

```
%8:
%.lcssa.ph.ex_phi.pocl_context.i = alloca [1 x [1 x [256 x float]]], align
... 64
%9 = 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 %9)
%10 = shl i64 %5, 8
%cmp561.i = icmp sgt i32 %3, 0
%11 = sext i32 %2 to i64
%wide.trip.count.i = zext i32 %3 to i64
%12 = add nsw i64 %wide.trip.count.i, -1
%trailer.i = and i64 %wide.trip.count.i, 3
%13 = icmp ult i64 %12, 3
%unroll_iter.i = sub nsw i64 %wide.trip.count.i, %trailer.i
%icmp.mod.i = icmp eq i64 %xtrailer.i, 0
br label %pregion_for_entry.entry.i
```

```
pregion_for_entry.entry.i:
%local_id_x.0 = phi i64 [ 0, %8 ], [ %85, %if.end.i ]
%14 = add nuw nsw i64 %local_id_x.0, %10
%conv.i = trunc i64 %14 to i32
%cmp.i = icmp slt i32 %conv.i, %2
br i1 %cmp.i, label %for.body.lr.ph.i, label %if.end.i
```

```
for.body.lr.ph.i:
%mul.i = mul nsw i32 %conv.i, %2
%sext.i = shl i64 %14, 32
%15 = ashr exact i64 %sext.i, 32
%16 = sext i32 %mul.i to i64
br i1 %cmp561.i, label %for.body.us.preheader.i, label %for.body.preheader.i
```

```
for.body.us.preheader.i:
%26 = 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
%27 = bitcast float* %26 to i32*
br label %for.body.us.i
```

```
for.body.us.i:
%indvars.iv.next70.us.i23 = phi i64 [ %indvars.iv.next70.us.i,
... %for.end.loopexit.us.i ], [ %15, %for.body.us.preheader.i ]
%28 = add nsw i64 %indvars.iv.next70.us.i23, %16
%arrayidx.us.i = getelementptr inbounds float, float* %0, i64 %28
store float 0.000000e+00, float* %arrayidx.us.i, align 4, !tbaa !12,
... !llvm.access.group !16
br i1 %13, label %for.body.us.i.for.end.loopexit.us.unr-lcssa.i_crit_edge,
... label %for.body7.us.i.preheader
```

```
for.body.us.i.for.end.loopexit.us.unr-lcssa.i_crit_edge:
%pre = load i32, i32* %27, align 4
br label %for.end.loopexit.us.unr-lcssa.i
```

```
for.body7.us.i.preheader:
br label %for.body7.us.i
```

```
for.body7.us.i:
%niter.nsub.3.i15 = phi i64 [ %niter.nsub.3.i, %for.body7.us.i ], [
... %unroll_iter.i, %for.body7.us.i.preheader ]
%indvars.iv.next.us.3.i12 = phi i64 [ %indvars.iv.next.us.3.i,
... %for.body7.us.i ], [ 0, %for.body7.us.i.preheader ]
%29 = phi float [ %53, %for.body7.us.i ], [ 0.000000e+00,
... %for.body7.us.i.preheader ]
%30 = mul nsw i64 %indvars.iv.next.us.3.i12, %11
%31 = add nsw i64 %30, %15
%arrayidx11.us.i = getelementptr inbounds float, float* %1, i64 %31
%32 = load float, float* %arrayidx11.us.i, align 4, !tbaa !12
%33 = add nsw i64 %30, %indvars.iv.next70.us.i23
%arrayidx15.us.i = getelementptr inbounds float, float* %1, i64 %33
%34 = load float, float* %arrayidx15.us.i, align 4, !tbaa !12
%35 = tail call float @llvm.fmuladd.f32(float %32, float %34, float %29) #3
store float %35, float* %arrayidx.us.i, align 4, !tbaa !12,
... !llvm.access.group !16
%indvars.iv.next.us.i = or i64 %indvars.iv.next.us.3.i12, 1
%36 = mul nsw i64 %indvars.iv.next.us.i, %11
%37 = add nsw i64 %36, %15
%arrayidx11.us.1.i = getelementptr inbounds float, float* %1, i64 %37
%38 = load float, float* %arrayidx11.us.1.i, align 4, !tbaa !12
%39 = add nsw i64 %36, %indvars.iv.next70.us.i23
%arrayidx15.us.1.i = getelementptr inbounds float, float* %1, i64 %39
%40 = load float, float* %arrayidx15.us.1.i, align 4, !tbaa !12
%41 = tail call float @llvm.fmuladd.f32(float %38, float %40, float %35) #3
store float %41, float* %arrayidx.us.i, align 4, !tbaa !12,
... !llvm.access.group !16
%indvars.iv.next.us.1.i = or i64 %indvars.iv.next.us.3.i12, 2
%42 = mul nsw i64 %indvars.iv.next.us.1.i, %11
%43 = add nsw i64 %42, %15
%arrayidx11.us.2.i = getelementptr inbounds float, float* %1, i64 %43
%44 = load float, float* %arrayidx11.us.2.i, align 4, !tbaa !12
%45 = add nsw i64 %42, %indvars.iv.next70.us.i23
%arrayidx15.us.2.i = getelementptr inbounds float, float* %1, i64 %45
%46 = load float, float* %arrayidx15.us.2.i, align 4, !tbaa !12
%47 = tail call float @llvm.fmuladd.f32(float %44, float %46, float %41) #3
store float %47, float* %arrayidx.us.i, align 4, !tbaa !12,
... !llvm.access.group !16
%indvars.iv.next.us.2.i = or i64 %indvars.iv.next.us.3.i12, 3
%48 = mul nsw i64 %indvars.iv.next.us.2.i, %11
%49 = add nsw i64 %48, %15
%arrayidx11.us.3.i = getelementptr inbounds float, float* %1, i64 %49
%50 = load float, float* %arrayidx11.us.3.i, align 4, !tbaa !12
%51 = add nsw i64 %48, %indvars.iv.next70.us.i23
%arrayidx15.us.3.i = getelementptr inbounds float, float* %1, i64 %51
%52 = load float, float* %arrayidx15.us.3.i, align 4, !tbaa !12
%53 = tail call float @llvm.fmuladd.f32(float %50, float %52, float %47) #3
store float %53, float* %arrayidx.us.i, align 4, !tbaa !12,
... !llvm.access.group !16
%indvars.iv.next.us.3.i = add nuw nsw i64 %indvars.iv.next.us.3.i12, 4
%niter.nsub.3.i = add i64 %niter.nsub.3.i15, -4
%niter.ncmp.3.i = icmp eq i64 %niter.nsub.3.i, 0
br i1 %niter.ncmp.3.i, label %for.end.loopexit.us.unr-lcssa.loopexit.i,
... label %for.body7.us.i
```

```
for.end.loopexit.us.unr-lcssa.loopexit.i:
%.lcssa = phi float [ %53, %for.body7.us.i ]
%indvars.iv.next.us.3.i.lcssa = phi i64 [ %indvars.iv.next.us.3.i,
... %for.body7.us.i ]
store float %.lcssa, float* %26, align 4, !llvm.access.group !16
%54 = bitcast float %.lcssa to i32
br label %for.end.loopexit.us.unr-lcssa.i
```

```
for.body.preheader.i:
%17 = sub nsw i64 %11, %15
%18 = xor i64 %15, -1
%19 = add nsw i64 %18, %11
%xtrailer6.i = and i64 %17, 3
%icmp.mod7.i = icmp eq i64 %xtrailer6.i, 0
br i1 %icmp.mod7.i, label %for.body.prol.loopexit.i, label
... %for.body.prol.i.preheader
```

```
for.body.prol.i.preheader:
br label %for.body.prol.i
```

```
for.body.prol.i:
%prol.iter.sub.i6 = phi i64 [ %prol.iter.sub.i, %for.body.prol.i ], [
... %xtrailer6.i, %for.body.prol.i.preheader ]
%indvars.iv.next70.prol.i3 = phi i64 [ %indvars.iv.next70.prol.i,
... %for.body.prol.i ], [ %15, %for.body.prol.i.preheader ]
%20 = add nsw i64 %indvars.iv.next70.prol.i3, %16
%arrayidx.prol.i = getelementptr inbounds float, float* %0, i64 %20
store float 0.000000e+00, float* %arrayidx.prol.i, align 4, !tbaa !12,
... !llvm.access.group !16
%21 = mul nsw i64 %indvars.iv.next70.prol.i3, %11
%22 = add nsw i64 %21, %15
%arrayidx28.prol.i = getelementptr inbounds float, float* %0, i64 %22
%23 = bitcast float* %arrayidx28.prol.i to i32*
store i32 0, i32* %23, align 4, !tbaa !12, !llvm.access.group !16
%indvars.iv.next70.prol.i = add nsw i64 %indvars.iv.next70.prol.i3, 1
%prol.iter.sub.i = add nsw i64 %prol.iter.sub.i6, -1
%prol.iter.cmp.i = icmp eq i64 %prol.iter.sub.i, 0
br i1 %prol.iter.cmp.i, label %for.body.prol.loopexit.i.loopexit, label
... %for.body.prol.i, !llvm.loop !18
```

```
for.body.prol.loopexit.i.loopexit:
%indvars.iv.next70.prol.i.lcssa = phi i64 [ %indvars.iv.next70.prol.i,
... %for.body.prol.i ]
br label %for.body.prol.loopexit.i
```

```
for.body.prol.loopexit.i:
%24 = phi i64 [ %15, %for.body.preheader.i ], [
... %indvars.iv.next70.prol.i.lcssa, %for.body.prol.loopexit.i.loopexit ]
%25 = icmp ult i64 %19, 3
br i1 %25, label %if.end.i, label %for.body.i.preheader
```

```
for.body.i.preheader:
br label %for.body.i
```

```
for.body.i:
%indvars.iv.next70.3.i8 = phi i64 [ %indvars.iv.next70.3.i, %for.body.i ], [
... %24, %for.body.i.preheader ]
%69 = add nsw i64 %indvars.iv.next70.3.i8, %16
%arrayidx.i = getelementptr inbounds float, float* %0, i64 %69
store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !12,
... !llvm.access.group !16
%70 = mul nsw i64 %indvars.iv.next70.3.i8, %11
%71 = add nsw i64 %70, %15
%arrayidx28.i = getelementptr inbounds float, float* %0, i64 %71
%72 = bitcast float* %arrayidx28.i to i32*
store i32 0, i32* %72, align 4, !tbaa !12, !llvm.access.group !16
%indvars.iv.next70.i = add nsw i64 %indvars.iv.next70.3.i8, 1
%73 = add nsw i64 %indvars.iv.next70.i, %16
%arrayidx.1.i = getelementptr inbounds float, float* %0, i64 %73
store float 0.000000e+00, float* %arrayidx.1.i, align 4, !tbaa !12,
... !llvm.access.group !16
%74 = mul nsw i64 %indvars.iv.next70.i, %11
%75 = add nsw i64 %74, %15
%arrayidx28.1.i = getelementptr inbounds float, float* %0, i64 %75
%76 = bitcast float* %arrayidx28.1.i to i32*
store i32 0, i32* %76, align 4, !tbaa !12, !llvm.access.group !16
%indvars.iv.next70.2.i = add nsw i64 %indvars.iv.next70.3.i8, 2
%77 = add nsw i64 %indvars.iv.next70.1.i, %16
%arrayidx.2.i = getelementptr inbounds float, float* %0, i64 %77
store float 0.000000e+00, float* %arrayidx.2.i, align 4, !tbaa !12,
... !llvm.access.group !16
%78 = mul nsw i64 %indvars.iv.next70.2.i, %11
%79 = add nsw i64 %78, %15
%arrayidx28.2.i = getelementptr inbounds float, float* %0, i64 %79
%80 = bitcast float* %arrayidx28.2.i to i32*
store i32 0, i32* %80, align 4, !tbaa !12, !llvm.access.group !16
%indvars.iv.next70.1.i = add nsw i64 %indvars.iv.next70.3.i8, 3
%81 = add nsw i64 %indvars.iv.next70.2.i, %16
%arrayidx.3.i = getelementptr inbounds float, float* %0, i64 %81
store float 0.000000e+00, float* %arrayidx.3.i, align 4, !tbaa !12,
... !llvm.access.group !16
%82 = mul nsw i64 %indvars.iv.next70.2.i, %11
%83 = add nsw i64 %82, %15
%arrayidx28.3.i = getelementptr inbounds float, float* %0, i64 %83
%84 = bitcast float* %arrayidx28.3.i to i32*
store i32 0, i32* %84, align 4, !tbaa !12, !llvm.access.group !16
%indvars.iv.next70.3.i = add nsw i64 %indvars.iv.next70.3.i8, 4
%exitcond75.3.i = icmp eq i64 %indvars.iv.next70.3.i, %11
br i1 %exitcond75.3.i, label %if.end.i.loopexit34, label %for.body.i
```

```
if.end.i.loopexit34:
br label %if.end.i
```

```
if.end.i:
%85 = add nuw nsw i64 %local_id_x.0, 1
%exitcond = icmp eq i64 %85, 256
br i1 %exitcond, label %covar_kernel.exit, label %pregion_for_entry.entry.i,
... !llvm.loop !21
```

```
covar_kernel.exit:
call void @llvm.lifetime.end.p0i8(i64 1024, i8* nonnull %9)
ret void
```

CFG for 'pocl_kernel_covar_kernel' function