

## 2009 Paper 7 Question 7

### Comparative Architectures

- (a) What dependencies exist between the instructions in the code fragment below? Identify both true data dependencies and name dependencies, and for each name dependence indicate whether it is an antidependence or an output dependence. [4 marks]

```
LI R1, 25      /* R1=25 */
LI R2, 8       /* R2=8 */
ADD R1, R1, R2 /* R1=R1+R2 */
LD R2, 0(R1)   /* R2=mem[R1] */
```

- (b) How would a hardware register renaming mechanism remove the name dependencies? Illustrate your answer by providing a version of the code showing the destination and source registers for each instruction after renaming has taken place. Clearly state what free physical registers you assume are available prior to renaming. [4 marks]
- (c) Why is the removal of name dependencies beneficial within a superscalar processor? [4 marks]
- (d) In addition to removing name dependencies, for what other purposes may register renaming hardware be used in a superscalar processor? [4 marks]
- (e) The out-of-order execution of ALU instructions in a superscalar processor is only constrained by the availability of functional units and true data dependencies. Why must the out-of-order execution of memory instructions (e.g. load and store instructions) be constrained further? [4 marks]