```
%10:
                                                                                    %11 = shl i64 \%7, 5
                                                                                    %12 = \text{shl } i64 \%8, 3
                                                                                     %cmp742.i = icmp sgt i32 %4, 0
                                                                                     %wide.trip.count.i = zext i32 %4 to i64
                                                                                     %13 = add nsw i64 %wide.trip.count.i, -1
                                                                                     %xtraiter.i = and i64 %wide.trip.count.i, 3
                                                                                     %14 = icmp ult i64 %13, 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.pregion for init.i
                                                                                    pregion for entry.pregion for init.i:
                                                                                     .
%_local_id_y.0 = phi i64 [0, %10 ], [ %49, %pregion_for_end.i ]
                                                                                     \%\overline{1}5 = \overline{a}d\overline{d} nuw nsw i64 % local id y.0, %12
                                                                                     \%conv2.i = trunc i64 \%15 to i32
                                                                                     %cmp.i = icmp slt i32 %conv2.i, %5
                                                                                     %mul.i = mul nsw i32 %conv2.i, %5
                                                                                     %mul9.i = mul nsw i32 %conv2.i, %4
                                                                                     %16 = sext i32 %mul9.i to i64
                                                                                     br label %pregion for entry.entry.i
                                                            pregion for entry.entry.i:
                                                            % [\cos \overline{1}] id [x.0] = phi i64 [0, %pregion for entry.pregion for init.i], [
                                                            %17 = add nuw nsw i64 %_local_id_x.0, %11
                                                            %conv.i = trunc i64 %17 to i32
                                                            %cmp4.i = icmp slt i32 %conv.i, %5
                                                            %or.cond.i = and i1 %cmp.i, %cmp4.i
                                                            br i1 %or.cond.i, label %if.then.i, label %if.end.i
                                                                                                                    F
                               if.then.i:
                               %add.i = add nsw i32 %mul.i, %conv.i
                               %idxprom.i = sext i32 %add.i to i64
                               %arrayidx.i = getelementptr inbounds float, float* %1, i64 %idxprom.i
                               %18 = load float, float* %arrayidx.i, align 4, !tbaa !12 %mul6.i = fmul float %18, %3
                               store float %mul6.i, float* %arrayidx.i, align 4, !tbaa !12,
                               ..!llvm.access.group!16
                               br i1 %cmp742.i, label %for.body.lr.ph.i, label %if.end.i
                                                                                        F
for.body.lr.ph.i:
%mul14.i = mul nsw i32 %conv.i, %4
 %19 = sext i32 %mul14.i to i64
br i1 %14, label %if.end.loopexit.unr-lcssa.i, label %for.body.i.preheader
                                               for.body.i.preheader:
                                                br label %for.body.i
                    for.body.i:
                    %niter.nsub.3.i7 = phi i64 [ %niter.nsub.3.i, %for.body.i ], [
                      %unroll iter.i, %for.body.i.preheader ]
                    %indvars.īv.next.3.i4 = phi i64 [ %indvars.iv.next.3.i, %for.body.i ], [ 0,
                      %for.body.i.preheader ]
                    %20 = phi float [ %40, %for.body.i ], [ %mul6.i, %for.body.i.preheader ]
                    %21 = add nsw i64 %indvars.iv.next.3.i4, %16
                    %arrayidx12.i = getelementptr inbounds float, float* %0, i64 %21
                    %22 = load float, float* %arrayidx12.i, align 4, !tbaa !12
                    %mul13.i = fmul float %22, %2
                    %23 = add nsw i64 %indvars.iv.next.3.i4, %19
                    %arrayidx17.i = getelementptr inbounds float, float* %0, i64 %23
                    %24 = load float, float* %arrayidx17.i, align 4, !tbaa !12
                    %25 = tail call float @llvm.fmuladd.f32(float %mul13.i, float %24, float
                    ... %20) #2
                    store float %25, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                    %indvars.iv.next.i = or i64 %indvars.iv.next.3.i4, 1
                    %26 = add nsw i64 %indvars.iv.next.i, %16
                    %arrayidx12.1.i = getelementptr inbounds float, float* %0, i64 %26
                    %27 = load float, float* %arrayidx12.1.i, align 4, !tbaa !12
                    %mul13.1.i = fmul float %27, %2
                    %28 = add nsw i64 %indvars.iv.next.i, %19
                    %arrayidx17.1.i = getelementptr inbounds float, float* %0, i64 %28
                    %29 = load float, float* %arrayidx17.1.i, align 4, !tbaa !12
                    %30 = tail call float @llvm.fmuladd.f32(float %mul13.1.i, float %29, float
                    ... %25) #2
                    store float %30, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                    %indvars.iv.next.1.i = or i64 %indvars.iv.next.3.i4, 2
                    %31 = add nsw i64 %indvars.iv.next.1.i, %16
                    %arrayidx12.2.i = getelementptr inbounds float, float* %0, i64 %31
                    %32 = load float, float* %arrayidx12.2.i, align 4, !tbaa !12
                    %mul13.2.i = fmul float %32, %2
                    %33 = add nsw i64 %indvars.iv.next.1.i, %19
                    %arrayidx17.2.i = getelementptr inbounds float, float* %0, i64 %33
                    %34 = load float, float* %arrayidx17.2.i, align 4, !tbaa !12
                    %35 = tail call float @llvm.fmuladd.f32(float %mul13.2.i, float %34, float
                    ... %30) #2
                    store float %35, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                    %indvars.iv.next.2.i = or i64 %indvars.iv.next.3.i4, 3
                    %36 = add nsw i64 %indvars.iv.next.2.i, %16
                    %arrayidx12.3.i = getelementptr inbounds float, float* %0, i64 %36
                    %37 = load float, float* %arrayidx12.3.i, align 4, !tbaa !12
                    %mul13.3.i = fmul float %37, %2
                    %38 = add nsw i64 %indvars.iv.next.2.i, %19
                    %arrayidx17.3.i = getelementptr inbounds float, float* %0, i64 %38
                    %39 = load float, float* %arrayidx17.3.i, align 4, !tbaa !12
                    %40 = tail call float @llvm.fmuladd.f32(float %mul13.3.i, float %39, float
                    ... %35) #2
                    store float %40, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                    %indvars.iv.next.3.i = add nuw nsw i64 %indvars.iv.next.3.i4, 4
                    %niter.nsub.3.i = add i64 %niter.nsub.3.i7, -4
                    %niter.ncmp.3.i = icmp eq i64 %niter.nsub.3.i, 0
                    br i1 %niter.ncmp.3.i, label %if.end.loopexit.unr-lcssa.i.loopexit, label
                    ... %for.body.i
                                      Τ
          if.end.loopexit.unr-lcssa.i.loopexit:
          %.lcssa = phi float [ %40, %for.body.i ]
           %indvars.iv.next.3.i.lcssa = phi i64 [ %indvars.iv.next.3.i, %for.body.i ]
          br label %if.end.loopexit.unr-lcssa.i
if.end.loopexit.unr-lcssa.i:
%41 = phi float [ %mul6.i, %for.body.lr.ph.i ], [ %.lcssa,
... %if.end.loopexit.unr-lcssa.i.loopexit ]
%42 = phi i64 [ 0, %for.body.lr.ph.i ], [ %indvars.iv.next.3.i.lcssa,
... %if.end.loopexit.unr-lcssa.i.loopexit ]
br i1 %lcmp.mod.i, label %if.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.i13 = phi i64 [ %epil.iter.sub.i, %for.body.epil.i ], [
                   ... %xtraiter.i, %for.body.epil.i.preheader ]
                   %indvars.iv.next.epil.i11 = phi i64 [ %indvars.iv.next.epil.i,
                  ... %for.body.epil.i ], [ %42, %for.body.epil.i.preheader ] %43 = phi float [ %48, %for.body.epil.i ], [ %41, %for.body.epil.i.preheader
                   %44 = add nsw i64 %indvars.iv.next.epil.i11, %16
                   %arrayidx12.epil.i = getelementptr inbounds float, float* %0, i64 %44 %45 = load float, float* %arrayidx12.epil.i, align 4, !tbaa !12 %mul13.epil.i = fmul float %45, %2
                   %46 = add nsw i64 %indvars.iv.next.epil.i11, %19
                   %arrayidx17.epil.i = getelementptr inbounds float, float* %0, i64 %46
                   %47 = load float, float* %arrayidx17.epil.i, align 4, !tbaa !12 %48 = tail call float @llvm.fmuladd.f32(float %mul13.epil.i, float %47,
                   ... float %43) #2
                   store float %48, float* %arrayidx.i, align 4, !tbaa !12, !llvm.access.group
                   ... !16
                   %indvars.iv.next.epil.i = add nuw nsw i64 %indvars.iv.next.epil.i11, 1
                   %epil.iter.sub.i = add nsw i64 %epil.iter.sub.i13, -1
                   %epil.iter.cmp.i = icmp eq i64 %epil.iter.sub.i, 0
                   br il %epil.iter.cmp.i, label %if.end.i.loopexit, label %for.body.epil.i,
                   ... !llvm.loop !19
                                                                               F
                                                       if.end.i.loopexit:
                                                        br label %if.end.i
                                                    if.end.i:
                                                    %50 = add nuw nsw i64 % local id x.0, 1
                                                     %exitcond = icmp eq i64 \%50, 3\overline{2}
                                                     br i1 %exitcond, label %pregion for end.i, label %pregion for entry.entry.i,
                                                    ...!llvm.loop!23
                                                                                                                 F
                                                                     pregion for end.i:
                                                                     %49 = add nuw nsw i64 \% local id y.0, 1
                                                                     \%exitcond14 = icmp eq i6\overline{4} %49, 8
                                                                     br i1 %exitcond14, label %syrk kernel.exit, label
                                                                     ... %pregion for entry.pregion for init.i, !llvm.loop!21
                                                                          syrk kernel.exit:
                                                                          ret void
```

CFG for 'pocl kernel syrk kernel' function