

```
%9:
%10 = sext i32 %3 to i64
%11 = icmp slt i64 %10, 256
%12 = select i1 %11, i64 %10, i64 256
%mul.i.i = shl i64 %6, 8
%cmp222.i = icmp sgt i32 %4, 0, !llvm.access.group !12
%wide.trip.count.i = zext i32 %4 to i64
%13 = icmp ugt i64 %12, 1
%umax = select i1 %13, i64 %12, i64 1
%min.iters.check = icmp ult i64 %umax, 8
br i1 %min.iters.check, label %preregion_for_entry.entry.i.preheader, label ... %vector.ph
```

T	F
---	---

```
vector.ph:
%n.vec = and i64 %umax, -8
%broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
%broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64> ... undef, <8 x i32> zeroinitializer
%broadcast.splatinsert10 = insertelement <8 x i64> undef, i64 %10, i32 0
%broadcast.splat11 = shufflevector <8 x i64> %broadcast.splatinsert10, <8 x ... i64> undef, <8 x i32> zeroinitializer
%broadcast.splatinsert12 = insertelement <8 x i64> undef, i64 ... %wide.trip.count.i, i32 0
%broadcast.splat13 = shufflevector <8 x i64> %broadcast.splatinsert12, <8 x ... i64> undef, <8 x i32> zeroinitializer
%broadcast.splatinsert18 = insertelement <8 x float> undef, float %2, i32 0
%broadcast.splat19 = shufflevector <8 x float> %broadcast.splatinsert18, <8 ... x float> undef, <8 x i32> zeroinitializer
br label %vector.body
```

```
vector.body:
%index = phi i64 [ 0, %vector.ph ], [ %index.next, %for.end.i16 ]
%vec.ind = phi <8 x i64> [ <i64 0, i64 1, i64 2, i64 3, i64 4, i64 5, i64 6, ... i64 7>, %vector.ph ], [ %vec.ind.next, %for.end.i16 ]
%14 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
%15 = shl <8 x i64> %14, <i64 32, i64 32, i64 32, i64 32, i64 32, i64 32, ... i64 32, i64 32>, !llvm.access.group !12
%16 = ashr exact <8 x i64> %15, <i64 32, i64 32, i64 32, i64 32, i64 32, i64 ... 32, i64 32, i64 32>, !llvm.access.group !12
%17 = getelementptr inbounds float, float* %0, <8 x i64> %16, ... !llvm.access.group !12
call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> zeroinitializer, <8 ... x float*> %17, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, ... i1 true, i1 true, i1 true>), !tbaa !14, !llvm.access.group !12
br i1 %cmp222.i, label %for.body.i8.preheader, label %for.end.i16
```

T	F
---	---

```
for.body.i8.preheader:
br label %for.body.i8
```

```
for.body.i8:
%vec.phi = phi <8 x i64> [ %22, %for.body.i8 ], [ zeroinitializer, ... %for.body.i8.preheader ]
%vec.phi9 = phi <8 x float> [ %21, %for.body.i8 ], [ zeroinitializer, ... %for.body.i8.preheader ]
%18 = mul nsw <8 x i64> %vec.phi, %broadcast.splat11, !llvm.access.group !12
%19 = add nsw <8 x i64> %18, %16, !llvm.access.group !12
%20 = getelementptr inbounds float, float* %1, <8 x i64> %19, ... !llvm.access.group !12
%wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8 ... x float*> %20, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, ... i1 true, i1 true, i1 true>, <8 x float> undef), !tbaa !14, !llvm.access.group ... !12
%21 = fadd <8 x float> %vec.phi9, %wide.masked.gather, !llvm.access.group !12
call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %21, <8 x float*> ... %17, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, ... i1 true, i1 true>), !tbaa !14, !llvm.access.group !12
%22 = add nuw nsw <8 x i64> %vec.phi, <i64 1, i64 1, i64 1, i64 1, i64 1, ... i64 1, i64 1, i64 1>, !llvm.access.group !12
%23 = icmp eq <8 x i64> %22, %broadcast.splat13, !llvm.access.group !12
%24 = extractelement <8 x i1> %23, i32 0
br i1 %24, label %for.end.i16.loopexit, label %for.body.i8
```

T	F
---	---

```
for.end.i16.loopexit:
%.lcssa = phi <8 x float> [ %21, %for.body.i8 ]
br label %for.end.i16
```

```
for.end.i16:
%vec.phi17 = phi <8 x float> [ zeroinitializer, %vector.body ], [ %.lcssa, ... %for.end.i16.loopexit ]
%25 = fdiv <8 x float> %vec.phi17, %broadcast.splat19, !fpmath !18, ... !llvm.access.group !12
call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %25, <8 x float*> ... %17, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, ... i1 true, i1 true>), !tbaa !14, !llvm.access.group !12
%index.next = add i64 %index, 8
%vec.ind.next = add <8 x i64> %vec.ind, <i64 8, i64 8, i64 8, i64 8, i64 8, ... i64 8, i64 8, i64 8>
%26 = icmp eq i64 %index.next, %n.vec
br i1 %26, label %middle.block, label %vector.body, !llvm.loop !19
```

T	F
---	---

```
middle.block:
%cmp.n = icmp eq i64 %umax, %n.vec
br i1 %cmp.n, label %mean_kernel.exit, label ... %preregion_for_entry.entry.i.preheader
```

T	F
---	---

```
preregion_for_entry.entry.i.preheader:
%_local_id_x.0.ph = phi i64 [ 0, %9 ], [ %n.vec, %middle.block ]
br label %preregion_for_entry.entry.i
```

```
preregion_for_entry.entry.i:
%_local_id_x.0 = phi i64 [ %31, %for.end.i ], [ %_local_id_x.0.ph, ... %preregion_for_entry.entry.i.preheader ]
%add1.i.i = add i64 %_local_id_x.0, %mul.i.i, !llvm.access.group !12
%sext.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
%idxprom.i = ashr exact i64 %sext.i, 32, !llvm.access.group !12
%arrayidx.i = getelementptr inbounds float, float* %0, i64 %idxprom.i, ... !llvm.access.group !12
store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !14, ... !llvm.access.group !12
br i1 %cmp222.i, label %for.body.i.preheader, label %for.end.i, ... !llvm.access.group !12
```

T	F
---	---

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

```
for.body.i:
%indvars.iv.next.i4 = phi i64 [ %indvars.iv.next.i, %for.body.i ], [ 0, ... %for.body.i.preheader ]
%add8.i1 = phi float [ %add8.i, %for.body.i ], [ 0.000000e+00, ... %for.body.i.preheader ]
%27 = mul nsw i64 %indvars.iv.next.i4, %10, !llvm.access.group !12
%28 = add nsw i64 %27, %idxprom.i, !llvm.access.group !12
%arrayidx5.i = getelementptr inbounds float, float* %1, i64 %28, ... !llvm.access.group !12
%29 = load float, float* %arrayidx5.i, align 4, !tbaa !14, ... !llvm.access.group !12
%add8.i = fadd float %add8.i1, %29, !llvm.access.group !12
store float %add8.i, float* %arrayidx.i, align 4, !tbaa !14, ... !llvm.access.group !12
%indvars.iv.next.i = add nuw nsw i64 %indvars.iv.next.i4, 1, ... !llvm.access.group !12
%exitcond.not.i = icmp eq i64 %indvars.iv.next.i, %wide.trip.count.i, ... !llvm.access.group !12
br i1 %exitcond.not.i, label %for.end.i.loopexit, label %for.body.i, ... !llvm.loop !22, !llvm.access.group !12
```

T	F
---	---

```
for.end.i.loopexit:
%add8.i.lcssa = phi float [ %add8.i, %for.body.i ]
br label %for.end.i
```

```
for.end.i:
%30 = phi float [ 0.000000e+00, %preregion_for_entry.entry.i ], [ ... %add8.i.lcssa, %for.end.i.loopexit ]
%div.i = fdiv float %30, %2, !fpmath !18, !llvm.access.group !12
store float %div.i, float* %arrayidx.i, align 4, !tbaa !14, ... !llvm.access.group !12
%31 = add nuw i64 %_local_id_x.0, 1
%exitcond.not = icmp eq i64 %31, %umax
br i1 %exitcond.not, label %mean_kernel.exit.loopexit, label ... %preregion_for_entry.entry.i, !llvm.loop !24
```

T	F
---	---

```
mean_kernel.exit.loopexit:
br label %mean_kernel.exit
```

```
mean_kernel.exit:
ret void
```

CFG for ' _pocl_kernel_mean_kernel' function