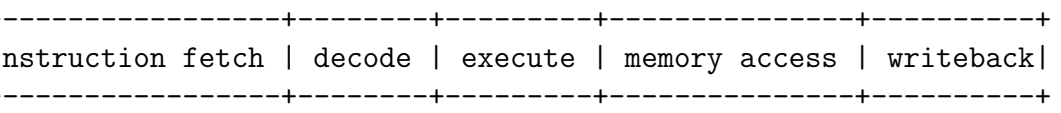
SOLUTION NOTES

Computer Design 2001 Paper 3 Question 6 (SWM)

(a) Why does pipelining introduce data hazards?

[4 marks]

Ans: Data hazards are introduced when data is not available for reading when it is required. Typically this occurs when data in a register file is not up-to-date due to an intermediate result not being written back to the register file immediately. For example, in the five stage pipeline (below) data produced by the execute stage will not be written back to the register file until the writeback stage (i.e. it is delayed by 2 clock cycles).



(b) What mechanism may be used to remove some data hazards?

[4 marks]

Ans: Feed forward paths may be used to forward data between pipeline stages so that intermediate results may be used prior to being written back to the register file. Feed forward paths could be used between the output of the execute stage and the output of the memory access stage and the input of the execute stage.

(c) Why is it not possible to remove all data hazards?

[4 marks]

Ans: Some intermediate results may not have been determined at a particular time so it is not possible to forward them. For example, in the 5-stage pipeline the result from the memory access (for a load instruction) is not ready until after this pipeline stage has completed. Thus, it is not possible to forward the result of a load to a dependent instruction which immediately follows it.

(d) What hardware is required to prevent data hazards from infringing the programmer's model of instruction execution?

[4 marks]

Ans: If data hazards cannot be resolved then the instruction stream must be halted to prevent an incorrect intermediate result from being

used. One approach is to use a scoreboard - a flag for each register in the register file which gets set when an intermediate result is pending and cleared when the intermediate result is available via a feed forward path or has been written back to the register file.

(e) What is the difference between a data hazard and a control hazard?
[4 marks]

Ans: A control hazard occurs when instructions have been incorrectly fetched (and perhaps executed) due to an earlier branch instruction not completing execution. E.g. in the 5-stage pipeline, a branch instruction will not be executed until after 2 other instructions have been fetched. If the branch is taken then these 2 instructions have been fetched in error and must be flushed from the pipeline.