

polly.par.setup:
%polly.par.LBPtr = alloca i64
%polly.par.UBPtr = alloca i64
%polly.par.userContext1 = bitcast i8* %polly.par.userContext to { i32, i64,
... double*, double*, double* }*
%0 = getelementptr inbounds { i32, i64, double*, double*, double* }, { i32,
... i64, double*, double*, double* }* %polly.par.userContext1, i32 0, i32 0
%polly.subfunc.arg.n = load i32, i32* %0
%1 = getelementptr inbounds { i32, i64, double*, double*, double* }, { i32,
... i64, double*, double*, double* }* %polly.par.userContext1, i32 0, i32 1
%polly.subfunc.arg. = load i64, i64* %1
%2 = getelementptr inbounds { i32, i64, double*, double*, double* }, { i32,
... i64, double*, double*, double* }* %polly.par.userContext1, i32 0, i32 2
%polly.subfunc.arg.x1 = load double*, double** %2
%3 = getelementptr inbounds { i32, i64, double*, double*, double* }, { i32,
... i64, double*, double*, double* }* %polly.par.userContext1, i32 0, i32 3
%polly.subfunc.arg.A = load double*, double** %3
%4 = getelementptr inbounds { i32, i64, double*, double*, double* }, { i32,
... i64, double*, double*, double* }* %polly.par.userContext1, i32 0, i32 4
%polly.subfunc.arg.y_1 = load double*, double** %4
br label %polly.par.checkNext

polly.par.checkNext:
%5 = call i8 @GOMP_loop_runtime_next(i64* %polly.par.LBPtr, i64*
... %polly.par.UBPtr)
%6 = icmp ne i8 %5, 0
br i1 %6, label %polly.par.loadIVBounds, label %polly.par.exit

T

F

polly.par.loadIVBounds:
%polly.par.LB = load i64, i64* %polly.par.LBPtr
%polly.par.UB = load i64, i64* %polly.par.UBPtr
%polly.par.UBAdjusted = sub i64 %polly.par.UB, 1
br label %polly.loop_preheader

polly.par.exit:
call void @GOMP_loop_end_nowait()
ret void

polly.loop_preheader:
%9 = zext i32 %polly.subfunc.arg.n to i64
br label %polly.loop_header

polly.loop_header:
%polly.indvar = phi i64 [%polly.par.LB, %polly.loop_preheader], [
... %polly.indvar_next, %polly.loop_exit4]
%7 = sext i32 %polly.subfunc.arg.n to i64
%8 = sub nsw i64 %7, 1
%pexp.p_div_q = udiv i64 %8, 32
br label %polly.loop_preheader3

polly.loop_preheader3:
br label %polly.loop_header2

polly.loop_header2:
%polly.indvar5 = phi i64 [0, %polly.loop_preheader3], [
... %polly.indvar_next6, %polly.loop_exit11]
%10 = mul nsw i64 32, %polly.indvar
%11 = sext i32 %polly.subfunc.arg.n to i64
%12 = sub nsw i64 %11, %10
%13 = sub nsw i64 %12, 1
%14 = icmp slt i64 31, %13
%15 = select i1 %14, i64 31, i64 %13
br label %polly.loop_if

polly.loop_if:
%polly.loop_guard = icmp sle i64 0, %15
br i1 %polly.loop_guard, label %polly.loop_preheader10, label
... %polly.loop_exit11

T

F

polly.loop_preheader10:
br label %polly.loop_header9

polly.loop_header9:
%polly.indvar12 = phi i64 [0, %polly.loop_preheader10], [
... %polly.indvar_next13, %polly.loop_exit19]
%16 = mul nsw i64 32, %polly.indvar5
%17 = sext i32 %polly.subfunc.arg.n to i64
%18 = sub nsw i64 %17, %16
%19 = sub nsw i64 %18, 1
%20 = icmp slt i64 31, %19
%21 = select i1 %20, i64 31, i64 %19
br label %polly.loop_if16

polly.loop_if16:
%polly.loop_guard20 = icmp sle i64 0, %21
br i1 %polly.loop_guard20, label %polly.loop_preheader18, label
... %polly.loop_exit19

T

F

polly.loop_preheader18:
br label %polly.loop_header17

polly.loop_header17:
%polly.indvar21 = phi i64 [0, %polly.loop_preheader18], [
... %polly.indvar_next22, %polly.stmt.for.body8]
%22 = mul nsw i64 32, %polly.indvar
%23 = add nsw i64 %22, %polly.indvar12
%24 = mul nsw i64 32, %polly.indvar5
%25 = add nsw i64 %24, %polly.indvar21
br label %polly.stmt.for.body8

polly.stmt.for.body8:
%scevgep = getelementptr double, double* %polly.subfunc.arg.x1, i64 %23
%_p_scalar_ = load double, double* %scevgep, align 8, !alias.scope !1,
... !noalias !3
%26 = mul i64 %9, %23
%27 = add i64 %25, %26
%scevgep25 = getelementptr double, double* %polly.subfunc.arg.A, i64 %27
%_p_scalar_26 = load double, double* %scevgep25, align 8, !alias.scope !7,
... !noalias !8
%scevgep27 = getelementptr double, double* %polly.subfunc.arg.y_1, i64 %25
%_p_scalar_28 = load double, double* %scevgep27, align 8, !alias.scope !4,
... !noalias !9
%p_mul = fmul double %_p_scalar_26, %_p_scalar_28
%p_add15 = fadd double %_p_scalar_, %p_mul
%scevgep29 = getelementptr double, double* %polly.subfunc.arg.x1, i64 %23
store double %p_add15, double* %scevgep29, align 8, !alias.scope !1,
... !noalias !3
%28 = trunc i64 %25 to i32
%29 = add i32 %28, 1
%p_exitcond17 = icmp ne i32 %29, %polly.subfunc.arg.n
%polly.indvar_next22 = add nsw i64 %polly.indvar21, 1
%polly.adjust_ub23 = sub i64 %21, 1
%polly.loop_cond24 = icmp sle i64 %polly.indvar21, %polly.adjust_ub23
br i1 %polly.loop_cond24, label %polly.loop_header17, label
... %polly.loop_exit19

T

F

polly.loop_exit19:
%polly.indvar_next13 = add nsw i64 %polly.indvar12, 1
%polly.adjust_ub14 = sub i64 %15, 1
%polly.loop_cond15 = icmp sle i64 %polly.indvar12, %polly.adjust_ub14
br i1 %polly.loop_cond15, label %polly.loop_header9, label %polly.loop_exit11

T

F

polly.loop_exit11:
%polly.indvar_next6 = add nsw i64 %polly.indvar5, 1
%polly.adjust_ub7 = sub i64 %pexp.p_div_q, 1
%polly.loop_cond8 = icmp sle i64 %polly.indvar5, %polly.adjust_ub7
br i1 %polly.loop_cond8, label %polly.loop_header2, label %polly.loop_exit4

T

F

polly.loop_exit4:
%polly.indvar_next = add nsw i64 %polly.indvar, 1
%polly.adjust_ub = sub i64 %polly.par.UBAdjusted, 1
%polly.loop_cond = icmp sle i64 %polly.indvar, %polly.adjust_ub
br i1 %polly.loop_cond, label %polly.loop_header, label %polly.loop_exit

T

F

polly.loop_exit:
br label %polly.par.checkNext