%mul.i.i = shl i64 %7, 5 %mul3.i.i = shl i64 %8, 3 %sub.i = add nsw i32 %2, -1 %cmp.i = icmp sgt i32 %sub.i, %5 %sub4.i = add nsw i32 %3, -1%sub8.i = add nsw i32 %4, -1 %cmp12.i = icmp sgt i32 %5, 0 %mul189.i = mul i32 %5, %3 %sub20.i = add nsw i32 %5, -1 %mul.i = mul nsw i32 %4, %3 %mul21.i = mul nsw i32 %sub20.i, %mul.i %add27.i = add nuw nsw i32 %5. 1%mul29.i = mul nsw i32 %add27.i, %mul.i %mul84.i = mul nsw i32 %mul.i, %5 %11 = trunc i64 %8 to i32 %12 = shl i32 %11, 3%13 = add i32 %mul189.i, %12 %14 = mul i32 %13, %4 %15 = trunc i64 %7 to i32 %16 = shl i32 %15, 5 %17 = add i32 %14, %16 br label %pregion for entry.pregion for init.i pregion\_for\_entry.pregion\_for\_init.i: %\_local\_id\_y.0 = phi i64 [ 0, %10 ], [ %56, %pregion\_for\_end.i ] %add6.i.i = add nuw nsw i64 %\_local\_id\_y.0, %mul3.i.i %conv2.i = trunc i64 %add6.i.i to i32%cmp5.i = icmp sgt i32 %sub4.i, %conv2.i %or.cond338.i = and i1 %cmp.i, %cmp5.i %cmp15.i = icmp sgt i32 %conv2.i, 0 %reass.add.i = add i32 %mul189.i, %conv2.i %reass.mul.i = mul i32 %reass.add.i. %4 %sub22.i = add nsw i32 %conv2.i, -1 %mul23.i = mul nsw i32 %sub22.i, %4 %add.i = add nsw i32 %mul23.i, %mul21.i %add32.i = add nsw i32 %mul23.i, %mul29.i %mul97.i = mul nsw i32 %conv2.i, %4 %add107.i = add nuw nsw i32 %conv2.i, 1 %mul108.i = mul nsw i32 %add107.i, %4 br i1 %or.cond338.i, label %pregion for entry.entry.i.us.preheader, label ... %vector.scevcheck pregion for entry.entry.i.us.preheader: br label %pregion for entry.entry.i.us pregion for entry.entry.i.us: % local id x.0.us = phi i64 [ %54, %if.end.r exit.i.us ], [ 0, ... %pregion for entry.entry.i.us.preheader ] vector.scevcheck: %add1.i.i.us = add nuw nsw i64 % local id x.0.us, %mul.i.i %18 = trunc i64 %\_local\_id\_y.0 to i32 %conv.i.us = trunc i64 %add1.i.i.us to i32 %19 = mul i 32 %18, %4%cmp9.i.us = icmp sgt i32 %sub8.i, %conv.i.us %or.cond.i.us = and i1 %cmp12.i, %cmp9.i.us %20 = add i32 %19, %17 %21 = icmp sgt i32 %20, 2147483616 br i1 %21, label %pregion\_for\_entry.entry.i.preheader, label %or.cond197.i.us = and i1 %cmp15.i, %or.cond.i.us %cmp18.i.us = icmp sgt i32 %conv.i.us, 0 ... %vector.body.preheader %or.cond199.i.us = and i1 %cmp18.i.us, %or.cond197.i.us br i1 %or.cond199.i.us, label %if.then.i.us, label %if.else.i.us if.then.i.us: %sub24.i.us = add nsw i32 %conv.i.us, -1 %add25.i.us = add nsw i32 %sub24.i.us, %add.i %idxprom.i.us = sext i32 %add25.i.us to i64 %arrayidx.i.us = getelementptr inbounds float, float\* %0, i64 %idxprom.i.us %29 = load float, float\* %arrayidx.i.us, align 4, !tbaa !12 %add34.i.us = add nsw i32 %sub24.i.us, %add32.i %idxprom35.i.us = sext i32 %add34.i.us to i64 %arrayidx36.i.us = getelementptr inbounds float, float\* %0, i64 ... %idxprom35.i.us %30 = load float, float\* %arrayidx36.i.us, align 4, !tbaa !12 %mul37.i.us = fmul float %30, 4.000000e+00 %31 = tail call float @llvm.fmuladd.f32(float %29, float 2.000000e+00, float ... %mul37.i.us) #3 %32 = tail call float @llvm.fmuladd.f32(float %29, float 5.000000e+00, float ... %31) #3 %33 = tail call float @llvm.fmuladd.f32(float %30, float 7.000000e+00, float ... %32) #3 %34 = tail call float @llvm.fmuladd.f32(float %29, float -8.000000e+00,.. float %33) #3 %35 = tail call float @llvm.fmuladd.f32(float %30, float 1.000000e+01, float ... %34) #3 %add87.i.us = add i32 %mul84.i, %conv.i.us %add89.i.us = add i32 %add87.i.us, %mul23.i %idxprom90.i.us = sext i32 %add89.i.us to i64 %arrayidx91.i.us = getelementptr inbounds float, float\* %0, i64 ... %idxprom90.i.us %36 = load float, float\* %arrayidx91.i.us, align 4, !tbaa !12 %37 = tail call float @llvm.fmuladd.f32(float %36, float -3.000000e+00, .. float %35) #3 %add100.i.us = add i32 %add87.i.us, %mul97.i %idxprom101.i.us = sext i32 %add100.i.us to i64 %arrayidx102.i.us = getelementptr inbounds float, float\* %0, i64 ... %idxprom101.i.us %38 = load float, float\* %arrayidx102.i.us, align 4, !tbaa !12 %39 = tail call float @llvm.fmuladd.f32(float %38, float 6.000000e+00, float ... %37) #3 %add111.i.us = add i32 %add87.i.us, %mul108.i %idxprom112.i.us = sext i32 %add111.i.us to i64 %arrayidx113.i.us = getelementptr inbounds float, float\* %0, i64 ... %idxprom112.i.us %40 = load float, float\* %arrayidx113.i.us, align 4, !tbaa !12 %41 = tail call float @llvm.fmuladd.f32(float %40, float -9.000000e+00, ... float %39) #3 %add192.i.us = add i32 %reass.mul.i, %conv.i.us %add121.i.us = add nuw nsw i32 %conv.i.us, 1 vector.body.preheader: pregion for entry.entry.i.preheader: %add122.i.us = add nsw i32 %add121.i.us, %add.i %idxprom193.i.us = sext i32 %add192.i.us to i64 br label %pregion for entry.entry.i br label %vector.body %idxprom123.i.us = sext i32 %add122.i.us to i64 br label %if.end.r exit.i.us %arrayidx124.i.us = getelementptr inbounds float, float\* %0, i64 ... %idxprom123.i.us %42 = load float, float\* %arrayidx124.i.us, align 4, !tbaa !12 %43 = tail call float @llvm.fmuladd.f32(float %42, float 2.000000e+00, float ... %41) #3 %add133.i.us = add nsw i32 %add121.i.us, %add32.i %idxprom134.i.us = sext i32 %add133.i.us to i64 %arrayidx135.i.us = getelementptr inbounds float, float\* %0, i64 ... %idxprom134.i.us %44 = load float, float\* %arrayidx135.i.us, align 4, !tbaa !12 %45 = tail call float @llvm.fmuladd.f32(float %44, float 4.000000e+00, float ... %43) #3 %add142.i.us = add i32 %add121.i.us, %mul21.i %add144.i.us = add i32 %add142.i.us, %mul97.i %idxprom145.i.us = sext i32 %add144.i.us to i64 %arrayidx146.i.us = getelementptr inbounds float, float\* %0, i64 ... %idxprom145.i.us %46 = load float, float\* %arrayidx146.i.us, align 4, !tbaa !12 %47 = tail call float @llvm.fmuladd.f32(float %46, float 5.000000e+00, float ... %45) #3 %add153.i.us = add i32 %add121.i.us, %mul29.i %add155.i.us = add i32 %add153.i.us, %mul97.i %idxprom156.i.us = sext i32 %add155.i.us to i64 %arrayidx157.i.us = getelementptr inbounds float, float\* %0, i64 ... %idxprom156.i.us %48 = load float, float\* %arrayidx157.i.us, align 4, !tbaa !12 %49 = tail call float @llvm.fmuladd.f32(float %48, float 7.000000e+00, float ... %47) #3 %add166.i.us = add i32 %add142.i.us, %mul108.i %idxprom167.i.us = sext i32 %add166.i.us to i64 %arrayidx168.i.us = getelementptr inbounds float, float\* %0, i64 ... %idxprom167.i.us %50 = load float, float\* %arrayidx168.i.us, align 4, !tbaa !12 %51 = tail call float @llvm.fmuladd.f32(float %50, float -8.000000e+00, .. float %49) #3 %add177.i.us = add i32 %add153.i.us, %mul108.i %idxprom178.i.us = sext i32 %add177.i.us to i64 %arrayidx179.i.us = getelementptr inbounds float, float\* %0, i64 ... %idxprom178.i.us %52 = load float, float\* %arrayidx179.i.us, align 4, !tbaa !12 %53 = tail call float @llvm.fmuladd.f32(float %52, float 1.000000e+01, float ... %51) #3 br label %if.end.r exit.i.us pregion for entry.entry.i: vector.body: % local id x.0 = phi i64 [ %55, %pregion\_for\_entry.entry.i ], [ 0, %index = phi i64 [ %index.next, %vector.body ], [ 0, %vector.body.preheader ] if.end.r exit.i.us: %idxprom193.i.us.sink = phi i64 [ %idxprom101.i.us, %if.then.i.us ], [ %22 = add nuw nsw i64 %index, %mul.i.i .. %pregion for entry.entry.i.preheader ] .. %idxprom193.i.us, %if.else.i.us] %add1.i.i = add nuw nsw i64 % local id x.0, %mul.i.i %23 = trunc i64 %22 to i32 %.sink = phi float [ %53, %if.then.i.us ], [ 0.000000e+00, %if.else.i.us ] %24 = add i32 %reass.mul.i, %23 %conv.i = trunc i64 %add1.i.i to i32 %arrayidx194.i.us = getelementptr inbounds float, float\* %1, i64 %add192.i = add i32 %reass.mul.i, %conv.i %25 = sext i32 %24 to i64 .. %idxprom193.i.us.sink %idxprom193.i = sext i32 %add192.i to i64 %26 = getelementptr inbounds float, float\* %1, i64 %25 %27 = bitcast float\* %26 to <8 x float>\* %arrayidx194.i = getelementptr inbounds float, float\* %1, i64 %idxprom193.i store float %.sink, float\* %arrayidx194.i.us, align 4, !tbaa !12, store float 0.000000e+00, float\* %arrayidx194.i, align 4, !tbaa !12, store <8 x float> zeroinitializer, <8 x float>\* %27, align 4, !tbaa !12, .. !llvm.access.group !16 %54 = add nuw nsw i64 % local id x.0.us, 1...!llvm.access.group!16 ..!llvm.access.group!16 %exitcond2.not = icmp eq i64 %54, 32 br i1 %exitcond2.not, label %pregion\_for\_end.i.loopexit, label %55 = add nuw nsw i64 % local id x.0, 1%index.next = add i64 %index, 8 %exitcond.not = icmp eq i64 %55, 32 br i1 %exitcond.not, label %pregion\_for\_end.i.loopexit8, label %28 = icmp eq i64 %index.next, 32 br i1 %28, label %pregion\_for\_end.i.loopexit9, label %vector.body, .. %pregion for entry.entry.i.us, !llvm.loop !22 .. %pregion\_for\_entry.entry.i, !llvm.loop !23 ... !llvm.loop !19 pregion\_for\_end.i.loopexit9: br label %pregion\_for\_end.i pregion\_for\_end.i.loopexit: pregion\_for\_end.i.loopexit8: br label %pregion for end.i br label %pregion\_for\_end.i pregion for end.i: 56 = add nuw nsw i64 %\_local\_id\_y.0, 1 %exitcond3.not = icmp eq i64 %56, 8 br i1 %exitcond3.not, label %Convolution3D\_kernel.exit, label ... %pregion for entry.pregion for init.i, !llvm.loop !24 Convolution3D kernel.exit: ret void CFG for 'pocl kernel Convolution3D kernel' function

%10: