

```
%10:
%mul.i.i = shl i64 %7, 8
%cmp228.i = icmp sgt i32 %5, 0
%11 = zext i32 %5 to i64
br i1 %cmp228.i, label %region_for_entry.entry.i.us.preheader, label
... %vector.memcheck
```

```
preregion_for_entry.entry.i.us.preheader:
br label %region_for_entry.entry.i.us
```

```
vector.memcheck:
%12 = trunc i64 %7 to i32
%13 = shl i32 %12, 8
%14 = sext i32 %13 to i64
%scevgep = getelementptr float, float* %1, i64 %14
%15 = add nsw i64 %14, 256
%scevgep9 = getelementptr float, float* %1, i64 %15
%scevgep11 = getelementptr float, float* %3, i64 %14
%scevgep13 = getelementptr float, float* %3, i64 %15
%bound0 = icmp ult float* %scevgep, %scevgep13
%bound1 = icmp ult float* %scevgep11, %scevgep9
%found.conflict = and i1 %bound0, %bound1
br i1 %found.conflict, label %preregion_for_entry.entry.i.preheader, label
... %vector.ph
```

```
preregion_for_entry.entry.i.preheader:
br label %preregion_for_entry.entry.i
```

```
vector.ph:
%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.splatinsert15 = insertelement <8 x i32> undef, i32 %5, i32 0
%broadcast.splat16 = shufflevector <8 x i32> %broadcast.splatinsert15, <8 x
... i32> undef, <8 x i32> zeroinitializer
br label %vector.body
```

```
preregion_for_entry.entry.i.us:
% local_id x.0.us = phi i64 [ %58, %if.end_r_exit.i.us.1 ], [ 0,
... %preregion_for_entry.entry.i.us.preheader ]
%add1.i.i.us = add nuw nsw i64 % local_id x.0.us, %mul.i.i
%conv.i.us = trunc i64 %add1.i.i.us to i32
%cmp.i.us = icmp slt i32 %conv.i.us, %5
br i1 %cmp.i.us, label %for.cond.preheader.i.us, label %if.end_r_exit.i.us
```

```
for.cond.preheader.i.us:
%sext27.i.us = shl i64 %add1.i.i.us, 32
%idxprom8.i.us = ashr exact i64 %sext27.i.us, 32
%arrayidx9.i.us = getelementptr inbounds float, float* %1, i64 %idxprom8.i.us
%pre.i.us = load float, float* %arrayidx9.i.us, align 4, !tbaa !12
br label %for.body.i.us
```

```
preregion_for_entry.entry.i:
% local_id x.0 = phi i64 [ %64, %if.end_r_exit.i.3 ], [ 0,
... %preregion_for_entry.entry.i.preheader ]
%add1.i.i = add nuw nsw i64 % local_id x.0, %mul.i.i
%conv.i = trunc i64 %add1.i.i to i32
%cmp.i = icmp slt i32 %conv.i, %5
br i1 %cmp.i, label %for.cond.preheader.i, label %if.end_r_exit.i
```

```
vector.body:
%index = phi i64 [ 0, %vector.ph ], [ %index.next.1, %vector.body ]
%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.1, %vector.body ]
%16 = add nuw nsw <8 x i64> %vec.ind, %broadcast.splat
%17 = trunc <8 x i64> %16 to <8 x i32>
%18 = icmp sgt <8 x i32> %broadcast.splat16, %17
%19 = extractelement <8 x i64> %16, i32 0
%20 = shl i64 %19, 32
%21 = ashr exact i64 %20, 32
%22 = getelementptr inbounds float, float* %1, i64 %21
%23 = bitcast float* %22 to <8 x float>*
%wide.masked.load = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %23, i32 4, <8 x i1> %18, <8 x float> undef), !tbaa !12, !alias.scope
... !16, !noalias !19
%24 = getelementptr inbounds float, float* %3, i64 %21
%25 = bitcast float* %24 to <8 x float>*
%wide.masked.load17 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %25, i32 4, <8 x i1> %18, <8 x float> undef), !tbaa !12, !alias.scope
... !19
%26 = fadd <8 x float> %wide.masked.load, %wide.masked.load17
%27 = bitcast float* %22 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %26, <8 x float>*
... %27, i32 4, <8 x i1> %18), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !21
%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>
%28 = add nuw nsw <8 x i64> %vec.ind.next, %broadcast.splat
%29 = trunc <8 x i64> %28 to <8 x i32>
%30 = icmp sgt <8 x i32> %broadcast.splat16, %29
%31 = extractelement <8 x i64> %28, i32 0
%32 = shl i64 %31, 32
%33 = ashr exact i64 %32, 32
%34 = getelementptr inbounds float, float* %1, i64 %33
%35 = bitcast float* %34 to <8 x float>*
%wide.masked.load.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8 x
... float>* %35, i32 4, <8 x i1> %30, <8 x float> undef), !tbaa !12, !alias.scope
... !16, !noalias !19
%36 = getelementptr inbounds float, float* %3, i64 %33
%37 = bitcast float* %36 to <8 x float>*
%wide.masked.load17.1 = call <8 x float> @llvm.masked.load.v8f32.p0v8f32(<8
... x float>* %37, i32 4, <8 x i1> %30, <8 x float> undef), !tbaa !12,
... !alias.scope !19
%38 = fadd <8 x float> %wide.masked.load.1, %wide.masked.load17.1
%39 = bitcast float* %34 to <8 x float>*
call void @llvm.masked.store.v8f32.p0v8f32(<8 x float> %38, <8 x float>*
... %39, i32 4, <8 x i1> %30), !tbaa !12, !alias.scope !16, !noalias !19,
... !llvm.access.group !21
%index.next.1 = add nuw nsw i64 %index, 16
%vec.ind.next.1 = add <8 x i64> %vec.ind, <i64 16, i64 16, i64 16, i64 16,
... i64 16, i64 16, i64 16>
%40 = icmp eq i64 %index.next.1, 256
br i1 %40, label %gemver_kernel2.exit.loopexit22, label %vector.body,
... !llvm.loop !23
```

```
for.body.i.us:
%indvars.iv.next.i3.us = phi i64 [ %indvars.iv.next.i.us, %for.body.i.us ],
... [ 0, %for.cond.preheader.i.us ]
%41 = phi float [ %46, %for.body.i.us ], [ %pre.i.us,
... %for.cond.preheader.i.us ]
%42 = mul nuw nsw i64 %indvars.iv.next.i3.us, %11
%43 = add nsw i64 %42, %idxprom8.i.us
%arrayidx.i.us = getelementptr inbounds float, float* %0, i64 %43
%44 = load float, float* %arrayidx.i.us, align 4, !tbaa !12
%mul4.i.us = fmul float %44, %4
%arrayidx6.i.us = getelementptr inbounds float, float* %2, i64
... %indvars.iv.next.i3.us
%45 = load float, float* %arrayidx6.i.us, align 4, !tbaa !12
%46 = tail call float @llvm.fmuladd.f32(float %mul4.i.us, float %45, float
... %41) #5
store float %46, float* %arrayidx9.i.us, align 4, !tbaa !12,
... !llvm.access.group !21
%indvars.iv.next.i.us = add nuw nsw i64 %indvars.iv.next.i3.us, 1
%exitcond.not.i.us = icmp eq i64 %indvars.iv.next.i.us, %11
br i1 %exitcond.not.i.us, label %for.end.loopexit.i.us, label
... %for.body.i.us, !llvm.loop !26
```

```
for.end.loopexit.i.us:
%lcssa = phi float [ %46, %for.body.i.us ]
%arrayidx11.i.us = getelementptr inbounds float, float* %3, i64
... %idxprom8.i.us
%47 = load float, float* %arrayidx11.i.us, align 4, !tbaa !12
%add14.i.us = fadd float %lcssa, %47
store float %add14.i.us, float* %arrayidx9.i.us, align 4, !tbaa !12,
... !llvm.access.group !21
br label %if.end_r_exit.i.us
```

```
if.end_r_exit.i.us:
%48 = or i64 % local_id x.0.us, 1
%add1.i.i.us.1 = add nuw nsw i64 %48, %mul.i.i
%conv.i.us.1 = trunc i64 %add1.i.i.us.1 to i32
%cmp.i.us.1 = icmp slt i32 %conv.i.us.1, %5
br i1 %cmp.i.us.1, label %for.cond.preheader.i.us.1, label
... %if.end_r_exit.i.us.1
```

```
for.cond.preheader.i.us.1:
%sext27.i.us.1 = shl i64 %add1.i.i.us.1, 32
%idxprom8.i.us.1 = ashr exact i64 %sext27.i.us.1, 32
%arrayidx9.i.us.1 = getelementptr inbounds float, float* %1, i64
... %idxprom8.i.us.1
%pre.i.us.1 = load float, float* %arrayidx9.i.us.1, align 4, !tbaa !12
br label %for.body.i.us.1
```

```
for.body.i.us.1:
%indvars.iv.next.i3.us.1 = phi i64 [ %indvars.iv.next.i.us.1,
... %for.body.i.us.1 ], [ 0, %for.cond.preheader.i.us.1 ]
%51 = phi float [ %56, %for.body.i.us.1 ], [ %pre.i.us.1,
... %for.cond.preheader.i.us.1 ]
%52 = mul nuw nsw i64 %indvars.iv.next.i3.us.1, %11
%53 = add nsw i64 %52, %idxprom8.i.us.1
%arrayidx.i.us.1 = getelementptr inbounds float, float* %0, i64 %53
%54 = load float, float* %arrayidx.i.us.1, align 4, !tbaa !12
%mul4.i.us.1 = fmul float %54, %4
%arrayidx6.i.us.1 = getelementptr inbounds float, float* %2, i64
... %indvars.iv.next.i3.us.1
%55 = load float, float* %arrayidx6.i.us.1, align 4, !tbaa !12
%56 = tail call float @llvm.fmuladd.f32(float %mul4.i.us.1, float %55, float
... %51) #5
store float %56, float* %arrayidx9.i.us.1, align 4, !tbaa !12,
... !llvm.access.group !21
%indvars.iv.next.i.us.1 = add nuw nsw i64 %indvars.iv.next.i3.us.1, 1
%exitcond.not.i.us.1 = icmp eq i64 %indvars.iv.next.i.us.1, %11
br i1 %exitcond.not.i.us.1, label %for.end.loopexit.i.us.1, label
... %for.body.i.us.1, !llvm.loop !26
```

```
for.end.loopexit.i.us.1:
%lcssa23 = phi float [ %56, %for.body.i.us.1 ]
%arrayidx11.i.us.1 = getelementptr inbounds float, float* %3, i64
... %idxprom8.i.us.1
%57 = load float, float* %arrayidx11.i.us.1, align 4, !tbaa !12
%add14.i.us.1 = fadd float %lcssa23, %57
store float %add14.i.us.1, float* %arrayidx9.i.us.1, align 4, !tbaa !12,
... !llvm.access.group !21
br label %if.end_r_exit.i.us.1
```

```
if.end_r_exit.i.us.1:
%58 = add nuw nsw i64 % local_id x.0.us, 2
%exitcond.not.1 = icmp eq i64 %58, 256
br i1 %exitcond.not.1, label %gemver_kernel2.exit.loopexit, label
... %preregion_for_entry.entry.i.us, !llvm.loop !28
```

```
for.cond.preheader.i:
%sext27.i = shl i64 %add1.i.i, 32
%idxprom8.i = ashr exact i64 %sext27.i, 32
%arrayidx9.i = getelementptr inbounds float, float* %1, i64 %idxprom8.i
%pre1.i = load float, float* %arrayidx9.i, align 4, !tbaa !12
%arrayidx11.i = getelementptr inbounds float, float* %3, i64 %idxprom8.i
%49 = load float, float* %arrayidx11.i, align 4, !tbaa !12
%add14.i = fadd float %pre1.i, %49
store float %add14.i, float* %arrayidx9.i, align 4, !tbaa !12,
... !llvm.access.group !21
br label %if.end_r_exit.i
```

```
if.end_r_exit.i:
%50 = or i64 % local_id x.0, 1
%add1.i.i.1 = add nuw nsw i64 %50, %mul.i.i
%conv.i.1 = trunc i64 %add1.i.i.1 to i32
%cmp.i.1 = icmp slt i32 %conv.i.1, %5
br i1 %cmp.i.1, label %for.cond.preheader.i.1, label %if.end_r_exit.i.1
```

```
for.cond.preheader.i.1:
%sext27.i.1 = shl i64 %add1.i.i.1, 32
%idxprom8.i.1 = ashr exact i64 %sext27.i.1, 32
%arrayidx9.i.1 = getelementptr inbounds float, float* %1, i64 %idxprom8.i.1
%pre1.i.1 = load float, float* %arrayidx9.i.1, align 4, !tbaa !12
%arrayidx11.i.1 = getelementptr inbounds float, float* %3, i64 %idxprom8.i.1
%59 = load float, float* %arrayidx11.i.1, align 4, !tbaa !12
%add14.i.1 = fadd float %pre1.i.1, %59
store float %add14.i.1, float* %arrayidx9.i.1, align 4, !tbaa !12,
... !llvm.access.group !21
br label %if.end_r_exit.i.1
```

```
if.end_r_exit.i.1:
%60 = or i64 % local_id x.0, 2
%add1.i.i.2 = add nuw nsw i64 %60, %mul.i.i
%conv.i.2 = trunc i64 %add1.i.i.2 to i32
%cmp.i.2 = icmp slt i32 %conv.i.2, %5
br i1 %cmp.i.2, label %for.cond.preheader.i.2, label %if.end_r_exit.i.2
```

```
for.cond.preheader.i.2:
%sext27.i.2 = shl i64 %add1.i.i.2, 32
%idxprom8.i.2 = ashr exact i64 %sext27.i.2, 32
%arrayidx9.i.2 = getelementptr inbounds float, float* %1, i64 %idxprom8.i.2
%pre1.i.2 = load float, float* %arrayidx9.i.2, align 4, !tbaa !12
%arrayidx11.i.2 = getelementptr inbounds float, float* %3, i64 %idxprom8.i.2
%61 = load float, float* %arrayidx11.i.2, align 4, !tbaa !12
%add14.i.2 = fadd float %pre1.i.2, %61
store float %add14.i.2, float* %arrayidx9.i.2, align 4, !tbaa !12,
... !llvm.access.group !21
br label %if.end_r_exit.i.2
```

```
if.end_r_exit.i.2:
%62 = or i64 % local_id x.0, 3
%add1.i.i.3 = add nuw nsw i64 %62, %mul.i.i
%conv.i.3 = trunc i64 %add1.i.i.3 to i32
%cmp.i.3 = icmp slt i32 %conv.i.3, %5
br i1 %cmp.i.3, label %for.cond.preheader.i.3, label %if.end_r_exit.i.3
```

```
for.cond.preheader.i.3:
%sext27.i.3 = shl i64 %add1.i.i.3, 32
%idxprom8.i.3 = ashr exact i64 %sext27.i.3, 32
%arrayidx9.i.3 = getelementptr inbounds float, float* %1, i64 %idxprom8.i.3
%pre1.i.3 = load float, float* %arrayidx9.i.3, align 4, !tbaa !12
%63 = load float, float* %arrayidx11.i.3, align 4, !tbaa !12
%add14.i.3 = fadd float %pre1.i.3, %63
store float %add14.i.3, float* %arrayidx9.i.3, align 4, !tbaa !12,
... !llvm.access.group !21
br label %if.end_r_exit.i.3
```

```
if.end_r_exit.i.3:
%64 = add nuw nsw i64 % local_id x.0, 4
%exitcond5.not.3 = icmp eq i64 %64, 256
br i1 %exitcond5.not.3, label %gemver_kernel2.exit.loopexit21, label
... %preregion_for_entry.entry.i, !llvm.loop !29
```

```
gemver_kernel2.exit.loopexit21:
br label %gemver_kernel2.exit
```

```
gemver_kernel2.exit.loopexit22:
br label %gemver_kernel2.exit
```

gemver_kernel2.exit:
ret void

CFG for 'pool_kernel_gemver_kernel2' function