Computer Architecture, EITF20
Final Exam

LUND
UNIVERSITY
Electrical and Information Technology
Jan. 14, 2017

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

a) Pair each concept from table **I** with the concept from table **II** that is most closely associated. (Each answer should be a number-letter pair.) (5)

I		II	
1.	page fault	A.	miss in the memory system
2.	reservation station	B.	WAR hazard
3.	write allocate	C.	output dependence
4.	WAW	D.	hit in the memory system
5.	little-endian	E.	forwarding
		F.	anti-dependence
		G.	the least significant unit is stored at the lowest memory address
		H.	RAW hazard
		I.	the least significant unit is stored at the highest memory address
		J.	issue multiple instructions dynamically each clock cycle

Briefly (1-3 sentences) describe each of the following items/concepts concerning computer architecture. Where relevant also give an example.

b) Cache coherence (1)

c) Multi-level page table (1)

d) Precise exceptions (1)

e) Speculative execution (1)

f) Inverted page table (1)

Problem 2

The CPU performance equation can be formulated as below.

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

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

b) Describe how these technologies affect the CPU performance equation. (point out ALL the possible effects)

1) Stack instruction-set architecture (2)

2) Larger cache block (2)

3) Reorder buffer (2)

4) Deeper pipeline (2)

Problem 3

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
3: ADDDI R5, R5, #4   ;Add immediate R5=R5+imm
4: LW R9, (R5)        ;Load Word R9=Mem(R5)
5: SUB R5, R3, R2     ;Subtract R5=R3-R2
```

All instructions take one cycle in the EXE-stage except MUL which takes 6 cycles in the EXE-stage. Assume all memory accesses are cache hits.

- Draw a table (clock number vs instruction number with entries showing the pipeline stage) for the execution pattern of the above program in a standard (with the exception for MUL) 5-stage pipeline without forwarding. (2)
- Discuss improvements in execution timing if you were to execute it on this pipeline with forwarding added. Describe and specify the possible forwarding paths in the pipeline and draw the table (clock number vs instruction number with entries showing the pipeline stage). (3)
- Discuss improvements in execution timing if you were to execute it on a pipeline with 2 arithmetic functional units and Tomasulo technique with 2 reservation stations for each arithmetic functional unit. Describe and specify the Tomasulo implementation in the pipeline and draw the table (clock number vs instruction number with entries showing the pipeline stage). (5)

Problem 4

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 1.2 together with unified cache. Each access between cache and main memory to read or write a cache block requires a setup time of 40 clock cycles with each transfer of 32-bit data adding another 6 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 25% dirty blocks.

2. The first proposed enhancement suggests to use a cache with 32 sets and associativity 2 while still having an average of 25% 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 40% of dirty blocks for write-back. Moreover, due to improved implementation, all conditional branches are executed 10% 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?(**Show all calculations!**) (10)

Problem 5

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.

a) Calculate L1 cache size, Page Size, L2 cache block size and L2 set-associativity (L2 cache size = 8MB). (4)

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

c) Point out potential mistakes in the address partitioning in the figure and explain why they are wrong. (2)