%11: %..lcssa.ph.ex phi.pocl context.i = alloca [1 x [1 x [256 x float]]], align $%12 = bitcast [1 x [1 x [256 x float]]]* %..lcssa.ph.ex_phi.pocl_context.i$... to i8* call void @llvm.lifetime.start.p0i8(i64 1024, i8* nonnull %12) %13 = shl i64 %8, 8 %cmp259.i = icmp sgt i32 %6, 0 %14 = sext i32 %5 to i64 %wide.trip.count.i = zext i32 %6 to i64 %15 = add nsw i64 %wide.trip.count.i, -1 %xtraiter.i = and i64 %wide.trip.count.i, 3 %16 = icmp ult i64 %15, 3 %unroll_iter.i = sub nuw nsw i64 %wide.trip.count.i, %xtraiter.i %lcmp.mod.i = icmp eq i64 %xtraiter.i, 0 br label %pregion for entry.entry.i pregion for entry.entry.i: %_local_id_x.0 = phi i64 [0, %11], [%51, %if.end32.r_exit.i] %17 = add nuw nsw i64 %_local_id_x.0, %13 %conv.i = trunc i64 %17 to i32 %cmp.i = icmp slt i32 %conv.i, %5 br il %cmp.i, label %if.then.i, label %if.end32.r exit.i if.then.i: %sext.i = shl i64 %17, 32 %idxprom.i = ashr exact i64 %sext.i, 32 %arrayidx.i = getelementptr inbounds float, float* %1, i64 %idxprom.i store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !12, ...!llvm.access.group!16 br i1 %cmp259.i, label %for.body.lr.ph.i, label %for.end.i F for.body.lr.ph.i: %arrayidx7.i = getelementptr inbounds float, float* %0, i64 %idxprom.i br i1 %16, label %for.body.lr.ph.i.for.end.loopexit.unr-lcssa.i crit edge, ... label %for.body.i.preheader for.body.i.preheader: br label %for.body.i for.body.i: %niter.nsub.3.i8 = phi i64 [%niter.nsub.3.i, %for.body.i], [... %unroll_iter.i, %for.body.i.preheader] %indvars.īv.next.3.i5 = phi i64 [%indvars.iv.next.3.i, %for.body.i], [0, ... %for.body.i.preheader] %18 = phi float [%38, %for.body.i], [0.000000e+00, %for.body.i.preheader] %19 = mul nsw i64 %indvars.iv.next.3.i5, %14 %20 = add nsw i64 %19, %idxprom.i %arrayidx5.i = getelementptr inbounds float, float* %2, i64 %20 %21 = load float, float* %arrayidx5.i, align 4, !tbaa !12 %22 = load float, float* %arrayidx7.i, align 4, !tbaa !12 %sub.i = fsub float %21, %22 %23 = tail call float @llvm.fmuladd.f32(float %sub.i, float %sub.i, float store float %23, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group ... !16 %indvars.iv.next.i = or i64 %indvars.iv.next.3.i5, 1 %24 = mul nsw i64 %indvars.iv.next.i, %14 %25 = add nsw i64 %24, %idxprom.i %arrayidx5.1.i = getelementptr inbounds float, float* %2, i64 %25 %26 = load float, float* %arrayidx5.1.i, align 4, !tbaa !12 %27 = load float, float* %arrayidx7.i, align 4, !tbaa !12 %sub.1.i = fsub float %26, %27 %28 = tail call float @llvm.fmuladd.f32(float %sub.1.i, float %sub.1.i, ... float %23) #3 for.body.lr.ph.i.for.end.loopexit.unr-lcssa.i crit edge: store float %28, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group %.phi.trans.insert = getelementptr $[1 \times [1 \times [256 \times float]]]$, $[1 \times [1 \times [256 \times float]]]$... !16 ... x float]]]* %..lcssa.ph.ex_phi.pocl_context.i, i64 0, i64 0, i64 0, i64 %indvars.iv.next.1.i = or i64 %indvars.iv.next.3.i5, 2 % local id x.0 %29 = mul nsw i64 %indvars.iv.next.1.i, %14 %.pre18 = load float, float* %.phi.trans.insert, align 4 %30 = add nsw i64 %29, %idxprom.i br label %for.end.loopexit.unr-lcssa.i %arrayidx5.2.i = getelementptr inbounds float, float* %2, i64 %30 %31 = load float, float* %arrayidx5.2.i, align 4, !tbaa !12 %32 = load float, float* %arrayidx7.i, align 4, !tbaa !12 %sub.2.i = fsub float %31, %32 %33 = tail call float @llvm.fmuladd.f32(float %sub.2.i, float %sub.2.i, ... float %28) #3 store float %33, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group ... !16 %indvars.iv.next.2.i = or i64 %indvars.iv.next.3.i5, 3 %34 = mul nsw i64 %indvars.iv.next.2.i, %14 %35 = add nsw i64 %34, %idxprom.i %arrayidx5.3.i = getelementptr inbounds float, float* %2, i64 %35 %36 = load float, float* %arrayidx5.3.i, align 4, !tbaa !12 %37 = load float, float* %arrayidx7.i, align 4, !tbaa !12 %sub.3.i = fsub float %36, %37 %38 = tail call float @llvm.fmuladd.f32(float %sub.3.i, float %sub.3.i, ... float %33) #3 store float %38, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group %indvars.iv.next.3.i = add nuw nsw i64 %indvars.iv.next.3.i5, 4 %niter.nsub.3.i = add i64 %niter.nsub.3.i8, -4 %niter.ncmp.3.i = icmp eq i64 %niter.nsub.3.i, 0 br i1 %niter.ncmp.3.i, label %for.end.loopexit.unr-lcssa.loopexit.i, label ... %for.body.i for.end.loopexit.unr-lcssa.loopexit.i: %.lcssa = phi float [%38, %for.body.i] %indvars.iv.next.3.i.lcssa = phi i64 [%indvars.iv.next.3.i, %for.body.i] %39 = getelementptr [1 x [1 x [256 x float]]], [1 x [1 x [256 x float]]]*%..lcssa.ph.ex_phi.pocl_context.i, i64 0, i64 0, i64 0, i64 %_local_id_x.0 store float %.lcssa, float* %39, align 4, !llvm.access.group !16 br label %for.end.loopexit.unr-lcssa.i for.end.loopexit.unr-lcssa.i: for.end.loopexit.unr-lcssa.i:
%40 = phi i64 [0, %for.body.lr.ph.i.for.end.loopexit.unr-lcssa.i_crit_edge
...], [%indvars.iv.next.3.i.lcssa, %for.end.loopexit.unr-lcssa.loopexit.i]
%41 = phi float [0.000000e+00,
... %for.body.lr.ph.i.for.end.loopexit.unr-lcssa.i_crit_edge], [%.lcssa,
... %for.end.loopexit.unr-lcssa.loopexit.i]
%42 = phi float [%.pre18,
... %for.body.lr.ph.i.for.end.loopexit.unr-lcssa.i_crit_edge], [%.lcssa,
... %for.end.loopexit.unr-lcssa.loopexit.i]
br i1 %lcmp.mod.i, label %for.end.i, label %for.body.epil.i.preheader for.body.epil.i.preheader: br label %for.body.epil.i for.body.epil.i: %epil.iter.sub.i14 = phi i64 [%epil.iter.sub.i, %for.body.epil.i], [... %xtraiter.i, %for.body.epil.i.preheader] %indvars.iv.next.epil.i12 = phi i64 [%indvars.iv.next.epil.i, ... %for.body.epil.i], [%40, %for.body.epil.i.preheader] %43 = phi float [%48, %for.body.epil.i], [%41, %for.body.epil.i.preheader %44 = mul nsw i64 %indvars.iv.next.epil.i12, %14 %45 = add nsw i64 %44, %idxprom.i %arrayidx5.epil.i = getelementptr inbounds float, float* %2, i64 %45 %46 = load float, float* %arrayidx5.epil.i, align 4, !tbaa !12 %47 = load float, float* %arrayidx7.i, align 4, !tbaa !12 %sub.epil.i = fsub float %46, %47 %48 = tail call float @llvm.fmuladd.f32(float %sub.epil.i, float ... %sub.epil.i, float %43) #3 store float %48, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group %indvars.iv.next.epil.i = add nuw nsw i64 %indvars.iv.next.epil.i12, 1 %epil.iter.sub.i = add nsw i64 %epil.iter.sub.i14, -1 %epil.iter.cmp.i = icmp eq i64 %epil.iter.sub.i, 0 br i1 %epil.iter.cmp.i, label %for.end.i.loopexit, label %for.body.epil.i, ... !llvm.loop !18 F for.end.i.loopexit: %.lcssa23 = phi float [%48, %for.body.epil.i] br label %for.end.i for.end.i: %49 = phi float [%42, %for.end.loopexit.unr-lcssa.i], [0.000000e+00]... %if.then.i], [%.lcssa23, %for.end.i.loopexit] %div.i = fdiv float %49, %3, !fpmath !20 %50 = tail call float @llvm.sqrt.f32(float %div.i) #3 %cmp27.i = fcmp ugt float %50, %4 %storemerge.i = select i1 %cmp27.i, float %50, float 1.000000e+00 store float %storemerge.i, float* %arrayidx.i, align 4, !tbaa !12, ...!llvm.access.group!16 br label %if.end32.r exit.i if.end32.r exit.i: %51 = add nuw nsw i64 % local id x.0, 1 %exitcond = icmp eq i64 %51, $2\overline{5}6$ br i1 %exitcond, label %std kernel.exit, label %pregion for entry.entry.i, ... !llvm.loop !21 std kernel.exit:

call void @llvm.lifetime.end.p0i8(i64 1024, i8* nonnull %12)