2019-04-15

Exam in DAT 105 (DIT 051) Computer Architecture

Time: April 24, 2019 14 – 18

Person in charge of the exam: Per Stenström, Phone: 0730-346 340

Supporting material/tools: Chalmers approved calculator.

Exam Review: On May 6, 2019 10-12 in Room 4128

Grading intervals:

- **Fail**: Result < 24
- **Grade 3**: 24 <= Result < 36
- **Grade 4:** 36 <= Result < 48
- **Grade 5:** 48 <= Result

NOTE 1: Bonus points from Real-stuff studies and Quizzes will be added to the exam results for approved exams used solely for higher grades.

NOTE 2: Answers must be given in English

GOOD LUCK!
Per Stenström

CHALMERS UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
412 96 Göteborg
Visiting address: Rännvägen 5
Phone: 031-772 1761 Fax: 031-772 3663
Org. Nr: 556479-5598
E-mail: pers@chalmers.se

[General disclaimer: If you feel that sufficient facts are not provided to solve a problem, either 1) ask the teacher when he visits the exam, or 2) make your own additional assumptions. Additional assumptions will be accepted if they are reasonable and required to solve the problem. Always make sure to motivate your answers.]

ASSIGNMENT 1

1A) We want to compare the performance of two computer systems A and B using the execution times of four programs P1, P2, P3 and P4. The table below lists the execution times of the programs on the two systems and a reference system R.

	P1 [s]	P2 [s]	P3 [s]	P4 [s]
System A	**1**	**10**	**5**	**4**
System B	**0.5**	**17**	**2**	**2**
System R	**2**	**20**	**10**	**8**

i) Derive the arithmetic mean of the execution times to compare the performance of the two systems (A and B). Why is the comparison unfair? **(3 points)**

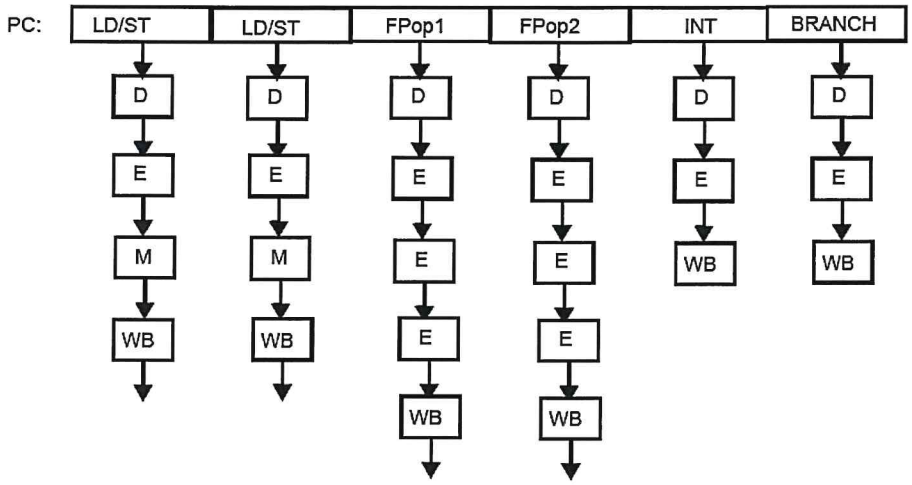
ii) Derive the geometric mean speedup over the reference machine R for the two systems A and B. Why is this metric better than the other? **(3 points)**

1B) We want to reduce the execution time of a program using a multiprocessor system. The execution time of the original sequential program is 10 seconds. Now assume that 80% of the sequential program execution can be distributed evenly across processors in a multiprocessor system. What is the speed-up achieved on ten processors? What is the maximum speed-up that can be ever achieved given an infinite number of processors? **(3 points)**

1C) Assume that a memory instruction has an average CPI=2 and that the average CPI for the rest of the instructions is CPI=1. Given that 20% of the instructions are memory instructions, that a program executes 1,000,000,000 instructions, and the clock frequency is 1 GHz, what is the execution time of the program? **(3 points)**

ASSIGNMENT 2

We consider in this assignment a VLIW architecture that can issue two memory, two floating-point, one integer, and one branch instruction each cycle according to the pipeline organization below. There are no forwarding units.



2A) How many cycles does it take to resolve a RAW dependency between instructions of different types, i.e., between a floating-point operation and a subsequent floating point operation, between a floating-point operation and a subsequent store operation, between a load operation and a subsequent floating point operation, and between a store operation and a subsequent load operation? **(3 points)**

2B)
Consider the following simple computation:

```
LOOP: L.D    F0, 0(R1)
      ADD.D  F4, F0 F2
      S.D    F4,0(R1)
      SUBI   R1,R1, #8
      BNE    R1, R2, LOOP
```

We want to use software pipelining to execute the loop on the VLIW architecture. Derive the kernel of a *software-pipelined loop* by filling out the schedule table below for as many cycles needed to fill the "software pipeline" (i.e., not the VLIW pipeline). **(3 points)**

	ITE 1	ITE 2	...
INST 1			
INST 2			
...			

2C)

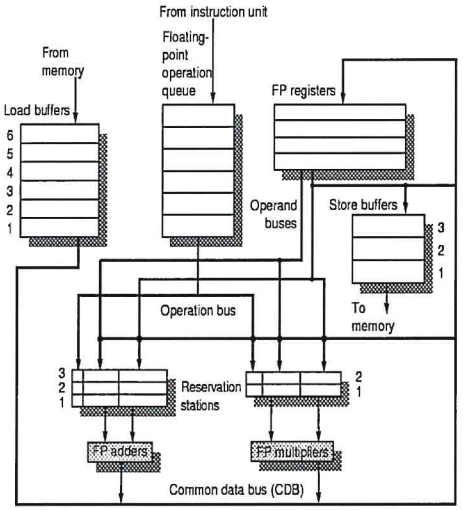
Under what circumstances do WAR hazards **not** occur? In case they occur, show how *Rotating Registers* can be used to avoid them by first explaining how Rotating Registers work and then by showing how the kernel is modified to make use of them. **(3 points)**

2D)
Consider the following code:

```
LW   R4, 0(R1)
ADDI R6,R4,#1
BEQ  R5,R4,LAB
LW   R6,0(R2)
LAB: SW R6, 0(R1)
```

Use *Predicated Instructions* to eliminate the branch in the code above and clearly define the semantics of the used predicated instructions. **(3 points)**

ASSIGNMENT 3



The above diagram shows a pipeline with Tomasulo's algorithm. Consider the following code:

SUB F0, F1,F2 (8 cycles)
DIV F3,F0,F4 (10 cycles)
ADD F4, F5,F6 (6 cycles)

3A) Show all name dependences in the code and explain how Tomasulo's algorithm avoids translating them into hazards by explaining **in detail** what happens when the instructions are (i) issued (ii) reading their operands and (iii) are being executed. **(4 points)**

3B) Assuming that there are two additional functional units and a division functional unit and that a single instruction is issued every cycle. The instruction latencies through the functional

units are listed above. How long time does it take until the last instruction has written back the result to the register file? **(2 points)**

3C) We want to add support for speculative execution. Explain how a reorder buffer and another pipeline stage called COMMIT can allow the following instruction sequence to execute speculatively and what is done in each of the four stages.

```
BNEZ R1, LABEL
ADDF  F1, F2, F3
DIVD  F4, F1, F2
SUBD  F2, F5, F6
```

Assume that the branch instruction is predicted NOT to be taken and that the prediction is correct. **(3 points)**

3D) Explain what problem a branch target buffer solves and how it is organized **(3 points)**

ASSIGNMENT 4

4A) What does it mean that inclusion is imposed between a level 1 and level 2 cache in a cache hierarchy with respect to the content and state of each cache block at level 1? **(3 points)**

4B) Consider the following program:

```
LOOP: LD.S F0, 0(R1)
      ADDI R1,R1,#8
      J LOOP
```

Assume that every fourth load instruction will miss in the cache and that it takes the same time to service a miss as to execute four iterations. The ISA supports a prefetch instruction (PREF DISP(Rx)) that prefetches from memory location DISP + (Rx). Rewrite the code so that prefetches get scheduled so that they cancel all the cache misses. **(3 points)**

4C) Explain in detail how a lock-up free cache works. **(3 points)**

4D) Mention one approach by which the number of misses for each of the Cs in the 3C miss classification can be reduced **(3 points)**

ASSIGNMENT 5

5A) A computer architect wants to establish the relative performance between a system with a blocking and a non-blocking cache. In the non-blocking cache there are 4 miss-status-holding registers. The cache-hit time is 1 cycle and the miss penalty is 100 cycles for both caches. Consider the following code:

```
for (i=0; i<100; i++)
   C+=A[i];
```

Assuming that there are 4 instructions between each load instruction, the CPI is 1 for all instructions except load instructions that miss in the cache, how much faster does the program run on the system with a non-blocking cache. **(4 points)**

5B) Explain what interleaved, blocked, and simultaneous multithreading is and what is required to support these techniques in a simple 5-stage pipeline. **(4 points)**

5C) Explain the principle difference between 4-instruction wide VLIW architecture and a 4-issue speculative superscalar processor. **(4 points)**

*** ***GOOD LUCK!*** ***