COL216 Computer Architecture

Lab Assignments 11

ARM CPU with Predication

Objective

The main objective of this assignment is to include full predication in ARM CPU designed so far. The secondary objective is to implement branch and link (BL) instruction.

Scope

There are four flags in ARM CPU – C, V, Z and N. These are updated by DP and multiply instructions if S-bit (bit 20) in the instruction is ‘1’. Flag updating functionality was within the scope of assignments 8 and 10. This assignment adds flag checking and predication functionality to the design. Inclusion of BL instruction completes the implementation of branch class of instructions.

Design

ARM architecture supports full predication. That means it is not just the branch instructions that check the flags, rather all instructions have a condition field (bits 31 to 28) that specifies the manner in which flags are to be checked. The instruction is executed if and only if the specified condition is true.

Logic for checking flags is shown in slides 31 and 32 of Lecture 10. A signal labeled as p carries the result of flag checking, p = ‘1’ (‘0’) indicates that the specified condition is true (false).

Predication of instruction can be done in one of the two ways. One way is to skip the cycles that modify the processor state when p = ‘0’. The other way is go through all the cycles but modify the processor state if and only if p = ‘1’.

Implementation of BL instruction requires target address computation in the same way as B instruction, but in addition to updating PC (register 15), it requires return address (address of the instruction immediately after BL instruction) to be saved in link register (register 14). It may be noted that PC gets incremented by 4 in the first cycle. Therefore, after the first cycle, the return address is available in PC itself.

Testing and demonstration

With this assignment all the ARM instructions that you may have used would have been implemented, except for SWI instruction. Therefore, at this stage you should be in a position to take any of your ARM program and demonstrate it on your CPU after removing or suitably substituting SWI instructions. In case you have been adventurous to use additional instructions (e.g. LDM and STM), you would need to take care of those as well. As a part of the report that you submit on moodle, include the programs tested and the results observed.