

```
%9:
%mul.i.i = shl i64 %6, 5
%mul3.i.i = shl i64 %7, 3
%sub.i.i = add nsw i32 %3, -1
%sub4.i.i = add nsw i32 %4, -1
%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
%22 = trunc i64 %7 to i32
%23 = mul i32 %22, %4
%24 = shl i32 %23, 3
%25 = trunc i64 %6 to i32
%26 = shl i32 %25, 5
%27 = add i32 %24, %26
%28 = zext i32 %4 to i64
%scvgevp10 = getelementptr float, float*, i64 32
%scvgevp15 = getelementptr float, float*, i64 32
%29 = or i32 %27, 1
%30 = zext i32 %29 to i64
%scvgevp20 = getelementptr float, float*, i64 32
%scvgevp25 = getelementptr float, float*, i64 32
%31 = shl i32 %22, 3
%32 = or i32 %31, 1
%33 = mul i32 %32, %4
%34 = add i32 %33, %26
%scvgevp30 = getelementptr float, float*, i64 32
br label %preloop for entry preloop for init.i
```

```

program for entry.pregion for init.i:
% local id y0 = phi i64 [0, %9 ], [%92, %region_for_end.i ]
%35 = mul i64 % local id y0, %0, %28
%36 = trunc i64 %35 to i32
%37 = add i32 %27, %36
%38 = sext i32 %37 to i64
%sevcgep = getelementptr float, float* %2, i64 %38
%sevcgep1 = getelementptr float, float* %sevcgep10, i64 %38
%39 = add i64 %35, %30
%sext = shl i64 %39, 32
%40 = ashr exact i64 %sext, 32
%sevcgep18 = getelementptr float, float* %0, i64 %40
%sevcgep21 = getelementptr float, float* %sevcgep20, i64 %40
%41 = trunc i64 %35 to i32
%42 = add i32 %34, %41
%43 = sext i32 %42 to i64
%sevcgep28 = getelementptr float, float* %1, i64 %43
%sevcgep31 = getelementptr float, float* %sevcgep30, i64 %43
%44 = mul i64 % local id y0, %16
%add6.i.1 = add nsw i64 % local id y0, %mul3.i.i
%conv2.i = trunc i64 %add6.i.1 to i32
%cmp.i = icmp sgt i32 %sub.i, %conv2.i
%mul.i = mul nsw i32 %conv2.i, %4
%add18.i = add nsw i32 %conv2.i, 1
%mul9.i = mul nsw i32 %add18.i, %4
br i1 %cmp.i, label %vector.scevccheck, label %region_for_end.i

```

```
vector.sceycheck:
%45 = trunc i64 %44 to i32
%46 = add i32 %21, %45
%47 = trunc i64 %44 to i32
%48 = add i32 %17, %47
%49 = trunc i64 %44 to i32
%50 = add i32 %15, %49
%51 = icmp sgt i32 %50, 2147483616
%52 = icmp sgt i32 %48, 2147483616
%53 = or i1 %51, %52
%54 = icmp sgt i32 %46, 2147483616
%55 = or i1 %53, %54
br i1 %55, label %region_for_entry.entry.i.us.preheader, label
%vector.membercheck
```

```
vector.memcheck:
%bound0 = icmp ugt float* %scevgep15, %2
%bound1 = icmp ugt float* %scevgep10, %0
%found.conflict = and i1 %bound0, %bound1
%bound033 = icmp ugt float* %scevgep, %scevgep21
%bound134 = icmp ugt float* %scevgep18, %scevgep11
%found.conflict35 = and i1 %bound033, %bound134
%conflict.rdx = or i1 %found.conflict, %found.conflict35
%bound036 = icmp ugt float* %scevgep25, %2
%bound137 = icmp ugt float* %scevgep10, %1
%found.conflict38 = and i1 %bound036, %bound137
%conflict.rdx39 = or i1 %conflict.rdx, %found.conflict38
%bound040 = icmp ugt float* %scevgep, %scevgep31
%bound141 = icmp ugt float* %scevgep28, %scevgep11
%found.conflict42 = and i1 %bound040, %bound141
%conflict.rdx43 = or i1 %conflict.rdx39, %found.conflict42
br i1 %conflict.rdx43, label %prejoin_for_entry.entry.us.preheader, label
%vector.nh
```

```
vector.ph:
%broadcast.splatinert = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
%broadcast.splat = shufflevector <8 x i64> %broadcast.splatinert, <8 x i64>
... undef, <8 x i32> %zeroinitializer
%broadcast.splatinert44 = insertelement <8 x i32> undef, i32 %sub4.i, i32 0
%broadcast.splat45 = shufflevector <8 x i32> %broadcast.splatinert44, <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 ]
%56 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
%57 = trunc <8 x i64> %56 to <8 x i32>
%58 = icmp sgt <8 x i32> %broadcast.splat45, %57
%59 = extractelement <8 x i32> %57, i32 0
%60 = add i32 %mul1, %59
%61 = sext i32 %60 to i64
%62 = getelementptr inbounds float, float* %2, i64 %61
%63 = bitcast float* %62 to <8 x float>*
%wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... float>* %63, i32 4, <8 x i1> %58, <8 x float> undef!, !tbaa !12, !alias.scope
... !116, !noalias !19
%64 = fpext <8 x float> %wide.masked.load to <8 x double>
%65 = add i32 %60, 1
%66 = sext i32 %65 to i64
%67 = getelementptr inbounds float, float* %0, i64 %66
%68 = bitcast float* %67 to <8 x float>*
%wide.masked.load46 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... float>* %68, i32 4, <8 x i1> %58, <8 x float> undef!, !tbaa !12, !alias.scope
... !124
%69 = getelementptr inbounds float, float* %0, i64 %61
%70 = bitcast float* %69 to <8 x float>*
%wide.masked.load47 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... float>* %70, i32 4, <8 x i1> %58, <8 x float> undef!, !tbaa !12, !alias.scope
... !125
%71 = fsub <8 x float> %wide.masked.load46, %wide.masked.load47
%72 = add nsw i32 %mul19.i, %59
%73 = sext i32 %72 to i64
%74 = getelementptr inbounds float, float* %1, i64 %73
%75 = bitcast float* %74 to <8 x float>*
%wide.masked.load48 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... float>* %75, i32 4, <8 x i1> %58, <8 x float> undef!, !tbaa !12, !alias.scope
... !126
%76 = fadd <8 x float> %71, %wide.masked.load48
%77 = getelementptr inbounds float, float* %1, i64 %61
%78 = bitcast float* %77 to <8 x float>*
%wide.masked.load49 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32<8 x
... float>* %78, i32 4, <8 x i1> %58, <8 x float> undef!, !tbaa !12, !alias.scope
... !127
%79 = fsub <8 x float> %76, %wide.masked.load49
%80 = fpext <8 x float> %79 to <8 x double>
%81 = call <8 x double> @llvm.fmuladd.v8f64<8 x double> %80, <8 x double>
... <@BFE666666666666666, double> @BFE6666666666666, double
... <@BFE666666666666666, double> @BFE66666666666666, double
... <@BFE6666666666666666, double> @BFE66666666666666, double
... <@BFE6666666666666666>, <8 x double> %64)
%82 = fptrunc <8 x double> %81 to <8 x float>
%83 = bitcast float* %62 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32<8 x float> %82, <8 x float>*
... %63, i32 4, <8 x i1> %58!, !tbaa !12, !alias.scope !116, !noalias !19,
... !llv.access_group !28
%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>
%84 = icmp eq i64 %index.next, 32
br i1 %84, label %region_for_end.i.loopexit51, label %vector.body,
... !llvm.loop !31

```

```

preparation for entry.y.i.us:
% local id x_0.us = phi i64 [%91,%if.end.r_exit.i.us ], [ 0,
... %entry.y.i.us for entry.y.i.us.preheader ]
%add1.i.us = add nuw nsw i64 % local id x_0.us,%mul.i
%conv.i.us = trunc i64 %add1.i.us to i32
%cmp5.i.us = icmp sgt i32 %sub4.i,%conv.i.us
br i1 %cmp5.i.us, label %if.then.i.us, label %if.end.r_exit.i.us

```

```
if then i.us:
%addi.i.us = add i32 %mul.i, %conv.i.us
%idxpromd.i.us = sext i32 %add.i.us to i64
%arrayidx1.i.us = getelementptr inbounds float, float* %2, i64 %idxprom.i.us
%85 = load float, float* %arrayidx1.i.us, align 4, !tbaa !12
%conv7.i.us = fpxext float %85 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
%86 = load float, float* %arrayidx12.i.us, align 4, !tbaa !12
%arrayidx16.i.us = getelementptr inbounds float, float* %0, i64 %idxprom.i.us
%87 = load float, float* %arrayidx16.i.us, align 4, !tbaa !12
%sub17.i.us = fsub float %86, %87
%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
%88 = load float, float* %arrayidx22.i.us, align 4, !tbaa !12
%add23.i.us = fadd float %sub17.i.us, %88
%arrayidx27.i.us = getelementptr inbounds float, float* %1, i64 %idxprom.i.us
%89 = load float, float* %arrayidx27.i.us, align 4, !tbaa !12
%sub28.i.us = fsub float %add23.i.us, %89
%conv29.i.us = fpxext float %sub28.i.us to double
%90 = tail call double @llvm.fmuladd.f64(double %conv29.i.us, double
... 0xBF66666666666666, float %conv7.i.us) #5
%conv31.i.us = fptrunc double %90 to float
store float %conv31.i.us, float* %arrayidx.i.us, align 4, !tbaa !12,
... !lvm.access.group 128
br label %if.end.r exit.i.us
```

if.end.r.exit.i.us: %91 = add nuw nsw i64 % local_id_x.0.us, 1 %exitcond.not = icmp eq i64 %91, 32 br i1 %exitcond.not, label %region_for_end.i.loopexit, label ... %region_for_entry.entry.i.us, !llvm.loop !34	
T	F

```
preregion_for_end.i.loopexit51  
br label %preregion_for_end.i
```

```

pregion for end.i:
%92 = add_nuw nsw i64 %local_id_y0, 1
%exitcond2.not = icmp eq i64 %92, 8
br i1 %exitcond2.not, label %fddt_kernel3.exit, label
... %pregion for entry, pregion for init.i, !llvm.loop !35

```

```
fdtd_kernel3.exit:
    ret void
```