

```
%10:
%11 = sext i32 %5 to i64
%12 = icmp slt i64 %11, 32
%13 = select i1 %12, i64 %11, i64 32
%14 = icmp slt i64 %11, 8
%15 = select i1 %14, i64 %11, i64 8
%mul3.i.i = shl i64 %7, 5
%mul3.i.i = shl i64 %8, 3
%16 = icmp ugt i64 %13, 1
%umax = select i1 %16, i64 %13, i64 1
%17 = icmp ugt i64 %15, 1
%umax1 = select i1 %17, i64 %15, i64 1
%min.itors.check = icmp ult i64 %umax1, 8
br i1 %min.itors.check, label
... %pregon_for_entry.region_for_init.i.preheader, label %vector.ph
```

T	F
---	---

```
vector.ph:
%n.vec = and i64 %umax1, -8
%broadcast.splatinsert = insertelement <8 x i64> undef, i64 %mul3.i.i, i32 0
%broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
... undef, <8 x i32> zeroinitializer
%broadcast.splatinsert3 = insertelement <8 x i32> undef, i32 %5, i32 0
%broadcast.splat4 = shufflevector <8 x i32> %broadcast.splatinsert3, <8 x
... i32> undef, <8 x i32> zeroinitializer
%broadcast.splatinsert6 = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
%broadcast.splat7 = shufflevector <8 x i64> %broadcast.splatinsert6, <8 x
... i64> undef, <8 x i32> zeroinitializer
%broadcast.splatinsert12 = insertelement <8 x i64> undef, i64 %umax, i32 0
%broadcast.splat13 = shufflevector <8 x i64> %broadcast.splatinsert12, <8 x
... i64> undef, <8 x i32> zeroinitializer
br label %vector.body
```

```
vector.body:
%index = phi i64 [ 0, %vector.ph ], [ %index.next, %pregon_for_end.i14 ]
%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, %pregon_for_end.i14 ]
%18 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
%19 = trunc <8 x i64> %18 to <8 x i32>, !llvm.access.group !12
%20 = shl <8 x i64> %18, <i64 32, i64 32, i64 32, i64 32, i64 32, i64 32,
... i64 32, i64 32>, !llvm.access.group !12
%21 = ashr exact <8 x i64> %20, <i64 32, i64 32, i64 32, i64 32, i64 32, i64
... 32, i64 32, i64 32>, !llvm.access.group !12
%22 = getelementptr inbounds float, float* %3, <8 x i64> %21,
... !llvm.access.group !12
%23 = getelementptr inbounds float, float* %4, <8 x i64> %21,
... !llvm.access.group !12
%24 = mul nsw <8 x i32> %broadcast.splat4, %19, !llvm.access.group !12
br label %pregon_for_entry.entry.i5
```

```
pregon_for_entry.entry.i5:
%vec.phi = phi <8 x i64> [ zeroinitializer, %vector.body ], [ %38,
... %pregon_for_entry.entry.i5 ]
%25 = add <8 x i64> %vec.phi, %broadcast.splat7, !llvm.access.group !12
%26 = trunc <8 x i64> %25 to <8 x i32>, !llvm.access.group !12
%wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
... x float*> %22, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true,
... i1 true, i1 true, i1 true>, <8 x float> undef), !tbaa !15, !llvm.access.group
... !12
%27 = shl <8 x i64> %25, <i64 32, i64 32, i64 32, i64 32, i64 32, i64 32,
... i64 32, i64 32>, !llvm.access.group !12
%28 = ashr exact <8 x i64> %27, <i64 32, i64 32, i64 32, i64 32, i64 32, i64
... 32, i64 32, i64 32>, !llvm.access.group !12
%29 = getelementptr inbounds float, float* %1, <8 x i64> %28,
... !llvm.access.group !12
%wide.masked.gather8 = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
... x float*> %29, 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 !15, !llvm.access.group
... !12
%wide.masked.gather9 = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
... x float*> %23, 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 !15, !llvm.access.group
... !12
%30 = getelementptr inbounds float, float* %2, <8 x i64> %28,
... !llvm.access.group !12
%wide.masked.gather10 = call <8 x float>
... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %30, i32 4, <8 x i1> <i1 true,
... i1 true, i1 true, i1 true, i1 true, i1 true, i1 true>, <8 x float>
... undef), !tbaa !15, !llvm.access.group !12
%31 = fmul <8 x float> %wide.masked.gather9, %wide.masked.gather10,
... !llvm.access.group !12
%32 = fmul <8 x float> %wide.masked.gather, %wide.masked.gather8,
... !llvm.access.group !12
%33 = fadd <8 x float> %32, %31, !llvm.access.group !12
%34 = add nsw <8 x i32> %24, %26, !llvm.access.group !12
%35 = sext <8 x i32> %34 to <8 x i64>, !llvm.access.group !12
%36 = getelementptr inbounds float, float* %0, <8 x i64> %35,
... !llvm.access.group !12
%wide.masked.gather11 = call <8 x float>
... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %36, i32 4, <8 x i1> <i1 true,
... i1 true, i1 true, i1 true, i1 true, i1 true, i1 true>, <8 x float>
... undef), !tbaa !15, !llvm.access.group !12
%37 = fadd <8 x float> %wide.masked.gather11, %33, !llvm.access.group !12
call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %37, <8 x float*>
... %36, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1 true,
... i1 true, i1 true>), !tbaa !15, !llvm.access.group !12
%38 = add nuw <8 x i64> %vec.phi, <i64 1, i64 1, i64 1, i64 1, i64 1, i64 1,
... i64 1, i64 1>
%39 = icmp eq <8 x i64> %38, %broadcast.splat13
%40 = extractelement <8 x i1> %39, i32 0
br i1 %40, label %pregon_for_end.i14, label %pregon_for_entry.entry.i5
```

T	F
---	---

```
pregon_for_end.i14:
%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>
%41 = icmp eq i64 %index.next, %n.vec
br i1 %41, label %middle.block, label %vector.body, !llvm.loop !19
```

T	F
---	---

```
middle.block:
%cmp.n = icmp eq i64 %umax1, %n.vec
br i1 %cmp.n, label %gemver_kernel1.exit, label
... %pregon_for_entry.region_for_init.i.preheader
```

T	F
---	---

```
pregon_for_entry.region_for_init.i.preheader:
%_local_id_y.0.ph = phi i64 [ 0, %10 ], [ %n.vec, %middle.block ]
br label %pregon_for_entry.region_for_init.i
```

```
pregon_for_entry.region_for_init.i:
%_local_id_y.0 = phi i64 [%50, %pregon_for_end.i ], [ %_local_id_y.0.ph,
... %pregon_for_entry.region_for_init.i.preheader ]
%add6.i.i = add i64 %_local_id_y.0, %mul3.i.i, !llvm.access.group !12
%conv2.i = trunc i64 %add6.i.i to i32, !llvm.access.group !12
%sext.i = shl i64 %add6.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* %3, i64 %idxprom.i,
... !llvm.access.group !12
%arrayidx9.i = getelementptr inbounds float, float* %4, i64 %idxprom.i,
... !llvm.access.group !12
%mul.i = mul nsw i32 %conv2.i, %5, !llvm.access.group !12
br label %pregon_for_entry.entry.i
```

```
pregon_for_entry.entry.i:
%_local_id_x.0 = phi i64 [ 0, %pregon_for_entry.region_for_init.i ], [
... %49, %pregon_for_entry.entry.i ]
%add1.i.i = add i64 %_local_id_x.0, %mul.i.i, !llvm.access.group !12
%conv.i = trunc i64 %add1.i.i to i32, !llvm.access.group !12
%42 = load float, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
... !12
%sext26.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
%idxprom6.i = ashr exact i64 %sext26.i, 32, !llvm.access.group !12
%arrayidx7.i = getelementptr inbounds float, float* %1, i64 %idxprom6.i,
... !llvm.access.group !12
%43 = load float, float* %arrayidx7.i, align 4, !tbaa !15,
... !llvm.access.group !12
%44 = load float, float* %arrayidx9.i, align 4, !tbaa !15,
... !llvm.access.group !12
%arrayidx11.i = getelementptr inbounds float, float* %2, i64 %idxprom6.i,
... !llvm.access.group !12
%45 = load float, float* %arrayidx11.i, align 4, !tbaa !15,
... !llvm.access.group !12
%mul12.i = fmul float %44, %45, !llvm.access.group !12
%46 = fmul float %42, %43, !llvm.access.group !12
%47 = fadd float %46, %mul12.i, !llvm.access.group !12
%add.i = add nsw i32 %mul.i, %conv.i, !llvm.access.group !12
%idxprom13.i = sext i32 %add.i to i64, !llvm.access.group !12
%arrayidx14.i = getelementptr inbounds float, float* %0, i64 %idxprom13.i,
... !llvm.access.group !12
%48 = load float, float* %arrayidx14.i, align 4, !tbaa !15,
... !llvm.access.group !12
%add15.i = fadd float %48, %47, !llvm.access.group !12
store float %add15.i, float* %arrayidx14.i, align 4, !tbaa !15,
... !llvm.access.group !12
%49 = add nuw i64 %_local_id_x.0, 1
%exitcond.not = icmp eq i64 %49, %umax
br i1 %exitcond.not, label %pregon_for_end.i, label
... %pregon_for_entry.entry.i, !llvm.loop !22
```

T	F
---	---

```
pregon_for_end.i:
%50 = add nuw i64 %_local_id_y.0, 1
%exitcond2.not = icmp eq i64 %50, %umax1
br i1 %exitcond2.not, label %gemver_kernel1.exit.loopexit, label
... %pregon_for_entry.region_for_init.i, !llvm.loop !25
```

T	F
---	---

```
gemver_kernel1.exit.loopexit:
br label %gemver_kernel1.exit
```

```
gemver_kernel1.exit:
ret void
```

CFG for ' _pocl_kernel_gemver_kernel1' function