```
entry:
                              %conv = zext i32 %num_cols to i64
                              %cmp12 = icmp sgt i32 %num_rows, 0
                              br i1 %cmp12, label %for.body.preheader, label %for.end, !prof !3
                                                                               F
  for.body.preheader:
   %idx.ext1 = sext i32 %dest row to i64
   %add.ptr2 = getelementptr inbounds i8*, i8** %output_array, i64 %idx.ext1
   %idx.ext = sext i32 %source row to i64
   %add.ptr = getelementptr inbounds i8*, i8** %input array, i64 %idx.ext
   br label %for.body
for.body:
%row.015 = phi i32 [ %dec, %for.body ], [ %num rows, %for.body.preheader ]
%input_array.addr.014 = phi i8** [ %incdec.ptr, %for.body ], [ %add.ptr,
... %for.body.preheader ]
%output array.addr.013 = phi i8** [%incdec.ptr4, %for.body], [%add.ptr2,
... %for.body.preheader ]
%incdec.ptr = getelementptr inbounds i8*, i8** %input array.addr.014, i64 1
%0 = load i8*, i8** %input array.addr.014, align 8, !tbaa !4
%incdec.ptr4 = getelementptr inbounds i8*, i8** %output array.addr.013, i64 1
%1 = load i8*, i8** %output array.addr.013, align 8, !tbaa !4
tail call void @llvm.memcpy.p0i8.p0i8.i64(i8* %1, i8* %0, i64 %conv, i32 1,
... i1 false)
\%dec = add nsw i32 %row.015. -1
%cmp = icmp sgt i32 %row.015, 1
br i1 %cmp, label %for.body, label %for.end.loopexit, !prof!3
                                                        for.end.loopexit:
                                                        br label %for.end
                                                                      for.end:
                                                                       ret void
```

CFG for 'jcopy\_sample\_rows' function