CHALMERS

2020-12-29

Exam in DAT 105 (DIT 051) Computer Architecture

Time: January 5, 2021, 14-18 (Canvas)

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

Supporting material/tools: Chalmers approved calculator, textbook “Parallel Computer Organization and Design” by Dubois et al.

Exam Review: More information on this will be available via Canvas

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

NOTE 3: Read the document Instructions for Canvas exam carefully. It is available at Canvas

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

A) We know that the geometric mean speedup of Machine A over a reference machine R is 2 for four applications. The speedup of Machine A over R is 4 and 2 for two other applications. **What is the geometric mean speedup of A over R for all six applications? (3 points)**

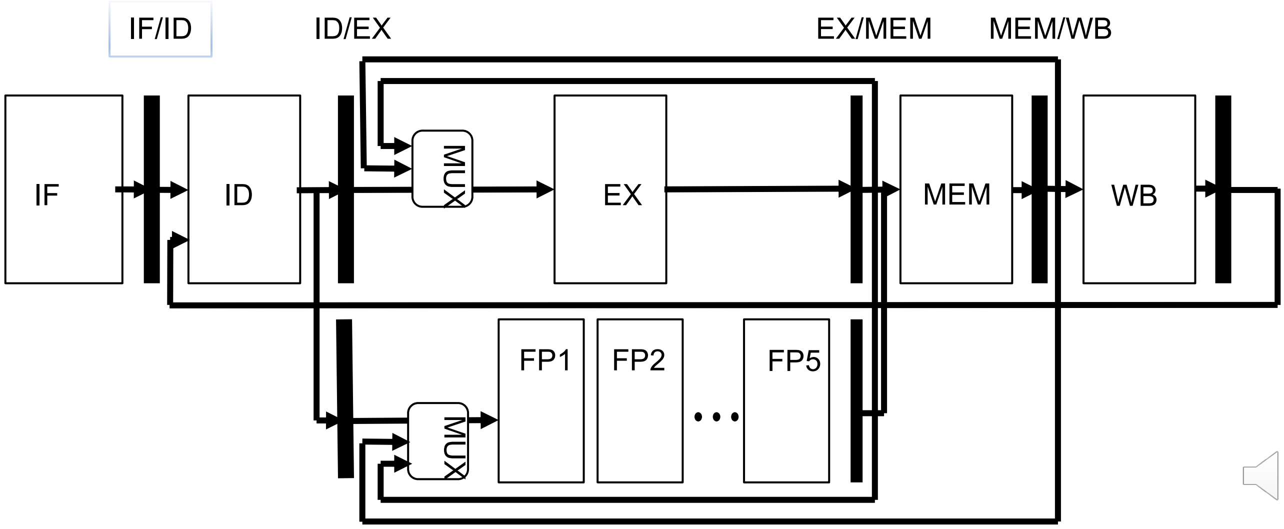
B) Let's assume that a program run from the start to the end executes 10^8 instructions and that it finishes after 2 seconds. CPI is 5 for all instructions that do not cause any cache misses and 100 for those instructions that do cause cache misses. The macine is clocked at 1 GHz. **How many cache misses are triggered? (3 points)**

C) A computer architecture has the option to either work on a new cache design that will reduce the number of cache misses by a factor of two for the machine in in B or improve the performance of a floating-point unit that will cut down CPI from 5 to 2 for the instructions that do not cause cache misses? **Which design alternative leads to the best improvement in performance? (3 points)**

D) Assume that every fourth instruction is a branch and that CPI=1 for all instructions except for branches that execute in 4 cycles if the branch prediction is wrong and in two cycles if branch prediction is correct. **What branch prediction accuracy would make the average CPI=1.5? (*3 points*)**

ASSIGNMENT 2

We consider in this assignment a pipeline with a 5-stage pipelined floating-point unit and a single-stage execution unit that executes integer, load/store and branch instructions. There are forwarding units from the output of each execution unit and from the memory stage.



2A) Consider the following instruction sequence:

I1: ADD F0,F1,F2
I2: ADD F3,F4,F5
I3: ADD F4,F3,F6

Determine the number of cycles it takes from I1 is fetched until I3 reaches the MEM stage if i) the floating-point execution unit is not pipelined and ii) in case it is. (3 points)

2B) Consider the following instruction sequence:

I1: ADD F0,F1,F2
I2: ADD R1,R2,R3
I3: ADD R4,R1,R2
I4: ADD R5,R6,R7
I5: ADD R8,R9,R10
I6: ADD R11,R12,R13
I7: ADD R14,R15,R16

In what order will the instructions reach the MEM stage in the case the floating-point unit is fully pipelined? (3 points)

2C)
We want to use static scheduling techniques to **eliminate** cycles lost due to various hazards in the code below. The numbers inside the parentheses show the number of cycles a certain instruction has to wait before it can start executing.

```
I1 Loop L.S F0,0(R1)    (1)
I2      L.S F1,0(R2)    (1)
I3      ADD.S F2,F1,F0  (2)
I4      S.S F2,0(R1)    (5)
I5      ADDI R1,R1,#4   (1)
I5      ADDI R2,R2,#4   (1)
I6      SUBI R3,R3,#1   (1)
I7      BNEZ R3,Loop    (3)
```

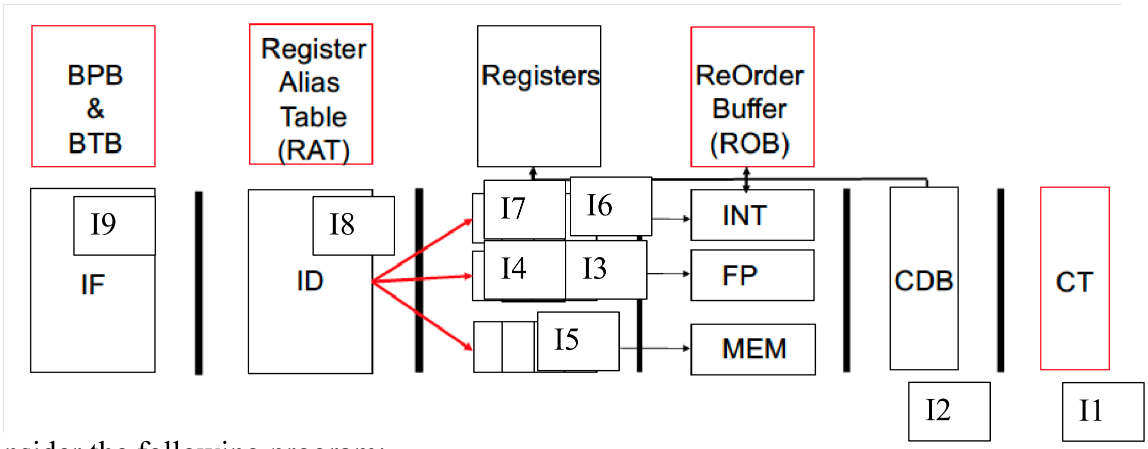
i) Mark all Read-After-Write hazards in the code.
ii) Show how static scheduling can eliminate the extra cycles for the store instruction (I4).
iii) How many cycles are left to remove after the scheduling in ii) and what technique can be used to go after these cycles?
(6 points)

ASSIGNMENT 3

The diagram below shows a pipeline with support for speculative execution. There are three functional units: one for integer/branch instructions (INT); one for floating-point instructions (FP) and one for memory instructions (MEM). There is a branch prediction mechanism (BPB) using 2 bits for prediction, meaning that the prediction is changed every in response to two mispredictions in a row. In addition, a branch target buffer (BTB) is in the IF stage.

For the functional units, it takes a single cycle to execute an INT instruction, three cycles for an FP instruction, a single cycle for a load and two cycles for a store in the MEM unit.

The pipeline supports register renaming using a register alias table (RAT) and data hazards are handled using the Tomasulo algorithm. Speculation is enabled by a reorder buffer and speculatively executed instructions are committed in the commit stage (CT).



Consider the following program:

```
I1: LOOP: L.S    F0,  0(R1)
I2:       L.S    F1,  0(R2)
I3:       ADD.S  F2,  F1,  F0
I4:       ADD.S  F1,  F3,  F4
I5:       SD     F4,  0(R1)
I6:       ADDI   R1,  R1,#8
I7:       ADDI   R2,  R2,#8
I8:       SUBI   R3,  R3,#1
I9:       BNEZ   R3,  LOOP
```

3A)

The diagram shows a snapshot of the execution of the program and in which pipeline stage each individual instruction is in that snapshot. **Which of the operands are available in the register file and which are available from the ROB? (3 points). Please fill out the content of the RAT given the content of the ROB below for the instructions under execution and their operands. (3 points)**

RAT

Register	Tag	Status
R1	--	--
R2	--	--
R3	--	--
F0	--	--
F1	--	--
F2	--	--
F3	--	--
F4	--	--

ROB

Entry	Instruction	Dest. reg	Value	Complete
16	L.S F0,0(R1)	F0	1.2E-4	YES
17	L.S F1,0(R2)	F1	5.8E-3	YES
18	ADD.S F2,F1,F0	F2	--	NO
19	ADD.S F1,F3,F4	F1	--	NO
20	SD F2,0(R1)	F2	N/A	NO
21	ADDI R1,R1,#4	R1	N/A	NO
22	ADDI R2,R2,#4	R2	N/A	NO
23	SUBI R3,R3,#1	R3	N/A	NO
24	BNEZ R3,Loop	F0	N/A	NO
25	L.S F0,0(R1)		N/A	NO

3C) Explain in detail what happens in the Commit (CT) stage when a branch that is incorrectly predicted is committed (**3 points)**

3D) Assume that the branch predictor is initialized to Not-Taken and that the loop in 3A) is executed 100 times. What is the fraction of correct branch predictions in percent? (**3 points)**

ASSIGNMENT 4

4A) Explain which of the statements, below, are **not true** and why they are **not true.**
For a lockup-free (or non-blocking) cache the following holds:

i) It blocks if there is an outstanding cache miss
ii) A miss-status-holding register is allocated only when a primary miss is encountered
iii) A miss-status-holding register is needed for both primary and secondary misses
iv) Prefetching requires a cache to be lockup-free to be effective

Note: A wrong answer cancels a correct answer. **(2 points)**

4B) The following code is executed:

```
LW R1, 0(R2)
ADDI R4,R1, #4
/** Block A **/
BEQ R5,R6,LAB1
LW R4,0(R2)
LAB: SW R4,0(R3)
/** Block B **/
LAB1: ADDI R4,R1,#4
        J LAB
/** Block C **/
```

When it is being profiled, it turns out that the most likely path is Block A followed by Block C. Use trace scheduling to minimize the number of instructions in the trace going through these blocks. **(4 points)**

4C) Explain by applying the concept of predication to an addition instruction of the type ADD R1, R2, R3 what the predicated instruction does and under what situation and what additional information is needed. (**2 points)**

4D) Consider a direct-mapped cache with 8 blocks and the following sequence of block accesses:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 0, 1, 2, 4, 5, 6, 7, 8

Determine the number of cold, capacity and conflict misses in the cache. Assume OPT for determination of capacity misses (**4 points)**

ASSIGNMENT 5

5A) Consider a multicore system comprising a number of processors (cores) on a chip that are connected to a single-level private cache. X_i=R_i and X_i=W_i, mean a read and a write request to the *same* address X from processor *i,* respectively, where W_i=*C* means that the value C is written by processor *i*. Now consider the following access sequence assuming that X is not present in any cache from the beginning and that X originally contains the value 0:

$W_{1=}1$
R_1
R_2
$W_{1=}2$
R_2

What is returned by each read operation and why assuming i) a write-back write policy and ii) a write-through cache policy. (4 points)

5B) How does the MSI cache protocol guarantee that the correct value is returned? (4 points)

5C) Multithreading refers to a general technique to switch to another thread when a high latency operation is encountered. Explain what actions are taken with coarse-grain multithreading technique when a cache miss is encountered. (**4 points).**

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

Solutions to the exam in DAT105/DIT 051 2020-10-26

ASSIGNMENT 1

A)

Geom mean (SP) = $(SP_{P1} \text{ x } SP_{P2} \text{ x } SP_{P3})^{1/3} \Leftrightarrow SP_{P3} = SP^3/(SP_{P1} \text{ x } SP_{P2})$. Hence,
For A SP_{P3}=5 and for B SP_{P3}=2. Hence for A T_{P3}=2s and for B T_{P3}=5s

B)

For A $SP_{average} = (2 + 2 + 5)/3 = 3$ and for B $SP_{average} = (4 + 3 + 2)/3 = 3$. The reason that B is not the fastest is ouliers get higher weight in arithmetic means than in geometric means and P3 is an outlier on B (5s) substantially slower than on A (2s).

C)

According to Amdahl's Law $SP_{max} = 1/(1-f)$ where f is the fraction that can get parallelized. Hence, f = 0.99. SP(50) =1/(1-f + f/50) =1/(0.01+ 0.99/50) = 33.6

D)

T = IC x CPI/f disregarding cache misses. Hence, T = 10^8 x 5 x 10^{-9} = 0.5 s. But the execution time is 2s so 1.5/2 = 75% of the time is spent servicing cache misses.

ASSIGNMENT 2

2A)

	IF	ID	EX	FP1	FP2	FP3	FP4	FP5	MEM	WB
C1	I1									
C2	I2	I1								
C3	I3	I2		I1						
C4		I2			I1					
C5		I2				I1				
C6		I2					I1			
C7		I2						I1		
C8		I3		I2					I1	
C9				I3	I2					I1
C10					I3	I2				
C11						I3	I2			
C12							I3	I2		
C13								I3	I2	
C14									I3	I2

In the case it is pipelined, the instructions will proceed through the pipeline as illustrated. In case it is not pipelined, there will be five more cycles, i.e. 14 and 19 cycles respectively.

2B)

	IF	ID	EX	FP1	FP2	FP3	FP4	FP5	MEM	WB
C1	I1									
C2	I2	I1								
C3	I3	I2		I1						
C4		I3	I2		I1					
C5		I2	I3			I1			I2	
C6		I2					I1		I3	I2
C7		I2						I1		I3

C8		**I3**		**I2**					**I1**	

I3 reaches the MEM stage in C6 regardless whether the floating-point execution unit is pipelined or not.

2C)

A few observations:

- All instructions proceed through the FP pipeline which has the same number of pipeline stages as before
- FP instructions will only see one more stage (IF2) until it reaches ME1

Hence, counted in clock cycles the number of cycles for the fully pipelined FP unit is 14+1 = 15 and for the not pipelined FP unit 19+1 = 20.

However, the time to execute the instruction sequence is going down almost by a factor of two due to twice as high clock frequency.

2D)

```
LOOP: LD F0,0(R1)
      LD  F1,8(R1)
      ADD F4,F0,F0
      ADD F5,F1,F1
      ADDI R1,R1,#16
      SUBI R3,R3,#2
      BNEZ R3,LOOP
      SD F4,-16(R1)
      SD F5,-8(R1)
```

It is sufficient to unroll twice if we also assume a delayed branch and schedule the two store instructions in the two branch delay slots.

ASSIGNMENT 3

3A)

The operands that are available from the registerfile: Only F0 as only I1 has committed.
The operands that are available from the ROB: None because F1 is pending by I4 so ROB 17 will be thrown away

Filled out RAT:

RAT

Register	Tag	Status
R1	21	Pending
R2	22	Pending
R3	23	Pending
F0	--	Commit.
F1	19	Pending
F2	18	Pending
F3	--	Commit
F4	--	Commit

3C)

The branch will be removed from the ROB and all instructions after the branch up until the next branch will be removed one by one as well.

3D)

There will be two mispredictions initially and then one misprediction when the loop is exited. So in total three mispredictions out of hundred so 3%.

ASSIGNMENT 4

4A)

i) **FALSE:** It blocks if there is an outstanding cache miss
Rationale: Because not blocking is the purpose of a non-blocking cache

ii) **FALSE:** A miss-status-holding register is allocated only when a primary miss is encountered
Rationale: If a miss-status holding register was not allocated for a secondary miss, it would be blocking on an access to another word in the same block followed by a miss which would defeat the purpose of a non-blocking cache

iii) **TRUE:** A miss-status-holding register is needed for both primary and secondary misses

iv) **TRUE:** Prefetching requires a cache to be lockup-free to be effective

4B)

```
       LW R1, 0(R2)
      ADDI R4,R1,#8
LAB:  SW R4,0(R2)
      BNEQ R5,R6, LAB2
      ….
LAB2: LW R4,0(R2)
       J LAB
```

The number of instructions in the original trace is six whereas in the optimized trace there are only three instructions.

4C)

CADD R1,R2,R3,R4

Here, R4 is the condition register so if the condition is Zero, the add instruction will be performed if and only if R4=0; otherwise it will not do anything at all.

4D) Consider a direct-mapped cache with 8 blocks and the following sequence of block accesses:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 0, 1, 2, 4, 5, 6, 7, 8

Number of cold misses (the number of unique blocks): 11 misses

Capacity misses: The first eight misses will fill up the cache. Then
8 Miss replaces 3 that will not be accessed
9 Miss and replaces 8 which will be accessed furthest into the future
10 Miss and replaces 7
0 Hit
1 Hit
2 Hit
4 Hit
5 Hit
6 Hit
7 Miss and replaces 0 for example
8 Hit

Number of misses: 12 misses so 1 capacity miss.

Conflict misses
First eight accesses will Miss and will fill up the cache
8 Miss and replaces 0
9 Miss and replaces 1
10 Miss and replaces 2
0 Miss and replaces 8
1 Miss and replaces 9
2 Miss and replaces 10
4 Hit
5 Hit
6 Hit
7 Hit
8 Miss and replaces 0

Total number of misses: 14 misses. Hence number of conflict misses = 14 – 11 – 1 = 2

ASSIGNMENT 5

5A)

	Write-back	Write-through
W1=1		
R1	1	1
R2	0 (wrong)	1
W1=2		
R2	0 (wrong)	1 (wrong)

5B)

The MSI cache protocol will issue a BusUpgrade on every write command that invalidates any potential stale cache block copies in other caches to force them to read from the cache with the up to date copy or from memory if memory has it.

5C)

In coarse-grain multithreading (or blocked multithreading) a thread switch happens when encountering a long-latency operation such as a cache miss. Since this is detected late in the pipeline, all instructions in earlier pipeline stages will have to be flushed leading to quite severe overhead which must be offset by the gain of the long-latency operation that forces the thread switch.