```
%10:
                                                                                                                                 %11 = sext i32 %4 to i64
                                                                                                                                  %12 = icmp slt i64 %11, 32
                                                                                                                                 %13 = select i1 %12, i64 %11, i64 32
                                                                                                                                  %14 = \text{sext i} 32 \% 3 \text{ to i} 64
                                                                                                                                  %15 = icmp slt i64 %14, 8
                                                                                                                                  %16 = select i1 %15, i64 %14, i64 8
                                                                                                                                  %mul.i.i = shl i64 %7, 5
                                                                                                                                  %mul3.i.i = shl i64 %8, 3
                                                                                                                                  %cmp638.i = icmp sgt i32 %5, 0, !llvm.access.group !12
                                                                                                                                  %wide.trip.count.i = zext i32 %5 to i64
                                                                                                                                  %17 = icmp ugt i64 %13, 1
                                                                                                                                  %umax = select i1 %17, i64 %13, i64 1
                                                                                                                                  %18 = icmp ugt i64 %16, 1
                                                                                                                                 %umax3 = select i1 %18, i64 %16, 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 %4, 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 %5, 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.splatinsert18 = insertelement <8 x i64> undef, i64 %11, i32 0
                                                                                                                                                                    %broadcast.splat19 = shufflevector <8 x i64> %broadcast.splatinsert18, <8 x
                                                                                                                                                                    ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                    %broadcast.splatinsert21 = insertelement <8 x i64> undef, i64
                                                                                                                                                                    ... %wide.trip.count.i, i32 0
                                                                                                                                                                    %broadcast.splat22 = shufflevector <8 x i64> %broadcast.splatinsert21, <8 x
                                                                                                                                                                    ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                                                    %broadcast.splatinsert25 = insertelement <8 x i64> undef, i64 %umax, i32 0
                                                                                                                                                                    %broadcast.splat26 = shufflevector <8 x i64> %broadcast.splatinsert25, <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.i27 ]
                                                                                                                                                                                       %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.i27]
                                                                                                                                                                                       %19 = add <8 x i64> %vec.ind, %broadcast.splat, !llvm.access.group !12
                                                                                                                                                                                       %20 = trunc <8 x i64> %19 to <8 x i32>, !llvm.access.group !12
                                                                                                                                                                                       %21 = mul nsw <8 x i32> %broadcast.splat6, %20, !llvm.access.group !12
                                                                                                                                                                                       %22 = mul nsw <8 x i32> %broadcast.splat8, %20
                                                                                                                                                                                       %23 = \text{sext} < 8 \times i32 > %22 \text{ to} < 8 \times i64 >
                                                                                                                                                                                       br label %pregion for entry.entry.i9
                                                                                                                                                                                     pregion_for entry.entry.i9:
                                                                                                                                                                                      %vec.phi = phi <8 x i64> [ zeroinitializer, %vector.body ], [ %41,
                                                                                                                                                                                      ... %if.end.r exit.i24 ]
                                                                                                                                                                                      %24 = add <8 x i64> %vec.phi, %broadcast.splat11, !llvm.access.group !12 %25 = trunc <8 x i64> %24 to <8 x i32>, !llvm.access.group !12
                                                                                                                                                                                      %26 = add nsw <8 x i32> %21, %25, !llvm.access.group !12
                                                                                                                                                                                      \%27 = \text{sext} < 8 \times i32 > \%26 \text{ to} < 8 \times i64 >, !llvm.access.group !12
                                                                                                                                                                                      \%28 = getelementptr inbounds float, float* \%2, <8 \times 164 > \%27,
                                                                                                                                                                                      ...!llvm.access.group!12
                                                                                                                                                                                     call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> zeroinitializer, <8 ... x float*> %28, 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
                                                                                                                                                                                      br i1 %cmp638.i, label %for.body.lr.ph.i14, label %if.end.r exit.i24
                                                                                                                                          for.body.lr.ph.i14:
                                                                                                                                            %29 = shl <8 x i64> %24, <i64 32, i64 32, i64 32, i64 32, i64 32, i64 32, i64 32,
                                                                                                                                            ... i64 32, i64 32>, !llvm.access.group !12
                                                                                                                                            \%30 = ashr exact < 8 \times i64 > \%29, < 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.i15
                                                                                                                                   for.body.i15:
                                                                                                                                   %vec.phi16 = phi <8 x i64> [ %38, %for.body.i15 ], [ zeroinitializer,
                                                                                                                                   ... %for.body.lr.ph.i14 ]
                                                                                                                                   \text{%vec.phi17} = \text{phi} < 8 \text{ x float} > [\text{%37}, \text{%for.body.i15}], [\text{zeroinitializer},]
                                                                                                                                   ... %for.body.lr.ph.i14 ]
                                                                                                                                   %31 = add nsw <8 x i64> %vec.phi16, %23, !llvm.access.group !12
                                                                                                                                   %32 = getelementptr inbounds float, float* %0, <8 x i64> %31,
                                                                                                                                   ...!llvm.access.group!12
                                                                                                                                   %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                                   ... x float*> %32, i32 4, <8 x i1> <i1 true, i1 
                                                                                                                                   ... i1 true, i1 true, i1 true, <8 x float> undef), !tbaa !15, !llvm.access.group
                                                                                                                                   %33 = mul nsw <8 x i64> %vec.phi16, %broadcast.splat19, !llvm.access.group
                                                                                                                                   ... !12
                                                                                                                                   %34 = add nsw < 8 \times i64 > %33, %30, !llvm.access.group !12
                                                                                                                                   %35 = getelementptr inbounds float, float* %1, <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, i1 true, i1 true, i1 true, i1 true, i1 true, i1 true>, <8 x float>
                                                                                                                                   ... undef), !tbaa !15, !llvm.access.group !12
                                                                                                                                   %36 = fmul <8 x float> %wide.masked.gather, %wide.masked.gather20,
                                                                                                                                   ...!llvm.access.group!12
                                                                                                                                   %37 = fadd <8 x float> %vec.phi17, %36, !llvm.access.group !12 call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %37, <8 x float*>
                                                                                                                                   ... %28, i32 4, <8 x i1> <i1 true, i1 t
                                                                                                                                   ... i1 true, i1 true>), !tbaa !15, !llvm.access.group !12
                                                                                                                                   %38 = add nuw nsw <8 x i64> %vec.phi16, <i64 1, i64 1, i64 1, i64 1, i64 1,
                                                                                                                                   ... i64 1, i64 1, i64 1>, !llvm.access.group !12
%39 = icmp eq <8 x i64> %38, %broadcast.splat22, !llvm.access.group !12
%40 = extractelement <8 x i1> %39, i32 0
                                                                                                                                   br i1 %40, label %if.end.r exit.i24.loopexit, label %for.body.i15
                                                                                                                                                                                     if.end.r_exit.i24.loopexit:
                                                                                                                                                                                      br label %if.end.r exit.i24
                                                                                                                                                                                      if.end.r exit.i24:
                                                                                                                                                                                      %41 = add nuw <8 x i64> %vec.phi, <i64 1, i64 1, i64 1, i64 1, i64 1, i64 1,
                                                                                                                                                                                      ... i64 1, i64 1>
                                                                                                                                                                                      %42 = icmp eq <8 x i64> %41, %broadcast.splat26
%43 = extractelement <8 x i1> %42, i32 0
                                                                                                                                                                                      br i1 %43, label %pregion for end.i27, label %pregion for entry.entry.i9
                                                                                                                                                                    pregion for end.i27:
                                                                                                                                                                     \%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>
                                                                                                                                                                    %44 = icmp eq i64 %index.next, %n.vec
br i1 %44, label %middle.block, label %vector.body, !llvm.loop !19
                                                                                                                                                         middle.block:
                                                                                                                                                          %cmp.n = icmp eq i64 %umax3, %n.vec
br i1 %cmp.n, label %mm3_kernel1.exit, label
                                                                                                                                                         ... %pregion for entry.pregion for init.i.preheader
                                                                        pregion_for_entry.pregion_for_init.i.preheader:
    %_local_id_y.0.ph = phi i64 [ 0, %10 ], [ %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 [ %56, %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, %4, !llvm.access.group !12 %mul8.i = mul nsw i32 %conv2.i, %5
                                                               %45 = sext i32 %mul8.i to i64
                                                               br label %pregion_for_entry.i
                                            pregion for entry.entry.i:
                                             % local id \bar{x}.0 = phi i64 [ 0, %pregion for entry.pregion for init.i ], [
                                             ... %55, %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
                                             store float 0.000000e+00, float* %arrayidx.i, align 4, !tbaa !15,
                                             ...!llvm.access.group!12
                                             br i1 %cmp638.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
                      %46 = 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 ]
%47 = phi float [ %54, %for.body.i ], [ 0.000000e+00, %for.body.lr.ph.i ]
%48 = add nsw i64 %indvars.iv.next.i2, %45, !llvm.access.group !12
%arrayidx11.i = getelementptr inbounds float, float* %0, i64 %48,
 ..!llvm.access.group!12
%49 = load float, float* %arrayidx11.i, align 4, !tbaa !15,
 ..!llvm.access.group!12
 %50 = mul nsw i64 %indvars.iv.next.i2, %11, !llvm.access.group !12
 %51 = add nsw i64 %50, %46, !llvm.access.group !12
 %arrayidx15.i = getelementptr inbounds float, float* %1, i64 %51,
 ..!llvm.access.group!12
%52 = load float, float* %arrayidx15.i, align 4, !tbaa !15,
...!llvm.access.group!12
%53 = fmul float %49, %52,!llvm.access.group!12
 %54 = fadd float %47, %53, !llvm.access.group !12
store float %54, 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:
                                                                     \%55 = add nuw i64 \% local id x.0, 1
                                                                      %exitcond.not = icmp eq i6\overline{4} %55, %umax
                                                                     br i1 %exitcond.not, label %pregion for end.i, label
                                                                      ... %pregion for entry.entry.i, !llvm.loop 124
                                                                               pregion for end.i:
                                                                               ^{1}\%56 = add nuw i64 % local id y.0, 1
                                                                               %exitcond4.not = icmp eq i64 %56, %umax3
br i1 %exitcond4.not, label %mm3_kernel1.exit.loopexit, label
                                                                               ... %pregion for entry.pregion for init.i, !llvm.loop !27
                                                                                                         mm3 kernel1.exit.loopexit:
                                                                                                          br label %mm3 kernel1.exit
                                                                                                                                mm3 kernel1.exit:
```

ret void

CFG for '\_pocl\_kernel\_mm3\_kernel1' function