NTNU – Trondheim
Norwegian University of
Science and Technology

Department of Computer and Information Science

Exam TDT4260 Computer Architecture
Solution Example

...

Examination date: 18. May 2015

Examination time (from-to): 09:00 – 13:00

Permitted examination support material: code D; No written or handwritten examination support materials are permitted. A specified, simple calculator is permitted.

Other information:

The exam accounts for 80% of the final grade, and the provided points show the maximal number of points that can be achieved on each assignment. Read the problem texts thoroughly. You can answer the questions in English or Norwegian.

For all multiple choice questions: Answer by writing the question-ID and one alternative, like this: "X1 b" where X1 is the question ID and b is your answer. You are awarded 3.0 points for a correct answer and 0 points if you do not answer. If your answer is wrong or you give more than one alternative, you will get -1.5 points.
...

Problem A: Benchmarks and technology trends (Max 12 points)

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

Problem B: Cache systems (Max 9 points)

B1: c B2: b B3: c

Problem C: Processors (Max 9 points)

C1: c C2: a C3: a

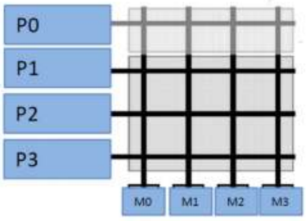
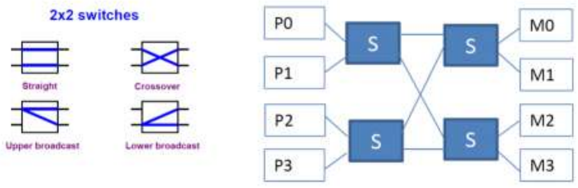
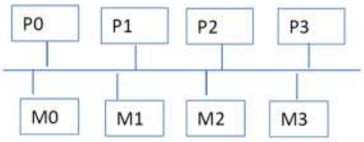
Problem D: Parallel processing and energy efficiency (Max 12 points)

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

Problem E: Multiprocessors and memory systems (Max 18 points)

E1: (Max 6 points)
* Finegrained means to change to a new thread after every instruction, while coarse grained means to switch only when the CPU see costy stalls such as cache-misses or waiting for other ressources.
* SMT exploits TLP together with ILP, i.e. executing instructions from different threads at the same time and also several instructions from one thread (ILP). This can be done by using the same properties of the processor that implements ILP (dynamic scheduling of instructions, simultaneous start of more than one instruction (multi-issue, superscalarity).

E2: (Max 6 points)
* Sketches for each of these three interconnection-techniques are found below. (Note that the mesh in a) (the crossbar) should not have two different gray-tones in the drawing, it is an unimportant “cut&paste error”).

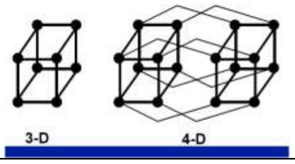
a)		b)	 2x2 switches Straight Crossover Upper broadcast Lower broadcast P0 P1 P2 P3 S S S S M0 M1 M2 M3
c)			

E3: (Max 6 points)
* (Background: A write buffer makes it possible for the processor to continue working while the write buffer prepares to write the word to memory. Different writes to the same block can be combined in the write buffer, in this optimization is called write merging)
* Such I/O addresses cannot allow write merging because separate I/O registers may not act like an array of words in memory. For example, they may require one address and data word per I/O register rather than use multiword writes using a single register.
* This is typically solved by marking the pages as requiring nonmerging write-through by the caches.

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

F1: (Max 5 points)
* An instruction in a dataflow machine can execute when all (incoming) operands are available. These operands arrive asynchronously, so normally an instruction must wait for another operand. Waiting operands wait in the matching unit (MU). (Operands may appear in different contexts, so in MDM operands (data, called tokens) are tagged with a context-identifier). It can be a huge amount of waiting operands, and to provide fast look-up (both instruction id and tag must match) MDM use hashing and parallel search hardware. Some operands are «alone» and they bypass the MU. When the MU storage is full (those part of the storage found by the hashing-function), the operands are sent to an overflow unit that is implemented in Software.

F2: (Max 6 points)
* a) take two HCs called A and B of size 8 and interconnect each node x in A with the same node x in B, see figure. In a 8-node (3D) HC the addresses will be in binary from 000 to 111. Going to 4D means adding a new bit in the address, 0000 to 1111.
* b) A clear disadvantage is that the processor boards or chips or whatever have all 3 ports for connecting to other processors in a 3D HC, but going to 4D requires every such physical processor to be extended with another physical HW-port for such communication. (It will in general not be possible without designing the processor with HW for the maximum configuration).

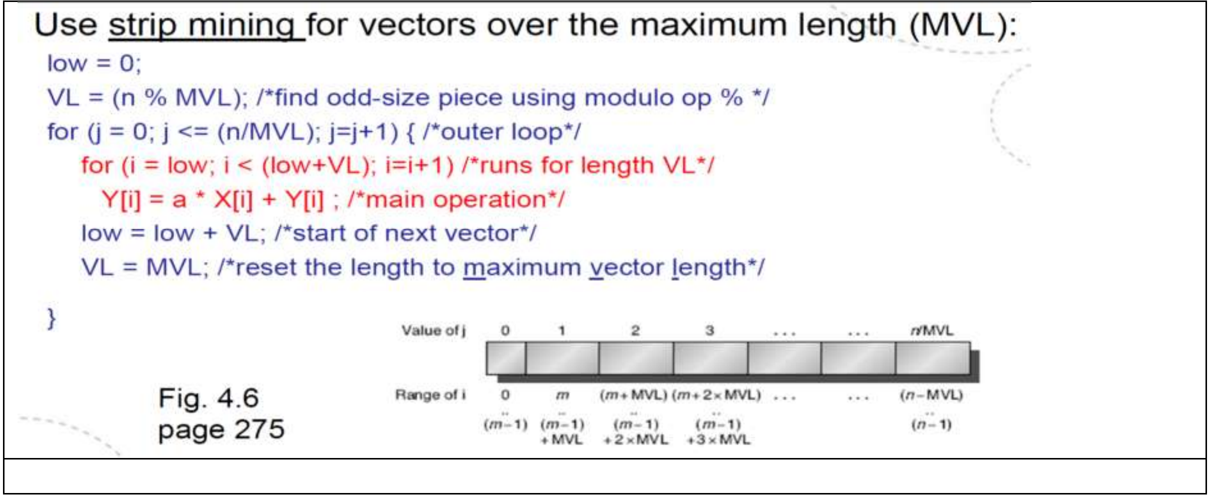
a)	

F3: (Max 5 points)

* They made a technology independent area model – found empirically, – core area and cache area measured in cache byte equivalents (CBE). In different technologies they found how much chip area is needed for one byte of cache (CBE), and how much chip area (measured in CBE) is needed for a given processor core. The paper study the relative costs in area versus the associated performance gains --- maximize performance per unit area for future technology generations. A given chip has a given amount of CBEs, and the designer can decide to have few large and powerful cores, many smaller cores, and also much or little area used for caches --- all configurations summing up the no of CBE to the total allowed by that chip. With smaller feature sizes, the available area for cache banks and processing cores increases. 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.

F4: (Max 4 points)

LF: * see textbook page 275, (Can be explained simpler, a big loop and a small «remainder-loop»)



Use strip mining for vectors over the maximum length (MVL):

```
low = 0;
VL = (n % MVL); /*find odd-size piece using modulo op % */
for (j = 0; j <= (n/MVL); j=j+1) { /*outer loop*/
   for (i = low; i < (low+VL); i=i+1) /*runs for length VL*/
     Y[i] = a * X[i] + Y[i] ; /*main operation*/
   low = low + VL; /*start of next vector*/
   VL = MVL; /*reset the length to maximum vector length*/

}
```

Fig. 4.6
page 275

…---oooOOOooo---…