%11: %12 = shl i64 %8, 5%13 = shl i64 %9, 3%cmp761.i = icmp sgt i32 %5, 0 %wide.trip.count.i = zext i32 %5 to i64 %xtraiter.i = and i64 %wide.trip.count.i, 1 %14 = icmp eq i32 %5, 1%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.pregion\_for\_init.i pregion for entry.pregion for init.i: % local id y.0 = phi i64 [0, %11], [%44, %pregion for end.i] $\%\overline{15} = \overline{a}d\overline{d}$  nuw nsw i64 % local id v.0, %13 %conv2.i = trunc i64 %15 to i32%cmp.i = icmp slt i32 %conv2.i, %6 %mul.i = mul nsw i32 %conv2.i, %6 %mul9.i = mul nsw i32 %conv2.i, %5 %16 = sext i32 %mul9.i to i64 br i1 %cmp.i, label %pregion for entry.entry.i.us.preheader, label ... %pregion for end.i F 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 [ %43, %if.end.i.us ], [ 0,... %pregion for entry.entry.i.us.preheader ] %17 = add nuw nsw i64 % local id x.0.us, %12 %conv.i.us = trunc i64 %17 to i32%cmp4.i.us = icmp slt i32 %conv.i.us, %6 br i1 %cmp4.i.us, label %if.then.i.us, label %if.end.i.us F if.then.i.us: %add.i.us = add nsw i32 %mul.i, %conv.i.us %idxprom.i.us = sext i32 %add.i.us to i64 %arrayidx.i.us = getelementptr inbounds float, float\* %2, i64 %idxprom.i.us %18 = load float, float\* %arrayidx.i.us, align 4, !tbaa !12 %mul6.i.us = fmul float %18, %4 store float %mul6.i.us, float\* %arrayidx.i.us, align 4, !tbaa !12, ...!llvm.access.group!16 br i1 %cmp761.i, label %for.body.lr.ph.i.us, label %if.end.i.us for.body.lr.ph.i.us: %mul14.i.us = mul nsw i32 %conv.i.us, %5 %19 = sext i32 %mul14.i.us to i64 br i1 %14, label %if.end.loopexit.unr-lcssa.i.us, label ... %for.body.i.us.preheader for.body.i.us.preheader: br label %for.body.i.us for.body.i.us: %niter.nsub.1.i8.us = phi i64 [ %niter.nsub.1.i.us, %for.body.i.us ], [ ... %unroll iter.i, %for.body.i.us.preheader ] %indvars.iv.next.1.i5.us = phi i64 [ %indvars.iv.next.1.i.us, %for.body.i.us ...], [0, %for.body.i.us.preheader] %add33.1.i2.us = phi float [ %add33.1.i.us, %for.body.i.us ], [ %mul6.i.us, ... %for.body.i.us.preheader ] %20 = add nsw i64 %indvars.iv.next.1.i5.us, %16 %arrayidx12.i.us = getelementptr inbounds float, float\* %0, i64 %20 %21 = load float, float\* %arrayidx12.i.us, align 4, !tbaa !12 %mul13.i.us = fmul float %21, %3 %22 = add nsw i64 %indvars.iv.next.1.i5.us, %19 %arrayidx17.i.us = getelementptr inbounds float, float\* %1, i64 %22 %23 = load float, float\* %arrayidx17.i.us, align 4, !tbaa !12 %arrayidx22.i.us = getelementptr inbounds float, float\* %1, i64 %20 %24 = load float, float\* %arrayidx22.i.us, align 4, !tbaa !12 %mul23.i.us = fmul float %24, %3 %arrayidx27.i.us = getelementptr inbounds float, float\* %0, i64 %22 %25 = load float, float\* %arrayidx27.i.us, align 4, !tbaa !12 %mul28.i.us = fmul float %mul23.i.us, %25 %26 = tail call float @llvm.fmuladd.f32(float %mul13.i.us, float %23, float ... %mul28.i.us) #2 %add33.i.us = fadd float %add33.1.i2.us, %26 store float %add33.i.us, float\* %arrayidx.i.us, align 4, !tbaa !12, ...!llvm.access.group!16 %indvars.iv.next.i.us = or i64 %indvars.iv.next.1.i5.us, 1 %27 = add nsw i64 %indvars.iv.next.i.us, %16 %arrayidx12.1.i.us = getelementptr inbounds float, float\* %0, i64 %27 %28 = load float, float\* %arrayidx12.1.i.us, align 4, !tbaa !12 %mul13.1.i.us = fmul float %28, %3 %29 = add nsw i64 %indvars.iv.next.i.us, %19 %arrayidx17.1.i.us = getelementptr inbounds float, float\* %1, i64 %29 %30 = load float, float\* %arrayidx17.1.i.us, align 4, !tbaa !12 %arrayidx22.1.i.us = getelementptr inbounds float, float\* %1, i64 %27 %31 = load float, float\* %arrayidx22.1.i.us, align 4, !tbaa !12 %mul23.1.i.us = fmul float %31, %3 %arrayidx27.1.i.us = getelementptr inbounds float, float\* %0, i64 %29 %32 = load float, float\* %arrayidx27.1.i.us, align 4, !tbaa !12 %mul28.1.i.us = fmul float %mul23.1.i.us, %32 %33 = tail call float @llvm.fmuladd.f32(float %mul13.1.i.us, float %30, ... float %mul28.1.i.us) #2 %add33.1.i.us = fadd float %add33.i.us, %33 store float %add33.1.i.us, float\* %arrayidx.i.us, align 4, !tbaa !12, ...!llvm.access.group!16 %indvars.iv.next.1.i.us = add nuw nsw i64 %indvars.iv.next.1.i5.us, 2 %niter.nsub.1.i.us = add i64 %niter.nsub.1.i8.us, -2 %niter.ncmp.1.i.us = icmp eq i64 %niter.nsub.1.i.us, 0 br i1 %niter.ncmp.1.i.us, label %if.end.loopexit.unr-lcssa.i.us.loopexit, ... label %for.body.i.us F if.end.loopexit.unr-lcssa.i.us.loopexit: %add33.1.i.us.lcssa = phi float [ %add33.1.i.us, %for.body.i.us ] %indvars.iv.next.1.i.us.lcssa = phi i64 [ %indvars.iv.next.1.i.us, . %for.body.i.us ] br label %if.end.loopexit.unr-lcssa.i.us if.end.loopexit.unr-lcssa.i.us: %34 = phi float [ %mul6.i.us, %for.body.lr.ph.i.us ], [ %add33.1.i.us.lcssa, ... %if.end.loopexit.unr-lcssa.i.us.loopexit ] %35 = phi i64 [ 0, %for.body.lr.ph.i.us ], [ %indvars.iv.next.1.i.us.lcssa, ... %if.end.loopexit.unr-lcssa.i.us.loopexit ] br i1 %lcmp.mod.i, label %if.end.i.us, label %for.body.epil.i.us for.body.epil.i.us: %36 = add nsw i64 %35, %16 %arrayidx12.epil.i.us = getelementptr inbounds float, float\* %0, i64 %36 %37 = load float, float\* %arrayidx12.epil.i.us, align 4, !tbaa !12 %mul13.epil.i.us = fmul float %37, %3 %38 = add nsw i64 %35, %19 %arrayidx17.epil.i.us = getelementptr inbounds float, float\* %1, i64 %38 %39 = load float, float\* %arrayidx17.epil.i.us, align 4, !tbaa !12 %arrayidx22.epil.i.us = getelementptr inbounds float, float\* %1, i64 %36 %40 = load float, float\* %arrayidx22.epil.i.us, align 4, !tbaa !12 %mul23.epil.i.us = fmul float %40, %3 %arrayidx27.epil.i.us = getelementptr inbounds float, float\* %0, i64 %38 %41 = load float, float\* %arrayidx27.epil.i.us, align 4, !tbaa !12 %mul28.epil.i.us = fmul float %mul23.epil.i.us, %41 %42 = tail call float @llvm.fmuladd.f32(float %mul13.epil.i.us, float %39, ... float %mul28.epil.i.us) #2 %add33.epil.i.us = fadd float %34, %42 store float %add33.epil.i.us, float\* %arrayidx.i.us, align 4, !tbaa !12, ...!llvm.access.group!16 br label %if.end.i.us if.end.i.us: %43 = add nuw nsw i64 % local id x.0.us, 1%exitcond = icmp eq i64  $\sqrt[6]{43}$ ,  $\sqrt[3]{2}$ br i1 %exitcond, label %pregion\_for\_end.i.loopexit, label ... %pregion for entry.entry.i.us, !llvm.loop!19 pregion\_for\_end.i.loopexit: br label %pregion for end.i pregion for end.i:  $^{1}\%44 = add$  nuw nsw i64 % local id y.0, 1 %exitcond10 = icmp eq i6 $\overline{4}$  %44 $\overline{4}$ , 8 br i1 %exitcond10, label %syr2k kernel.exit, label ... %pregion for entry.pregion for init.i, !llvm.loop !21 syr2k kernel.exit: