COL216 Computer Architecture

Lab Assignment 4 : Work for week 6 - Completing datapath

Datapath modules designed in week 6 are to be integrated to form the complete datapath. This will be a "multi-cycle" datapath, which means that it will allow instruction execution in multiple cycles. As opposed to a "single cycle" datapath, it provides for storage of temporary values computed at the end of intermediate cycles.

There are many ways in which the task of an instruction may be split into steps or micro-operations. What is proposed for this exercise is described below for various instruction groups. For DP and DT instructions Rn, Rd, Rs and Rm denote IR[19-16], IR[15-12], IR[11-8] and IR[3-0], respectively, where IR refers to the instruction. For multiply instructions Rn, Rd, Rs and Rm denote IR[15-12], IR[19-16], IR[11-8] and IR[3-0], respectively.

DP group

1. Fetch instruction from memory, address given by PC. Add 4 to PC.
2. Read operand(s) from register file.
    a. Read operand1 from Rn for all instructions except for move sub-group
    b. Read operand2 from Rm, if it is not immediate type
3. Read shift amount from register Rs in register file, if it is specified by a register
4. Shift/rotate operand2, if required.
5. Perform DP operation. Set flags if required.
6. Write result into register Rd of register file, if it is not compare or test sub-group instruction.

DT group

1. Fetch instruction from memory, address given by PC. Add 4 to PC.
2. Read base/offset from register file.
    a. Read base address from Rn in all cases
    b. Read offset from Rm, if it is specified by a register
3. Shift/rotate the offset, if required.
4. Add/subtract offset to base. In case of store instruction, read data from Rd of register file.
5. Perform memory read/write using address with/without offset (depending upon pre/post indexing). Write address (with offset) into register Rn of register file, if required.
6. Write the data read from memory into register Rd of register file in case of a load instruction.

Branch group

1. Fetch instruction from memory, address given by PC. Add 4 to PC.

2. In case of BL instruction, write PC into lr register in register file. Add 4 to PC.
3. Add offset to PC.

Multiply group

1. Fetch instruction from memory, address given by PC. Add 4 to PC.
2. Read multiplicands from Rn and Rs of register file.
3. Read addend from Rm of register file, if it is MLA instruction. Perform multiplication.
4. Perform addition, if it is MLA instruction.
5. Write result into register Rd of register file.

Notes:

1. Actions shown in blue should be performed subject to the condition specified by IR[31-28] being satisfied.

2. The first step in all the instructions is common. It will help in simplifying controller design if the second step for all instructions is also made common (that is, always read from Rn and Rm, whether required or not). Then controller can look at the instruction type in step 2 and decide to follow different sequences for different cases from step 3 onwards.

3. The data path need not bother about sequencing of the steps or deciding whether a step is required or not. That is the job of the controller. In every cycle, the controller instructs the datapath to perform some step or micro-operations.