CSL216 : Assignment 5 - ARM Assembly Simulator with Multicycle operations and Pipelining Design Document (Updated)

Aakar Sharma (2016CSJ0066) & Sahil Bansal(2016CSJ0008)

April 30, 2018

1 Algorithm for the Simulation of the Pipeline :

- The main function used for simulation of the pipeline is **run()** in **arm.cpp**.

  function **run()**:
  - while(IF/ID.PC < sizeOf (Instruction Vector)*4 + 1016)
    - WB();
    - MEM();
    - EX();
    - ID();
    - IF();
- The value of PC is stored in the IF/ID pipeline register implemented as a structure and updated when the IF() function is called.

- Every pipeline register has a **latency_value** which when becomes 1, then only the main portion of related pipeline stage is executed. Otherwise, stalls are created or propagated.

- function **IF()**:
  - update variable named **instructionIndex**:
    - -1 when no instruction in the pipeline to be fetched
    - -2 when there is a stall due to **loaduse** data-hazard.
    - -3 when there is a stall due to high latency of the instruction
  - when **latency_value** becomes 1 then
    - detect whether stall in next cycle

    * check whether PC has changed in next Pipeline register and update accordingly
    * propagate appropriate values to the next Pipeline register

- function **ID():**
  - all the basic functionalites of ID stage are covered
  - both types of data hazards (EX and MEM) are handled
  - branch and compare instructions are handled here, as if an extra hardware is present in this stage preventing **control hazards**, Thus, no branch prediction implemented.

- function **EX():**
  - all the basic functionalities of EX stage are covered
  - an extra variable **instOnHalt** to store the instruction which takes more than 1 cycle to complete
  - depending on the value of IF/ID.instructionIndex and the type of instruction, the latency is appropriately handled.

- function **MEM():**
  - Similar to EX() stage, it has an **instOnHalt** to control latency of load-store type instructions.
  - it sends a **regWrite** control singal to the write-back stage.

- function **WB():**
  - if(regWrite is ON):
    * at the destination register, write the data from MEM_WB pipeline register.

Handling different types of instructions :

- **Memory-Reference Instructions:**
  - Load instruction goes through the entire 5 stages of the pipeline.
  - Store instruction is active only in the first 4 stages of the pipeline, since there is no write back needed.
- **Arithmetic-logical instructions:**
  - Active in the first 3 stages of the pipeline.
  - Also active in the 5th stage, i.e. WriteBack.
- **Branch Instructions:**
  - These instructions are only active in the first 2 stages of the pipeline.
  - They also update the PC when the branch is taken.

2 List of Assumptions

- IF, ID and WB stages all take one clock cycle.
- The Arithemetic instructions take the given amount of latency only in the EX stage.
- The Memory-reference instructions take the given amount of latency only in the MEM stage.
- The Branch and compare instructions are appropriately executed in the ID stage only.
- Memory has a size of 100 words.
- LDR with offset takes the same amount of time as LDR without offset.

3 Learning/Experimental Observations

IPC has improved from the previous assignment when run for the same program. Thus, pipeline has improved the performance.