```
%12:
                                                                                                                                                                                                              %13 = \text{sext i} 32 \%6 \text{ to i} 64
                                                                                                                                                                                                               %14 = icmp slt i64 %13, 32
                                                                                                                                                                                                               %15 = select i1 %14, i64 %13, i64 32
                                                                                                                                                                                                               %16 = \text{sext i} 32 \% 5 \text{ to i} 64
                                                                                                                                                                                                              %17 = icmp slt i64 %16, 8
                                                                                                                                                                                                               %18 = select i1 %17, i64 %16, i64 8
                                                                                                                                                                                                               %mul.i.i = shl i64 %9, 5
                                                                                                                                                                                                               %mul3.i.i = shl i64 %10, 3
                                                                                                                                                                                                               %cmp740.i = icmp sgt i32 %7, 0, !llvm.access.group !12
                                                                                                                                                                                                               %wide.trip.count.i = zext i32 %7 to i64
                                                                                                                                                                                                              %19 = icmp ugt i64 \%15, 1
                                                                                                                                                                                                              %umax = select i1 %19, i64 %15, i64 1
                                                                                                                                                                                                              %20 = icmp ugt i64 \%18, 1
                                                                                                                                                                                                              %umax3 = select i1 %20, i64 %18, i64 1
                                                                                                                                                                                                               %min.iters.check = icmp ult i64 %umax3, 8
                                                                                                                                                                                                               br i1 %min.iters.check, label
                                                                                                                                                                                                              ... %pregion for entry.pregion for init.i.preheader, label %vector.ph
                                                                                                                                                                                                                                                                   vector.ph:
                                                                                                                                                                                                                                                                     %n.vec = and i64 %umax3, -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.splatinsert5 = insertelement <8 x i32> undef, i32 %6, i32 0
                                                                                                                                                                                                                                                                     %broadcast.splat6 = shufflevector <8 x i32> %broadcast.splatinsert5, <8 x
                                                                                                                                                                                                                                                                      .. i32> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                    %broadcast.splatinsert7 = insertelement <8 x i32> undef, i32 %7, i32 0
                                                                                                                                                                                                                                                                     %broadcast.splat8 = shufflevector <8 x i32> %broadcast.splatinsert7, <8 x
                                                                                                                                                                                                                                                                     .. i32> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                     %broadcast.splatinsert10 = insertelement <8 x i64> undef, i64 %mul.i.i, i32 0
                                                                                                                                                                                                                                                                     %broadcast.splat11 = shufflevector <8 x i64> %broadcast.splatinsert10, <8 x
                                                                                                                                                                                                                                                                     .. i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                     %broadcast.splatinsert12 = insertelement <8 x float> undef, float %4, i32 0
                                                                                                                                                                                                                                                                     %broadcast.splat13 = shufflevector <8 x float> %broadcast.splatinsert12, <8
                                                                                                                                                                                                                                                                     .. x float> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                     %broadcast.splatinsert21 = insertelement <8 x float> undef, float %3, i32 0
                                                                                                                                                                                                                                                                     %broadcast.splat22 = shufflevector <8 x float> %broadcast.splatinsert21, <8
                                                                                                                                                                                                                                                                      .. x float> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                     %broadcast.splatinsert23 = insertelement <8 x i64> undef, i64 %13, i32 0
                                                                                                                                                                                                                                                                     %broadcast.splat24 = shufflevector <8 x i64> %broadcast.splatinsert23, <8 x
                                                                                                                                                                                                                                                                     .. i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                     %broadcast.splatinsert26 = insertelement <8 x i64> undef, i64
                                                                                                                                                                                                                                                                     .. %wide.trip.count.i, i32 0
                                                                                                                                                                                                                                                                    %broadcast.splat27 = shufflevector <8 x i64> %broadcast.splatinsert26, <8 x
                                                                                                                                                                                                                                                                     .. i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                     %broadcast.splatinsert30 = insertelement <8 x i64> undef, i64 %umax, i32 0
                                                                                                                                                                                                                                                                     %broadcast.splat31 = shufflevector <8 x i64> %broadcast.splatinsert30, <8 x
                                                                                                                                                                                                                                                                     .. i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                                                                                                                     br label %vector.body
                                                                                                                                                                                                                                                                                              vector.body:
                                                                                                                                                                                                                                                                                                 %index = phi i64 [ 0, %vector.ph ], [ %index.next, %pregion_for_end.i32 ] %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, %pregion_for_end.i32 ]
                                                                                                                                                                                                                                                                                                 %21 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12 %22 = trunc <8 x i64> %21 to <8 x i32>, !llvm.access.group !12 %23 = mul nsw <8 x i32> %broadcast.splat6, %22, !llvm.access.group !12 %24 = mul nsw <8 x i32> %broadcast.splat6, %22
                                                                                                                                                                                                                                                                                                  %25 = \text{sext} < 8 \times i32 > \%24 \text{ to } < 8 \times i64 > 3 \times 
                                                                                                                                                                                                                                                                                                  br label %pregion for entry.entry.i9
                                                                                                                                                                                                                                                                                         pregion for entry.entry.i9:
                                                                                                                                                                                                                                                                                          %vec.phi = phi <8 x i64> [ zeroinitializer, %vector.body ], [ %44,
                                                                                                                                                                                                                                                                                          ... %if.end.r exit.i29 ]
                                                                                                                                                                                                                                                                                          %26 = add <8 x i64> %vec.phi, %broadcast.splat11, !llvm.access.group !12 %27 = trunc <8 x i64> %26 to <8 x i32>, !llvm.access.group !12
                                                                                                                                                                                                                                                                                          %28 = add nsw <8 x i32> %23, %27, !llvm.access.group !12
%29 = sext <8 x i32> %28 to <8 x i64>, !llvm.access.group !12
                                                                                                                                                                                                                                                                                          \%30 = \text{getelementptr inbounds float, float* } \%2, <8 \times i64 > \%29,
                                                                                                                                                                                                                                                                                          ...!llvm.access.group!12
                                                                                                                                                                                                                                                                                          %wide.masked.gather = 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, i1 true, <8 x float> undef), !tbaa !15, !llvm.access.group
                                                                                                                                                                                                                                                                                          ... !12
                                                                                                                                                                                                                                                                                          %31 = fmul <8 x float> %wide.masked.gather, %broadcast.splat13,
                                                                                                                                                                                                                                                                                          ...!llvm.access.group!12
                                                                                                                                                                                                                                                                                         call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %31, <8 x float*> ... %30, i32 4, <8 x i1> <i1 true, i1 true, i
                                                                                                                                                                                                                                                                                          ... i1 true, i1 true>), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                                                                                                                          br i1 %cmp740.i, label %for.body.lr.ph.i16, label %if.end.r_exit.i29
                                                                                                                                                                                                                                                                                                                                                                                                                                                    F
                                                                                                                                                                                                                                for.body.lr.ph.i16:
                                                                                                                                                                                                                                  \%32 = shl < 8 \times i64 > \%26, < i64 32, i64
                                                                                                                                                                                                                                  ... i64 32, i64 32>, !llvm.access.group !12
                                                                                                                                                                                                                                  %33 = ashr exact <8 x i64> %32, <i64 32, i64 32, i64 32, i64 32, i64 32, i64 32, i64
                                                                                                                                                                                                                                  ... 32, i64 32, i64 32>, !llvm.access.group !12
                                                                                                                                                                                                                                  br label %for.body.i17
                                                                                                                                                                                                                   for.body.i17:
                                                                                                                                                                                                                    \text{wec.phi18} = \text{phi} < 8 \text{ x i64} > [\text{w41}, \text{wfor.body.i17}], [\text{zeroinitializer},]
                                                                                                                                                                                                                    ... %for.body.lr.ph.i16 ]
                                                                                                                                                                                                                    \text{%vec.phi19} = \text{phi} < 8 \text{ x float} > [\%40, \%\text{for.body.i17}], [\%31,
                                                                                                                                                                                                                    ... %for.body.lr.ph.i16 ]
                                                                                                                                                                                                                    %34 = add nsw <8 x i64> %vec.phi18, %25, !llvm.access.group !12 %35 = getelementptr inbounds float, float* %0, <8 x i64> %34,
                                                                                                                                                                                                                    ...!llvm.access.group!12
                                                                                                                                                                                                                    %wide.masked.gather20 = call <8 x float>
                                                                                                                                                                                                                    ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %35, i32 4, <8 x i1> <i1 true,
                                                                                                                                                                                                                   ... i1 true, i7 true, i1 true,
                                                                                                                                                                                                                   ... undef), !tbaa !15, !llvm.access.group !12
%36 = fmul <8 x float> %wide.masked.gather20, %broadcast.splat22,
                                                                                                                                                                                                                    ...!llvm.access.group!12
                                                                                                                                                                                                                    %37 = mul nsw <8 x i64> %vec.phi18, %broadcast.splat24, !llvm.access.group
                                                                                                                                                                                                                    ... !12
                                                                                                                                                                                                                    %38 = add nsw <8 x i64> %37, %33, !llvm.access.group !12
                                                                                                                                                                                                                   %39 = getelementptr inbounds float, float* %1, <8 x i64> %38,
                                                                                                                                                                                                                    ...!llvm.access.group!12
                                                                                                                                                                                                                   %wide.masked.gather25 = call <8 x float>
                                                                                                                                                                                                                   ... @llvm.masked.gather.v8f32.v8p0f32(<8 x float*> %39, i32 4, <8 x i1> <i1 true, i1 true, i1
                                                                                                                                                                                                                   ... undef), !tbaa !15, !llvm.access.group !12
                                                                                                                                                                                                                    %40 = call <8 x float> @llvm.fmuladd.v8f32(<8 x float> %36, <8 x float>
                                                                                                                                                                                                                   ... %wide.masked.gather25, <8 x float> %vec.phi19), !llvm.access.group !12
                                                                                                                                                                                                                    call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %40, <8 x float*>
                                                                                                                                                                                                                   ... %30, i32 4, <8 x i1> <i1 true, i1 t
                                                                                                                                                                                                                   ... i64 1, i64 1>, !llvm.access.group !12
%42 = icmp eq <8 x i64> %41, %broadcast.splat27, !llvm.access.group !12
                                                                                                                                                                                                                    %43 = \text{extractelement} < 8 \times i1 > %42, i32 0
                                                                                                                                                                                                                    br i1 %43, label %if.end.r exit.i29.loopexit, label %for.body.i17
                                                                                                                                                                                                                                                                                                     if.end.r exit.i29.loopexit:
                                                                                                                                                                                                                                                                                                      br label %if.end.r exit.i29
                                                                                                                                                                                                                                                                                                      if.end.r exit.i29:
                                                                                                                                                                                                                                                                                                       %44 = add nuw <8 x i64> %vec.phi, <i64 1, i64 1, i64 1, i64 1, i64 1, i64 1,
                                                                                                                                                                                                                                                                                                       ... i64 1, i64 1>
                                                                                                                                                                                                                                                                                                       %45 = icmp eq < 8 \times i64 > %44, %broadcast.splat31
                                                                                                                                                                                                                                                                                                       %46 = \text{extractelement} < 8 \times i1 > %45, i32 0
                                                                                                                                                                                                                                                                                                       br i1 %46, label %pregion for end.i32, label %pregion for entry.entry.i9
                                                                                                                                                                                                                                                                          pregion for end.i32:
                                                                                                                                                                                                                                                                          %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>
                                                                                                                                                                                                                                                                          %47 = icmp eq i64 %index.next, %n.vec
br i1 %47, label %middle.block, label %vector.body, !llvm.loop !19
                                                                                                                                                                                                                                                                                                                                                                                                                        F
                                                                                                                                                                                                                                                        middle.block:
                                                                                                                                                                                                                                                         %cmp.n = icmp eq i64 %umax3, %n.vec
                                                                                                                                                                                                                                                         br i1 %cmp.n, label %gemm.exit, label
                                                                                                                                                                                                                                                         ... %pregion for entry.pregion for init.i.preheader
                                                                                                                     pregion_for_entry.pregion_for_init.i.preheader:
                                                                                                                     % local id y.0.ph = phi i\overline{64} [\overline{0}, %1\overline{2}], [%n.vec, %middle.block]
                                                                                                                     br label %pregion_for_entry.pregion_for_init.i
                                                                                                    pregion_for_entry.pregion_for_init.i:
                                                                                                     % local_id_y.0 = phi i64 [ %59, %pregion_for_end.i ], [ %_local_id_y.0.ph, ... %pregion_for_entry.pregion_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 %mul.i = mul nsw i32 %conv2.i, %6, !llvm.access.group !12
                                                                                                       %mul9.i = mul nsw i32 %conv2.i, %7
                                                                                                      %48 = sext i32 %mul9.i to i64
                                                                                                      br label %pregion for entry.entry.i
                                                                  pregion for entry.entry.i:
                                                                    % local id \bar{x}.0 = phi i64 [ 0, %pregion for entry.pregion for init.i ], [
                                                                   ... %58, %if.end.r exit.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
                                                                    %add.i = add nsw i32 %mul.i, %conv.i, !llvm.access.group !12
                                                                    %idxprom.i = sext i32 %add.i to i64, !llvm.access.group !12
                                                                    %arrayidx.i = getelementptr inbounds float, float* %2, i64 %idxprom.i,
                                                                    ...!llvm.access.group!12
                                                                   %49 = load float, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
                                                                   ...!12
                                                                   %mul6.i = fmul float %49, %4, !llvm.access.group !12 store float %mul6.i, float* %arrayidx.i, align 4, !tbaa !15,
                                                                    ...!llvm.access.group!12
                                                                    br i1 %cmp740.i, label %for.body.lr.ph.i, label %if.end.r_exit.i,
                                                                    ...!llvm.access.group!12
                                                                                                                                                                                                               F
                                 for.body.lr.ph.i:
                                  %sext.i = shl i64 %add1.i.i, 32, !llvm.access.group !12
                                  %50 = ashr exact i64 %sext.i, 32, !llvm.access.group !12
                                  br label %for.body.i, !llvm.access.group !12
for.body.i:
 %indvars.iv.next.i2 = phi i64 [ %indvars.iv.next.i, %for.body.i ], [ 0,
 ... %for.body.lr.ph.i ]
 %51 = phi float [ %57, %for.body.i ], [ %mul6.i, %for.body.lr.ph.i ]
 %52 = add nsw i64 %indvars.iv.next.i2, %48, !llvm.access.group !12
 %arrayidx12.i = getelementptr inbounds float, float* %0, i64 %52,
 ... !llvm.access.group !12
%53 = load float, float* %arrayidx12.i, align 4, !tbaa !15,
 ..!llvm.access.group!12
 %mul13.i = fmul float %53, %3, !llvm.access.group !12
 %54 = mul nsw i64 %indvars.iv.next.i2, %13, !llvm.access.group !12
 %55 = add nsw i64 %54, %50, !llvm.access.group !12
 %arrayidx17.i = getelementptr inbounds float, float* %1, i64 %55,
  ..!llvm.access.group!12
 %56 = load float, float* %arrayidx17.i, align 4, !tbaa !15,
 ..!llvm.access.group!12
 %57 = tail call float @llvm.fmuladd.f32(float %mul13.i, float %56, float
 .. %51) #5, !llvm.access.group !12
 store float %57, float* %arrayidx.i, align 4, !tbaa !15, !llvm.access.group
 ... !12
 %indvars.iv.next.i = add nuw nsw i64 %indvars.iv.next.i2, 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 %if.end.r exit.i.loopexit, label %for.body.i,
 ...!llvm.loop!22,!llvm.access.group!12
                                                                         if.end.r exit.i.loopexit:
                                                                          br label %if.end.r exit.i
                                                                                                                if.end.r exit.i:
                                                                                                                \%58 = add \text{ nuw } i64 \% \text{ local } id x.0, 1
                                                                                                               %exitcond.not = icmp eq i6\overline{4} %58, %umax br i1 %exitcond.not, label %pregion for end.i, label
                                                                                                                ... %pregion for entry.entry.i, !llvm.loop \bar{1}24
                                                                                                                                                 pregion for end.i:
                                                                                                                                                  \frac{1}{859} = \frac{1}{800} and \frac{1}{100} nuw i64 % local id y.0, 1
                                                                                                                                                  %exitcond4.not = icm\bar{p} eq i\bar{6}4 \%59, %umax3
                                                                                                                                                  br i1 %exitcond4.not, label %gemm.exit.loopexit, label
                                                                                                                                                  ... %pregion for entry.pregion for init.i, !llvm.loop !27
                                                                                                                                                                                               gemm.exit.loopexit:
                                                                                                                                                                                                br label %gemm.exit
```

ret void

CFG for '\_pocl\_kernel\_gemm' function

gemm.exit: