i.1 to i32
%cmp.i.1 = icmp slt i32 %conv2.i.1, %3
%mul.i.1 = mul nsw i32 %conv2.i.1, %4
br i1 %cmp.i.1, label %vector.scevcheck, label %preregion_for_end.i

```

```
vector.scevcheck:
%36 = trunc i64 %35 to i32
%37 = add i32 %18, %36
%38 = trunc i64 %35 to i32
%39 = add i32 %15, %38
%40 = icmp sgt i32 %39, 2147483616
%41 = icmp sgt i32 %37, 2147483616
%42 = or i1 %40, %41
br i1 %42, label %pregion_for_entry.entry.i.us.preheader, label
... %vector.memcheck
```

```
vector.memcheck
%bound0 = icmp ult float* %scevgep, %scevgep13
%bound1 = icmp ult float* %scevgep10, %scevgep8
%found.conflict = and i1 %bound0, %bound1
%bound020 = icmp ugt float* %scevgep17, %0
%bound121 = icmp ugt float* %scevgep7, %2
%found.conflict22 = and i1 %bound020, %bound121
%conflict.rdx = or i1 %found.conflict, %found.conflict22
br i1 %conflict.rdx, label %region_for_entry.entry.i.us.preheader, label
... %vector.ph
```

```
pregion for entry.entry.i.us.preheader:  
br label %pregion for entry.entry.i.us
```

```
vector.ph:
%broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
%broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
..., undef, <8 x i32> zeroinitializer
%broadcast.splatinsert23 = insertelement <8 x i32> undef, i32 %4, i32 0
%broadcast.splat24 = shufflevector <8 x i32> %broadcast.splatinsert23, <8 x
..., i32> undef, <8 x i32> zeroinitializer
br label %vector.body
```

```

vector.body:
%index = phi i64 [ 0, %vector.ph ], [ %index.next, %vector.body ]
%vec.ind = phi <8 x i64> [ <i64 0, i64 1, i64 2, i64 3, i64 4, i64 5, i64 6,
... i64 7>, %vector.ph ], [ %vec.ind.next, %vector.body ]
%43 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
%44 = trunc <8 x i64> %43 to <8 x i32>
%45 = icmp sgt <8 x i32> %0, %broadcast.splat24, %44
%46 = icmp sgt <8 x i32> %44, zeroinitializer
%47 = and <8 x i1> %45, %46
%48 = extractelement <8 x i32> %44, i32 0
%49 = add i32 %mul.i, %48
%50 = sext i32 %49 to i64
%51 = getelementptr inbounds float, float* %0, i64 %50
%52 = bitcast float* %51 to <8 x float>*
%wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %52, i32 4, <8 x i1> %47, <8 x float> undef), !tbaa !12, !alias.scope
... !16, !noalias !19
%53 = fpext <8 x float> %wide.masked.load to <8 x double>
%54 = getelementptr inbounds float, float* %2, i64 %50
%55 = bitcast float* %54 to <8 x float>*
%wide.masked.load25 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %55, i32 4, <8 x i1> %47, <8 x float> undef), !tbaa !12, !alias.scope
... !22
%56 = add i32 %49, -1
%57 = sext i32 %56 to i64
%58 = getelementptr inbounds float, float* %2, i64 %57
%59 = bitcast float* %58 to <8 x float>*
%wide.masked.load26 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %59, i32 4, <8 x i1> %47, <8 x float> undef), !tbaa !12, !alias.scope
... !23
%60 = fsub <8 x float> %wide.masked.load25, %wide.masked.load26
%61 = fpext <8 x float> %60 to <8 x double>
%62 = call <8 x double> @llvm.fmuladd.v8f64(<8 x double> %61, <8 x double>
... <double -5.0000000e-01, double -5.000000e-01, double -5.000000e-01, double
... -5.000000e-01, double -5.000000e-01, double -5.000000e-01, double
... -5.000000e-01, double -5.000000e-01>, <8 x double> %53)
%63 = fptrunc <8 x double> %62 to <8 x float>
%64 = bitcast float* %51 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %63, <8 x float>*
... %64, i32 4, <8 x i1> %47), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !24
%index.next = add i64 %index, 8
%vec.ind.next = add <8 x i64> %vec.ind, <i64 8, i64 8, i64 8, i64 8, i64 8,
... i64 8, i64 8, i64 8>
%65 = icmp eq i64 %index.next, 32
br i1 %65, label %pregion_for_end.i.loopexit28, label %vector.body,
... !llvm.loop !27

```

```

prejoin for entry.entry.i.us:
% local id_x_0.us = phi i64 [%70, %if.end.i.us], [0,
... %prejoin for entry.entry.i.us.preheader]
%addl.i.i.us = add nsw nsw i64 %local id_x_0.us, %mul.i.i
%conv.i.us = trunc i64 %addl.i.i.us to i32
%cmp4.i.us = icmp slt i32 %conv.i.us, %4
%cmp7.i.us = icmp sgt i32 %conv.i.us, 0
%or.cond.i.us = and i1 %cmp4.i.us, %cmp7.i.us
br i1 %or.cond.i.us, label %if.then.i.us, label %if.end.i.us

```

```
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* %0, i64 %idxprom.i.us
  %66 = load float, float* %arrayidx.i.us, align 4, !tbaa !12
  %conv9.i.us = fpxext float %66 to double
  %arrayidx13.i.us = getelementptr inbounds float, float* %2, i64 %idxprom.i.us
  %67 = load float, float* %arrayidx13.i.us, align 4, !tbaa !12
  %add15.i.us = add i32 %add.i.us, -1
  %idxprom16.i.us = sext i32 %add15.i.us to i64
  %arrayidx17.i.us = getelementptr inbounds float, float* %2, i64
  ... %idxprom16.i.us
  %68 = load float, float* %arrayidx17.i.us, align 4, !tbaa !12
  %sub18.i.us = fsub float %67, %68
  %conv19.i.us = fpxext float %sub18.i.us to double
  %69 = tail call @llvm.fmuladd.f64(double %conv19.i.us, double
  ... -5.000000e-01, double %conv9.i.us) #5
  %conv21.i.us = fptrunc double %69 to float
  store float %conv21.i.us, float* %arrayidx.i.us, align 4, !tbaa !12,
  ... !llvm.access.group !24
  br label %if.end.i.us
```

```
if.end.i.us:
%70 = add nuw nsw i64 % local id x.0.us, 1
%exitcond.not = icmp eq i64 %70, 32
br i1 %exitcond.not, label %region_for_end.i.loopexit, label
... %region_for_entry.entry.i.us, !llvm.loop !30
```

```
preregion_for_end.i.loopexit28:  
br label %preregion_for_end.i
```

```

pregion_for_end.i:
%71 = add nuw nsw i64 % local_id_y.0, 1
%exitcond2.not = icmp eq i64 %71, 8
br i1 %exitcond2.not, label %fddt_kernel2.exit, label
... %pregion_for_entry.pregion_for_init.i, !llvm.loop !31

```

```
fdtd_kernel2.exit:  
    ret void
```