COMPUTER SCIENCE TRIPOS Part II – 2022 – Paper 9

9 Optimising Compilers (tmj32)

You work for a company that writes compilers for four different processors:

(*a*) A processor for an instruction set that has only six general-purpose (integer) registers.

On a function call, the first argument to the function must be placed in register `r0` and the processor automatically writes the PC to return to after the call into register `r5`.

(*b*) A microcontroller with a static branch predictor and a very small instruction memory.

When the processor encounters a branch it always predicts that it will not be taken and continues speculatively fetching instructions immediately after the branch. All binaries for this processor must fit within 8KiB of memory.

(*c*) An in-order very long instruction word (VLIW) processor with some slow integer operations and a branch-delay slot.

This processor executes four independent operations concurrently in each very long instruction. Although most integer operations take only one cycle to execute, multiplication takes three cycles and division takes eight. The branch-delay slot occurs immediately after a very long instruction containing a branch, and holds an instruction that will be executed regardless of whether the branch is taken or not.

(*d*) An out-of-order multicore with a loop cache.

Each core executes instructions out-of-order by selecting those that have their operands ready from a window of 128 instructions. The loop cache improves power consumption and performance of loops with 16 or fewer instructions. This works by keeping versions of those instructions internally, avoiding the need to fetch and decode them from the instruction cache. The processor contains four homogeneous cores to exploit parallelism within or across programs.

For each processor, describe any challenges and opportunities in register allocation, instruction scheduling and optimisation for the compiler. [5 marks each]