```
%call = tail call i64 @ Z13get global idj(i32 0) #4
                                                   %conv = trunc i64 %call to i3\overline{2}
                                                   call = tail call i64 @_Z13get_global_idj(i32 1) #4
                                                   %conv2 = trunc i64 %call1 to i32
                                                   %cmp = icmp slt i32 %conv2, %ni
                                                   %cmp4 = icmp slt i32 %conv, %nj
                                                   %or.cond = and i1 %cmp4, %cmp
                                                   br i1 %or.cond, label %if.then, label %if.end
                                                                                           F
                    if.then:
                    %mul = mul nsw i32 %conv2, %nj
                    %add = add nsw i32 %mul, %conv
                    %idxprom = sext i32 %add to i64
                    %arrayidx = getelementptr inbounds float, float* %tmp, i64 %idxprom
                    store float 0.000000e+00, float* %arrayidx, align 4, !tbaa !11
                    %cmp639 = icmp sgt i32 %nk, 0
                    br i1 %cmp639, label %for.body.lr.ph, label %if.end
                                                                           F
                 for.body.lr.ph:
                 %mul8 = mul nsw i32 %conv2, %nk
                  \%0 = \text{sext i} 32 \% \text{nj to i} 64
                 %sext = shl i64 %call, 32
                  %1 = ashr exact i64 %sext, 32
                  %2 = \text{sext i} 32 \% \text{mul} 8 \text{ to i} 64
                 %wide.trip.count = zext i32 %nk to i64
                 br label %for.body
\%3 = \text{phi float } [0.000000e+00, \% \text{for.body.lr.ph}], [\%9, \% \text{for.body}]
%indvars.iv = phi i64 [ 0, %for.body.lr.ph ], [ %indvars.iv.next, %for.body ]
%4 = add nsw i64 %indvars.iv, %2
%arrayidx11 = getelementptr inbounds float, float* %A, i64 %4
%5 = load float, float* %arrayidx11, align 4, !tbaa !11
%mul12 = fmul float %5, %alpha
%6 = mul nsw i64 %indvars.iv, %0
%7 = add nsw i64 %6, %1
%arrayidx16 = getelementptr inbounds float, float* %B, i64 %7
%8 = load float, float* %arrayidx16, align 4, !tbaa !11
%9 = tail call float @llvm.fmuladd.f32(float %mul12, float %8, float %3)
store float %9, float* %arrayidx, align 4, !tbaa !11
%indvars.iv.next = add nuw nsw i64 %indvars.iv, 1
%exitcond = icmp eq i64 %indvars.iv.next, %wide.trip.count
br i1 %exitcond, label %if.end, label %for.body, !llvm.loop !15
                                                         F
                 T
                                                                               if.end:
                                                                                ret void
```

CFG for 'mm2 kernel1' function

for.body: