

SOLUTION NOTES

Computer Design 2003 Paper 3 Question 1 (IAP)

The way in which an instruction's operands are specified is dependent on the type of internal storage in the processor. Processors which have no internal storage use memory.

- (a) Describe the advantages of including some form of internal storage, such as an accumulator, in the processor. [5 marks]

Accessing external memory is very slow relative to accessing internal storage. Internal storage (registers) will usually be built from flipflops and can be very efficiently accessed. Having to do two read cycles followed by a memory write cycle for every add instruction would be very slow indeed, and would make effective pipelining virtually impossible due to the huge dependence on the memory system, even if it was cached.

Further, it is harder for the processor to track dependencies between instructions when operands are memory addresses than when they belong to some small set of internal registers.

- (b) Describe three types of internal storage. In each case describe the format of an addition instruction. Indicate the presence of any implicit operands where necessary. [7 marks]

Accumulator: Accumulator architectures have some small number of registers with specialist functions (e.g. accumulator, index register, stack pointer). `adda bar` adds the contents of memory location 'bar' to the current value of the accumulator (implicitly specified).

GPR : general purpose register set architectures have 8,16,32 or more registers that can be used as the source and destination operands in instructions, avoiding the need to reference memory in every instruction, and hence allowing easy pipelining. Some architectures allow two operands per instruction, others 3. E.g. `add srcreg, dstreg` (add srcreg to current contents of dstreg), or `add srcreg1, srcreg2, dstreg`

stack: stack architectures uses internal storage to provide a stack from which instructions consume (pop) and produce (push) source and destination operands respectively. An add instruction would have no explicit operands, and would pop the top two items on the stack, add them, and push the result onto the stack.

- (c) Describe how data-forwarding improves performance in a pipelined load/store architecture (RISC). What common characteristic of programs means this performance improvement is often very significant? [8 marks]

In the classic 5-stage RISC pipeline, it is two cycles before the results of ALU operations are written back to the register file and are thus available for reading by following instructions. Data-forwarding allows the results from ALU operations (and the following stage) to be fed directly back to the inputs of the ALU as source operands for following instructions. A similar forwarding path often exists from the Memory Access stage for read data.

Forwarding is critical to performance, as programs frequently contain chains of interdependent instructions, with no non-dependent instructions that could be scheduled to interleave them.