Computer Architecture, EITF20
Final Exam

LUND
UNIVERSITY
Electrical and Information Technology
Jan. 11, 2018

The exam consists of 5 problems with a total of 50 points.
Grading: 20 p $\leq$ grade 3 $<$ 30 p $\leq$ grade 4 $<$ 40 p $\leq$ grade 5

Instructions:

- Turn off and put away your mobile phone – **No mobile phones**
- You may use a pocket calculator and an English dictionary on this exam, but **no other aids**
- Please start answering each problem on a new sheet – **New problem $\Longrightarrow$ New sheet**
- Write your anonymous code on each sheet of paper that you hand in – **Code on each sheet**
- You must motivate your answers thoroughly.
  Show all your calculations.
  If there, in your opinion, is not enough information to solve a problem, you can make reasonable assumptions that you need in order to solve the problem.
  State your assumptions clearly!

GOOD LUCK :-)

Problem 1

The CPU performance equation can be formulated as below.

a) Fill in the 4 terms with "?" in the equation. (2)

$$\textbf{CPU execution Time} =$$
$$(?) * \left(CPI_{ideal} + \frac{structural\ stalls}{instruction} + \frac{data\ hazard\ stalls}{instruction} + (?) \right.$$
$$\left. +(?) * miss\ rate * (?)) * T_C \right.$$

b) Describe each of the following items/concepts concerning computer architecture. Where relevant also give an example. Describe how these technologies affect the CPU performance equation. (point out **ALL** the possible effects)

1) Complex Instruction Set Computers (CISC) (2)

2) Multi-level cache (2)

3) Reorder buffer (2)

4) Deeper pipeline (2)

Problem 2

Consider following part of an assembly–language program:

```
1: DADD R5,R2,R3      ;Addition R5 = R2 + R3
2: LW R8, (R5)        ;Load word R8=Mem(R5)
3: MUL R9, R8, R5     ;Multiply R9=R8*R5
4: ADDDI R5, R5, #4   ;Add immediate R5=R5+imm
5: LW R8, (R5)        ;Load Word R8=Mem(R5)
```

Basic architecture setup: A standard 5-stage pipeline architecture without forwarding; One arithmetic functional unit (for both multiplication and addition); All instructions take one cycle in the EXE-stage **except** MUL which takes 6 cycles in the EXE-stage; All memory accesses are cache hits (perfect memory system).

- Draw an instruction execution timing table (clock number vs instruction number with entries showing the pipeline stage) of the above program. (4)

Now, we improve the basic pipeline architecture with 2 arithmetic functional units and Tomasulo technique with 2 reservation stations for each arithmetic functional unit. Assume the execution time for each individual instruction remains the same.

- Describe and specify the Tomasulo implementation in the pipeline (Key concept and main changes in hardware). (2)
- Draw the instruction execution timing table (clock number vs instruction number with entries showing the pipeline stage). (4)

Problem 3

We are interested in comparing two proposed enhancements to a baseline processor to decide which design offers highest performance. All processors assume an average CPI of 1.5 together with cache. Assume instruction fetch has 100% cache hit rate (i.e., instruction cache is perfect). Each access between cache and main memory to read or write a cache block requires a setup time of 30 clock cycles with each transfer of 32-bit data adding another 2 clock cycles penalty. The cache size for all 3 processor designs is 32,768 bytes, i.e. 32 kB.

Instruction mix	
instruction type	**% of all instructions**
load	23.75
store	9.66
uncond branch	5.84
cond branch	18.45
int computation	42.30
fp computation	0.00

1. The baseline processor runs at 3.5 GHz clock frequency, has a cache with 128 sets, associativity 1 and write-back with an average of 20% dirty blocks.
2. The first proposed enhancement suggests to use a cache with 32 sets and associativity 2 while still having an average of 20% dirty blocks for write-back. To improve performance the execution time of all integer computations is decreased by a factor of 3.
3. The second proposed enhancement suggests a cache with 32 sets, associativity 4 and 30% of dirty blocks for write-back. Moreover, due to improved implementation, all conditional branches are executed 20% faster (on average).

Following cache miss-ratio are expected depending on configuration:

Cache miss-ratio			
No. of	**Associativity**		
sets	**1**	**2**	**4**
16	0.778109	0.594727	0.303048
32	0.597252	0.312289	0.136385
64	0.397689	0.145433	0.083915
128	0.207784	0.088052	0.051590

Clock cycle time dependence on associativity is:

	Associativity		
	1	**2**	**4**
relative clock cycle time	1	1.06	1.093

Assume that the processor cannot continue until the complete block is fetched into the cache on a cache miss.

- Which of the 3 designs will give best performance? (8)

Caches play an important role in any modern computer system to increase overall performance and speed up program execution.

- Shortly discuss which are the four main categories of cache performance optimization and relate those to the formula for average memory access time. (2)

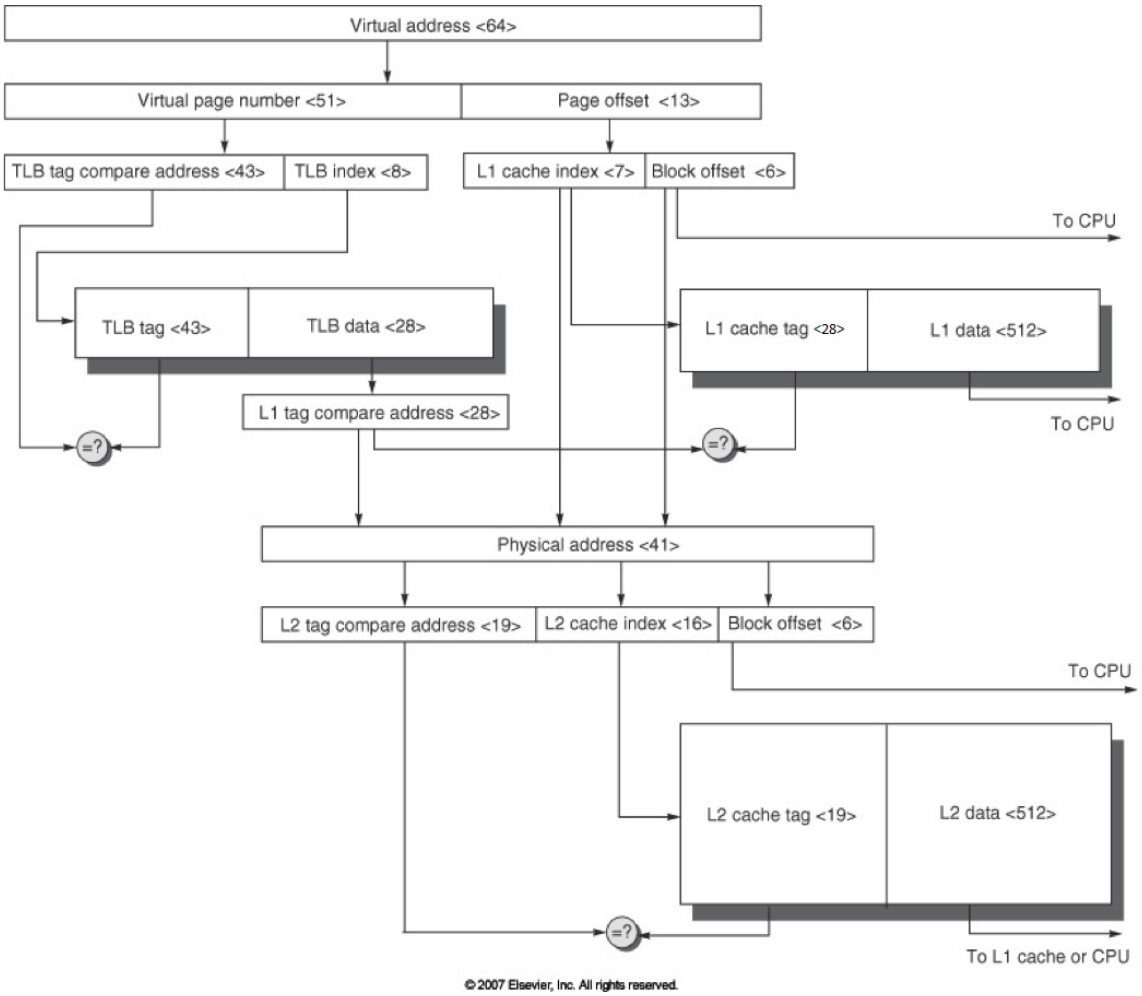
Problem 4

The following figure shows a hypothetical memory hierarchy going from virtual address to L2 cache access. The virtual address is 64-bit and the physical address is 41-bit. The corresponding address bit partitioning is illustrated in the figure (numbers in the ⟨⟩ are the number of bits).

a) Calculate Page size, L1 cache size, TLB set-associativity (TLB has 256 entries), and L2 set-associativity (L2 cache size = 8MByte). (4)

b) Briefly describe the operation steps required from virtual address to L2 cache access (in cases of both cache miss and cache hit). (4)

c) L1 Cache is virtually indexed and physically tagged. Discuss this addressing scheme comparing to virtually indexed and tagged scheme and physically indexed and tagged scheme. (2)



Problem 5

Consider a multiprocessor system with one shared main memory and three processors. Each processor has its own local cache.

a) Describe the cache inconsistency problem in multiprocessor systems with one shared memory. Also, **explain** three main reasons that can cause cache inconsistency in such systems. (4)

b) Consider that MESI protocol with the state diagram shown in Fig. 1, is used in all three caches of this system to solve the inconsistency problem.
Now, consider the sequence of instructions listed in Table 1, which read the content of a certain address in the main memory, i.e. "X", and modify its value. Fill in Table 1 by writing the state of MESI state diagram in all caches and the content of "X" in caches and main memory after execution of each instruction. Note that the initial state of all caches in MESI protocol is "Invalid" and the initial content of "X" in the main memory is equal to **"5"**. (6)

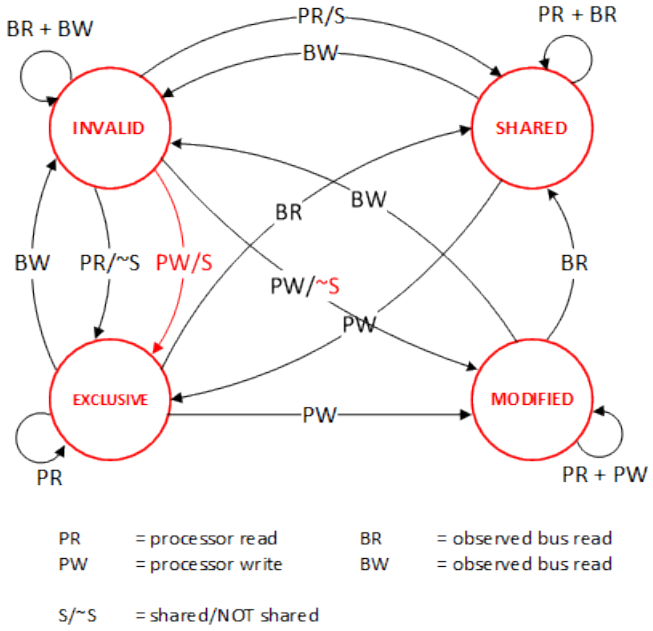


Figure 1: State diagram of MESI protocol.

	Processor1 (P1)		**Processor2 (P2)**		**Processor3 (P3)**		**Main Mem.**
Instruction	MESI Status	content of "X"	MESI Status	content of "X"	MESI Status	content of "X"	content of "X"
P2 Reads "X"							
P3 Reads "X"							
P1 Writes "7" to "X"							
P3 Reads "X"							
P2 Writes "4" to "X"							
P1 Reads "X"							

Table 1: Running a sequence of instructions in the multiprocessor system (Problem 5).