ABSTRACT

MINIMIZING HARDWARE PIPELINE BREAKS USING SOFTWARE SCHEDULING TECHNIQUES DURING COMPILATION

A compiler module is disclosed which minimizes pipeline breaks by reordering object code instructions to avoid conflicts between closely grouped instructions to the extent possible. Representation of each object code instruction in a small sequential group is temporarily held in a buffer and is assigned a pair of Attribute Words. Potential conflicts which a newly called instruction may have with those instructions already in the buffer are ascertained by logically AND-ing its Attribute Words with those of the other instructions and examining the result. If a conflict does exist, an attempt is made to resolve it by determining if the conflicting instruction already in the buffer can be moved ahead of one or more other instructions in the buffer such that the conflict is eliminated or minimized. This procedure involves a comparison of the Attribute Words of the candidate instruction to be moved, I_m, with the other instructions in the buffer. If movement of the conflicting instruction is possible and will resolve or minimize the conflict, the instructions in the buffer are reordered as appropriate.

(Figure 2)

IThis Sheet Has Been Replacec

