```
entry:
                                                                                                                                  \%0 = \text{sext i} 32 \% \text{nj to i} 64
                                                                                                                                  %1 = \text{sext i} 32 \% \text{nj to i} 64
                                                                                                                                  %2 = \text{sext i} 32 \% \text{nk to i} 64
                                                                                                                                  br i1 false, label %scalar.ph, label %vector.ph
                                                                                                                                                                                                           F
                                                                                                                                             vector.ph:
                                                                                                                                               %broadcast.splatinsert = insertelement <8 x i64> undef, i64 %1, i32 0
                                                                                                                                               %broadcast.splat = shufflevector <8 x i64> %broadcast.splatinsert, <8 x i64>
                                                                                                                                              ... undef, <8 x i32> zeroinitializer
                                                                                                                                               %broadcast.splatinsert1 = insertelement <8 x i64> undef, i64 %2, i32 0
                                                                                                                                              %broadcast.splat2 = shufflevector <8 x i64> %broadcast.splatinsert1, <8 x
                                                                                                                                              ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                               %broadcast.splatinsert4 = insertelement <8 x float> undef, float %alpha, i32
                                                                                                                                               %broadcast.splat5 = shufflevector <8 x float> %broadcast.splatinsert4, <8 x
                                                                                                                                              ... float> undef, <8 x i32> zeroinitializer
                                                                                                                                              %broadcast.splatinsert6 = insertelement <8 x i64> undef, i64 %0, i32 0
                                                                                                                                               %broadcast.splat7 = shufflevector <8 x i64> %broadcast.splatinsert6, <8 x
                                                                                                                                               ... i64> undef, <8 x i32> zeroinitializer
                                                                                                                                               br label %vector.body
                                                                                                                                                  vector.body:
                                                                                                                                                   %index = phi i64 [ 0, %vector.ph ], [ %index.next, %for.inc2510 ]
                                                                                                                                                   %vec.ind = phi <8 x i64> [ <i64 0, i64 1, i64 2, i64 3, i64 4, i64 5, i64 6,
                                                                                                                                                   ... i64 7>, %vector.ph ], [ %vec.ind.next, %for.inc2510 ]
                                                                                                                                                   %3 = \text{mul nsw} < 8 \times \text{i} 64 > \text{%vec.ind}, \text{%broadcast.splat}
                                                                                                                                                   %4 = getelementptr inbounds float, float* %tmp, <8 x i64> %3
                                                                                                                                                   call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> zeroinitializer, <8
                                                                                                                                                   ... x float*> %4, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1 true,
                                                                                                                                                   ... i1 true, i1 true, i1 true>), !tbaa !2, !llvm.access.group !6
                                                                                                                                                   %5 = mul nsw <8 x i64> %vec.ind, %broadcast.splat2
                                                                                                                                                   br label %for.body93
                                                                                                                                            for.body93:
                                                                                                                                            \text{%vec.phi} = \text{phi} < 8 \times \text{i}64 > [\%13, \%\text{for.body}93], [zeroinitializer,]
                                                                                                                                             ... %vector.body ]
                                                                                                                                            \%6 = \text{add nsw} < 8 \times i64 > \%\text{vec.phi}, \%5
                                                                                                                                            %7 = getelementptr inbounds float, float* %A, <8 x i64> %6
                                                                                                                                            %wide.masked.gather = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                                            ... x float*> %7, 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 !2, !llvm.access.group
                                                                                                                                            ... !6
                                                                                                                                            %8 = fmul <8 x float> %wide.masked.gather, %broadcast.splat5
                                                                                                                                            %9 = mul nsw <8 x i64> %vec.phi, %broadcast.splat7
                                                                                                                                            %10 = getelementptr inbounds float, float* %B, <8 x i64> %9
                                                                                                                                            %wide.masked.gather8 = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                                            ... x float*> %10, i32 4, <8 x i1> <i1 true, i1 
                                                                                                                                            ... i1 true, i1 true, i1 true, <8 x float> undef), !tbaa !2, !llvm.access.group
                                                                                                                                            %11 = fmul <8 x float> %8, %wide.masked.gather8
                                                                                                                                            %wide.masked.gather9 = call <8 x float> @llvm.masked.gather.v8f32.v8p0f32(<8
                                                                                                                                            ... x float*> %4, 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 !2, !llvm.access.group
                                                                                                                                            ... !6
                                                                                                                                            %12 = fadd <8 x float> %wide.masked.gather9, %11
                                                                                                                                            call void @llvm.masked.scatter.v8f32.v8p0f32(<8 x float> %12, <8 x float*>
                                                                                                                                            ... %4, i32 4, <8 x i1> <i1 true, i1 true, i1 true, i1 true, i1 true, i1 true, i1
                                                                                                                                            ... true, i1 true>), !tbaa !2, !llvm.access.group !6
                                                                                                                                            \%13 = \text{add nuw nsw} < 8 \times i64 > \%\text{vec.phi}, < i64 1, i
                                                                                                                                             ... i64 1. i64 1. i64 1>
                                                                                                                                            %14 = icmp eq <8 x i64> %13, <i64 1024, i64 1024, i64 1024, i64 1024, i64
                                                                                                                                             ... 1024, i64 1024, i64 1024, i64 1024>
                                                                                                                                            %15 = \text{extractelement} < 8 \times i1 > %14, i32 0
                                                                                                                                            br i1 %15, label %for.inc2510, label %for.body93
                                                                                                                                                                                      for.inc2510:
                                                                                                                                                                                       %16 = add nuw nsw < 8 \times i64 > %vec.ind, < i64 1, 
                                                                                                                                                                                        .. i64 1, i64 1, i64 1>
                                                                                                                                                                                       %17 = icmp eq <8 x i64> %16, <i64 1024, i64 1024>
                                                                                                                                                                                       %18 = \text{extractelement} < 8 \times i1 > %17, i32 0
                                                                                                                                                                                       %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>
                                                                                                                                                                                       %19 = icmp eq i64 %index.next, 1024
                                                                                                                                                                                       br i1 %19, label %middle.block, label %vector.body, !llvm.loop !7
                                                                                                                                                                                                                                                                                                                     F
                                                                                                                                                                middle.block:
                                                                                                                                                                 %cmp.n = icmp eq i64 1024, 1024
                                                                                                                                                                 br i1 %cmp.n, label %for.cond.cleanup3, label %scalar.ph
                                                           scalar.ph:
                                                           %bc.resume.val = phi i64 [ 1024, %middle.block ], [ 0, %entry ]
                                                            br label %if.then
                                                        if.then:
                                                         %indvars.iv56 = phi i64 [ %bc.resume.val, %scalar.ph ], [
                                                         ... %indvars.iv.next57, %for.inc25 ]
                                                         %20 = mul nsw i64 %indvars.iv56, %1
                                                         %arrayidx = getelementptr inbounds float, float* %tmp, i64 %20 store float 0.000000e+00, float* %arrayidx, align 4, !tbaa !2,
                                                         ...!llvm.access.group!6
%21 = mul nsw i64 %indvars.iv56, %2
                                                          br label %for.body9
for.bodv9:
 %indvars.iv = phi i64 [ 0, %if.then ], [ %indvars.iv.next, %for.body9 ]
 %22 = add nsw i64 %indvars.iv, %21
%arrayidx13 = getelementptr inbounds float, float* %A, i64 %22
 %23 = load float, float* %arrayidx13, align 4, !tbaa !2, !llvm.access.group
 ... !6
 %mul14 = fmul float %23, %alpha
 %24 = mul nsw i64 %indvars.iv. %0
 %arravidx18 = getelementptr inbounds float, float* %B, i64 %24
 %25 = load float, float* %arrayidx18, align 4, !tbaa !2, !llvm.access.group
... !6
%mul19 = fmul float %mul14, %25
%26 = load float, float* %arrayidx, align 4, !tbaa !2, !llvm.access.group !6
 %add24 = fadd float %26. %mul19
 store float %add24, float* %arrayidx, align 4, !tbaa !2, !llvm.access.group
 .. !6
 %indvars.iv.next = add nuw nsw i64 %indvars.iv, 1
 %exitcond.not = icmp eq i64 %indvars.iv.next, 1024
 br i1 %exitcond.not, label %for.inc25, label %for.body9, !llvm.loop !11
                                    for.inc25:
                                    %indvars.iv.next57 = add nuw nsw i64 %indvars.iv56, 1
                                    %exitcond60.not = icmp eq i64 %indvars.iv.next57, 1024
                                    br i1 %exitcond60.not, label %for.cond.cleanup3, label %if.then, !llvm.loop
                                    ... !12
                                                                                                                                                              F
                                                                         Τ
                                                                                                                         for.cond.cleanup3:
                                                                                                                           ret void
```

CFG for 'mm2 kernel1' function