NTNU – Trondheim
Norwegian University of
Science and Technology

Exam TDT4260 Computer Architecture, solution example

Note: For the questions that are not multiple choice, it can in general be several possible answers that can give the maximum no of points.

Examination date: 31. May 2014, 4 hours (09:00 – 13:00)

Problem A: Architecture and Systems, multiple choice (Max 15 points)

A1: b **A2: b** **A3: a** **A4: c** **A5: b**

Problem B: Cache systems (Max 15 points)

B1: a **B2: c** **B3: b** **B4: a** **B5: c**

Problem C: Performance and software (Max 15 points)

C1: c **C2: c** **C3 : c** **C4: a** **C5: b**

Problem D: Parallel processing (Max 15 points)

D1: b **D2: a** **D3: c** **D4: a** **D5: a**

Problem E: Research papers and multiprocessors (Max 20 points)

E1: The processor can do 4-way simultaneous multithreading. This requires 4x program counters, 4x register files, 4x copies of the instruction buffers, 4x of store buffers (Dcache, DTLB details not required), and Thread selection logic including thread select multiplexers.

E2: (a) From the second group of bars, we see that going to super-scalar processor (from 486 to Pentium) gives a more than 3x increase in FP performance. (b) The dotted black arrow in the figure shows clearly that Integer Performance/Watt increased rapidly when the processors changed from deep pipeline back to non-deep pipeline.

E3: A long pipeline with many stages. (A buffer (in fact several) for tokens on the input side). A token must be checked against all tokens stored in the matching unit (MU) that can be tokens for the same instruction. The storage of tokens is organised as a parallel hash table. A hash function is used to route the token to one of the many hash table boards and to address the parallel hash table memory at that board. Each bank compares its tag and destination contents with those of the incoming token, and a match causes the data field of the matching hash location to be output to the store buffer register along with the incoming token. Overflows occur when all the accessed locations are occupied, in which case the nonmatching incoming token is sent to the Overflow Unit and indicator flags are set to notify subsequent tokens of this. The overflow unit is much slower; it is currently (when they wrote the paper) emulated by software in a microcomputer attached to the overflow interface. Thay also say that a special-purpose microcoded processor is under construction.

E4:
(a) Two steps *Map and Reduce*. Map applies a programmer-supplied function to each logical input record. This can be done with extreme degree of parallelism (on thousands of records on thousands of computers in parallel). (The step produces one intermediate result for each input record. The intermediate result is a key-value pairs.). Reduce collects the output from the distributed Map tasks and collapses them using another programmer-defined function.
(b) SIMD executes a single instruction on many different instances of operand data. MapReduce does the same thing at a higher level, in two steps it executes the same function on many different sets of data (might be a more complex set of data than for the more low-level SIMD-case).

E5: The authors developed a technology independent area model for a CMP – found empirically. Core area and cache area are measured in cache byte equivalents (CBE). CBE is the unit for one byte of cache. The figure (Table 3 in the paper) displays die area in terms of the cache-byte-equivalents (CBE), and PIN and POUT columns show how many of each type of processor with 32KB separate L1 instruction and data caches could be implemented on the chip if no L2 cache area were required. (PIN is a simple in-order execution processor, POUT is a larger out-of-order exec processor). (And, for reference, Lambda-squared where lambda is equal to one half of the feature size.) Both the size of a cache and a processor core are measured in number of CBE's. The primary goal of this paper is to determine the best balance between per-processor cache area, area consumed by different processor organizations, and the number of cores on a single die. The CBE unit made it possible to study the relative costs in area versus the associated performance gains --- maximize performance per unit area for future technology generations. With smaller feature sizes, the available area for cache banks and processing cores increases.

…---oooOOOooo---…