COMPUTER SCIENCE TRIPOS Part IB – 2012 – Paper 5

2 Computer Design (SWM)

Consider the following code which takes an array `d` of `n` 32-bit integers and performs a bubble sort.

```
for(i=0; i<n; i++) {
  int m = n-i-1;
  for(j=0; j<m; j++)
    if(d[j]>d[j+1]) {
      t = d[j];
      d[j] = d[j+1];
      d[j+1] = t;
    }
}
```

(*a*) Given the following register allocation, how would the inner loop be encoded in assembler for a MIPS-32 style processor (i.e. a RISC machine with a load-store architecture and conditional branches being the only conditional instruction)? Assume that there are no delayed branches (unlike MIPS-32), do not unfold the loop and please do comment your code.

register	variable	description
r4	`d`	base address of array d
r5	`m`	value of m
r6	`j`	register used to hold j
r7	`t`	register to hold temporary t

[8 marks]

(*b*) The classic 32-bit ARM instruction set makes every instruction conditional. How can your code make use of conditional instructions to reduce the number of data-dependent branches? [5 marks]

(*c*) Given the classic 5-stage pipeline (below), how do the control and data hazards differ between your code in parts (*a*) and (*b*)? What is the impact on performance? [7 marks]

instruction fetch	branch, decode & register read	execute	memory access	write back