```
%10 = \text{shl } i64 \%6, 5
                                                                                                                                                                                         %11 = shl i64 \%7.3
                                                                                                                                                                                         %12 = trunc i64 %7 to i32
                                                                                                                                                                                         %13 = mul i32 %12, %4
                                                                                                                                                                                         %14 = shl i32 %13, 3
                                                                                                                                                                                         %15 = trunc i64 %6 to i32
                                                                                                                                                                                         %16 = \text{shl i} 32 \%15, 5
                                                                                                                                                                                         %17 = add i32 %14, %16
                                                                                                                                                                                         %18 = \text{zext i} 32 \%4 \text{ to i} 64
                                                                                                                                                                                         %19 = add i32 %17, -8
                                                                                                                                                                                         %20 = \text{ or } i32 \%19, 7
                                                                                                                                                                                         %21 = trunc i64 %7 to i32
                                                                                                                                                                                         %22 = mul i32 %21, %4
                                                                                                                                                                                         %23 = \text{shl i} 32 \%22, 3
                                                                                                                                                                                         %24 = trunc i64 %6 to i32
                                                                                                                                                                                         %25 = \text{shl i} 32 \%24, 5
                                                                                                                                                                                         %26 = add i32 %23, %25
                                                                                                                                                                                         %27 = \text{zext i} 32 \%4 \text{ to i} 64
                                                                                                                                                                                         %scevgep7 = getelementptr float, float* %0, i64 32
                                                                                                                                                                                         %28 = add i32 %26, -8
                                                                                                                                                                                         %29 = \text{ or } i32 \%28, 7
                                                                                                                                                                                         %scevgep12 = getelementptr float, float* %2, i64 32
                                                                                                                                                                                         %scevgep17 = getelementptr float, float* %2, i64 32
                                                                                                                                                                                         br label %pregion for entry.pregion for init.i
                                                                                                                                                                                 pregion for entry.pregion for init.i:
                                                                                                                                                                                  % local id y.0 = phi i64 [0, \%9], [\%75, \%pregion for end.i ]
                                                                                                                                                                                  \%30 = \text{mul } i64 \% \text{ local id y.0, } \%27
                                                                                                                                                                                  %31 = \text{trunc } i64 \ \%30 \ \text{to} \ i32
                                                                                                                                                                                  %32 = add i32 %26, %31
                                                                                                                                                                                  %33 = \text{sext i} 32 \% 32 \text{ to i} 64
                                                                                                                                                                                  %scevgep = getelementptr float, float* %0, i64 %33
                                                                                                                                                                                  %scevgep8 = getelementptr float, float* %scevgep7, i64 %33
                                                                                                                                                                                  %34 = trunc i64 %30 to i32
                                                                                                                                                                                  %35 = add i32 %29, %34
                                                                                                                                                                                  %36 = \text{sext i} 32 \% 35 \text{ to i} 64
                                                                                                                                                                                  %scevgep10 = getelementptr float, float* %2, i64 %36
                                                                                                                                                                                  %scevgep13 = getelementptr float, float* %scevgep12, i64 %36 %37 = mul i64 %_local_id_y.0, %18
                                                                                                                                                                                  \%38 = \text{add nuw nsw } i6\overline{4} \% \text{ local id y.0, } \%11
                                                                                                                                                                                  \%conv2.i = trunc i64 \%38 to i32
                                                                                                                                                                                  %cmp.i = icmp slt i32 %conv2.i, %3
                                                                                                                                                                                  %mul.i = mul nsw i32 %conv2.i, %4
                                                                                                                                                                                  br i1 %cmp.i, label %vector.scevcheck, label %pregion_for_end.i
                                                                                                                                                                                                    Τ
                                                                                       vector.scevcheck:
                                                                                       %39 = trunc i64 %37 to i32
                                                                                        %40 = add i32 %20, %39
                                                                                        %41 = trunc i64 %37 to i32
                                                                                        %42 = add i32 %17, %41
                                                                                       %43 = icmp sgt i32 %42, 2147483616
                                                                                        %44 = icmp \ sgt \ i32 \ %40, \ 2147483616
                                                                                       %45 = \text{ or i } 1 \% 43, \% 44
                                                                                       br i1 %45, 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 %pregion for entry.entry.i.us.preheader, label
                                                                                                      ... %vector.ph
                                                                                                                           vector.ph:
                                                                                                                           %broadcast.splatinsert = insertelement <8 x i64> undef, i64 %10, i32 0
                                                                                                                           %broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
                                                                                                                           .. undef, < 8 \times i32 > zeroinitializer
                                                                    pregion for entry.entry.i.us.preheader:
                                                                                                                           %broadcast.splatinsert23 = insertelement <8 x i32> undef, i32 %4, i32 0
                                                                     br label %pregion for entry.entry.i.us
                                                                                                                           %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 ] %46 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
                                                                                                                         %47 = trunc <8 x i64> %46 to <8 x i32>
                                                                                                                         %48 = icmp sgt <8 x i32> %broadcast.splat24, %47
%49 = icmp sgt <8 x i32> %47, zeroinitializer
%50 = and <8 x i1> %48, %49
                                                                                                                         %51 = \text{extractelement} < 8 \times i32 > %47, i32 0
                                                                                                                         %52 = add i32 %mul.i, %51
                                                                                                                         %53 = \text{sext i} 32 \% 52 \text{ to i} 64
                                                                                                                         %54 = getelementptr inbounds float, float* %0, i64 %53
                                                                                                                         %55 = bitcast float* %54 to <8 x float>*
                                                                                                                         %wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
                                                                                                                         ... float>* %55, i32 4, <8 x i1> %50, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                         ... !16, !noalias !19
                                                                                                                         %56 = fpext <8 x float> %wide.masked.load to <8 x double> %57 = getelementptr inbounds float, float* %2, i64 %53
                                                                                                                         %58 = bitcast float* %57 to <8 x float>*
                                            pregion for entry.entry.i.us:
                                             %_local_id_x.0.us = phi i64 [ %74, %if.end.i.us ], [ 0, ... %pregion_for_entry.entry.i.us.preheader ] %69 = add nuw nsw i64 %_local_id_x.0.us, %10
                                                                                                                         %wide.masked.load25 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x ... float>* %58, i32 4, <8 x i1> %50, <8 x float> undef), !tbaa !12, !alias.scope
                                                                                                                         %59 = add i32 %52, -1
                                             %conv.i.us = trunc i64 \%69 to i3\overline{2}
                                                                                                                         \%60 = \text{sext i} 32 \% 59 \text{ to i} 64
                                             %cmp4.i.us = icmp slt i32 %conv.i.us, %4
                                                                                                                         %61 = getelementptr inbounds float, float* %2, i64 %60
                                             %cmp7.i.us = icmp sgt i32 %conv.i.us, 0
                                             %or.cond.i.us = and i1 %cmp4.i.us, %cmp7.i.us
                                                                                                                          \%62 = bitcast float* \%61 to <8 x float>*
                                                                                                                         %wide.masked.load26 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x ... float>* %62, i32 4, <8 x i1> %50, <8 x float> undef), !tbaa !12, !alias.scope
                                             br i1 %or.cond.i.us, label %if.then.i.us, label %if.end.i.us
                                                                                                                         %63 = fsub <8 x float> %wide.masked.load25, %wide.masked.load26
                                                                                                                         \%64 = \text{fpext} < 8 \times \text{float} > \%63 \text{ to } < 8 \times \text{double} >
                                                                                                                         \%65 = \text{call} < 8 \text{ x double} > \text{@llvm.fmuladd.v8f64} (< 8 \text{ x double} > \%64, < 8 \text{ x double} >
                                                                                                                         ... <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, double -5.000000e-01>, <8 x double> %56)
                                                                                                                         \%66 = \text{fptrunc} < 8 \times \text{double} > \%65 \text{ to} < 8 \times \text{float} >
                                                                                                                         %67 = bitcast float* %54 to <8 x float>*
                                                                                                                         call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %66, <8 x float>*
                                                                                                                         ... %67, i32 4, <8 x i1> %50), !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>
                                                                                                                         %68 = icmp eq i64 %index.next, 32
                                                                                                                         br i1 %68, label %pregion for end.i.loopexit28, label %vector.body,
                                                                                                                         ...!llvm.loop!27
                                                                                                                                                                                                     F
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 %70 = load float, float* %arrayidx.i.us, align 4, !tbaa !12 %conv9.i.us = fpext float %70 to double
%arrayidx13.i.us = getelementptr inbounds float, float* %2, i64 %idxprom.i.us %71 = 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
%72 = load float, float* %arrayidx17.i.us, align 4, !tbaa !12 %sub18.i.us = fsub float %71, %72
%conv19.i.us = fpext float %sub18.i.us to double
%73 = tail call double @llvm.fmuladd.f64(double %conv19.i.us, double ... -5.000000e-01, double %conv9.i.us) #5
%conv21.i.us = fptrunc double %73 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:
                                                               \%74 = \text{add nuw nsw } i64 \% \text{ local id } x.0.\text{us, } 1
                                                               %exitcond = icmp eq i64 %74, 32
br i1 %exitcond, label %pregion_for_end.i.loopexit, label
                                                                                                                                                                    pregion for end.i.loopexit28:
                                                                                                                                                                    br label %pregion for end.i
                                                               ... %pregion for entry.entry.i.us, !llvm.loop!30
                                                                                                              pregion_for_end.i.loopexit:
  br label %pregion_for_end.i
                                                                                                                                                                             pregion for end.i:
                                                                                                                                                                             ^{1}\%75 = add nuw nsw i64 % local id y.0, 1
                                                                                                                                                                             \%exitcond2 = icmp eq i64 \%75, 8
                                                                                                                                                                             br i1 %exitcond2, label %fdtd kernel2.exit, label
                                                                                                                                                                            ... %pregion for entry.pregion for init.i, !llvm.loop !31
                                                                                                                                                                                 fdtd kernel2.exit:
                                                                                                                                                                                  ret void
```