```
\%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 label %if.then
                         if.then:
                         %indvars.iv56 = phi i64 [ 0, %entry ], [ %indvars.iv.next57, %for.inc25 ]
                         %3 = mul nsw i64 %indvars.iv56, %1
                         %arravidx = getelementptr inbounds float, float* %tmp, i64 %3
                         store float 0.000000e+00, float* %arrayidx, align 4, !tbaa!2,
                         ...!llvm.access.group!6
                         %4 = mul nsw i64 %indvars.iv56, %2
                         br label %for.body9
for.body9:
%indvars.iv = phi i64 [ 0, %if.then ], [ %indvars.iv.next, %for.body9 ]
%5 = add nsw i64 %indvars.iv, %4
%arrayidx13 = getelementptr inbounds float, float* %A, i64 %5
%6 = load float, float* %arrayidx13, align 4, !tbaa !2, !llvm.access.group !6
%mul14 = fmul float %6, %alpha
%7 = mul nsw i64 %indvars.iv, %0
%arrayidx18 = getelementptr inbounds float, float* %B, i64 %7
%8 = load float, float* %arrayidx18, align 4, !tbaa !2, !llvm.access.group !6
%mul19 = fmul float %mul14, %8
%9 = load float, float* %arrayidx, align 4, !tbaa !2, !llvm.access.group !6
%add24 = fadd float %9, %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 !7
                  Т
                                                          F
   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
   ... !9
                     Т
                                                             F
            for.cond.cleanup3:
             ret void
```

entry:

CFG for 'mm2 kernel1' function