CHALMERS

2024-10-09

Exam in DAT 105 (DIT 051) Computer Architecture

Time: October 28, 2024, 14-18 Johanneberg

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

Supporting material/tools: Chalmers approved calculator

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

GOOD LUCK!
Per Stenström

CHALMERS UNIVERSITY OF TECHNOLOGY *DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING*
412 96 Göteborg
Visiting address: Rännvägen 5
Phone: 031-772 1761
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

You are supposed to compare the performance of two computers, A and B, with data regarding three programs P1, P2 and P3.

The following data has been collected.

	P1[s]	P2[s]	P3[s]
Machine A	2	2	X
Machine B	4	4	Y
Machine R	64	64	64

A) The execution times of three programs P1-P3 on Machine A, B and R are given in the table, except for the ones for P3 on A and B. Assume that the geometric mean of speedups for Machine A and B over R are 16 and 12.7, respectively. **What is the execution time of P3 on A (X) and B (Y) and which machine is the fastest given the geometric mean? (3 points)**

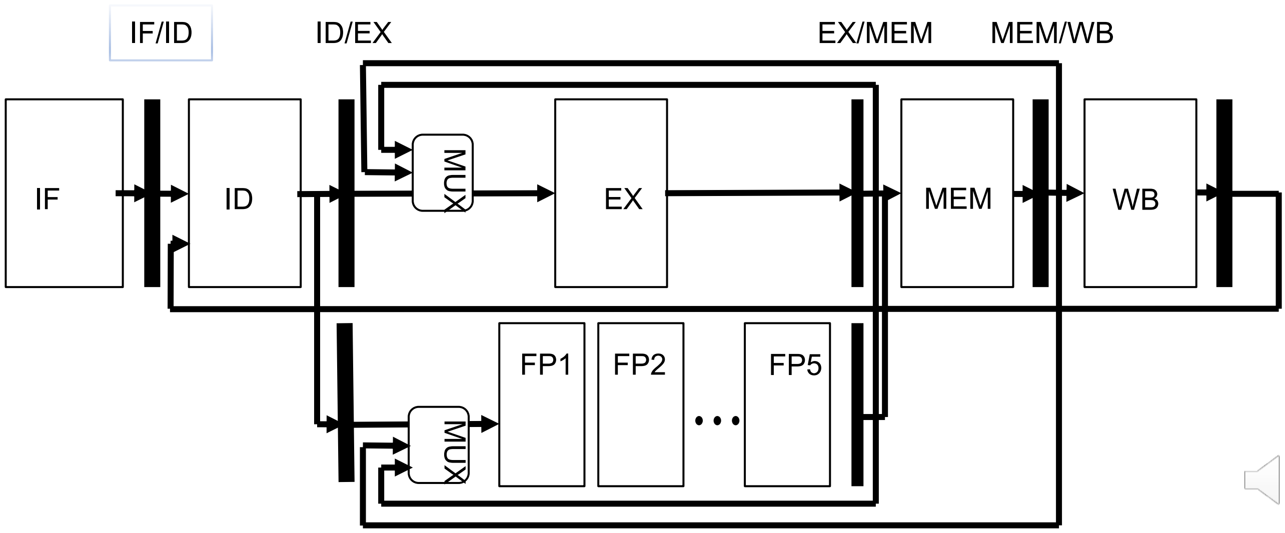
B) **Calculate the arithmetic mean of execution time. Why is A not the fastest when using the arithmetic mean of execution time? (3 points)**

C) A parallel program can enjoy a speedup of maximally 200. **How big fraction of the execution time, when the program is run on a single processor, can be parallelized and what is the speedup on 100 processors? (3 points)**

D) Let's assume that the number of instruction cache accesses for program P1 on A is 10^9 and that CPI is 2 if all cache misses are disregarded and that A is clocked at 2 GHz. **How big a fraction of the execution time is spent on servicing cache misses? (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 back to the input of the execution stage.

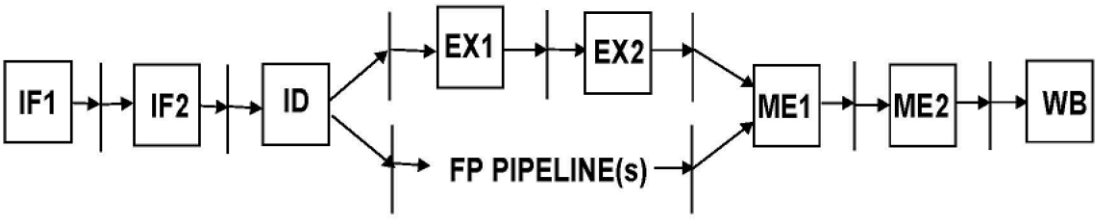


2A) Consider the following instruction sequence:

I1: ADD F3,F1,F2
I2: ADD F6,F4,F5
I3: ADD F8,F6,F7

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)
Unlike the pipeline in Assignment 2A), in the pipeline below the IF, EX and ME stages have been divided into two pipeline stages to allow for higher clock frequency. The floating-point functional unit comprises five stages as in 2A) and is fully pipelined.

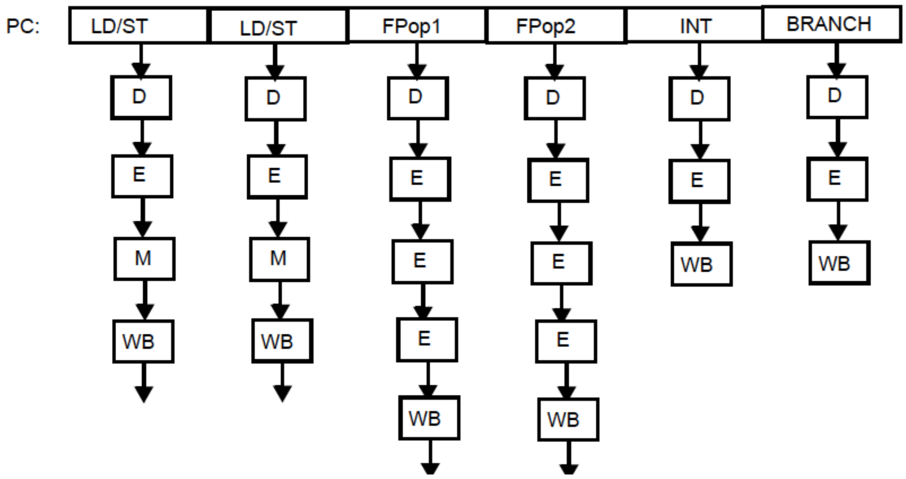


Consider the following program:

```
I1: LD.D F4, 0(R1)
I2: LD.D F5, 0(R2)
I3: ADD F6,F4,F5
I4: ADD F9,F8,F7
I5: ADDI R1,R1,#8
I6: ADDI R2,R2,#8
```

Determine the number of cycles it takes from I1 is issued from the ID-stage until I6 enters the ME1-stage. (6 points)

2C)



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.

Consider the following code:

```
for (i = 0; i < N; i++)
    {A[i+2] = A[i] + 2;
     B[i] = A[i+2] + 1;}
```

This translates into

```
Loop:  L.S F0, 0(R2)        ; –O1
       ADD.S F3, F0, F1     ; –O2
       ADD.S F4, F3, F1     ; –O3
       S.S. F3, 16(R2)      ; –O4
       S.S. F4, 0(R3)       ; –O5
       ADDI R2,R2, 8
       ADDI R3,R3, 8
       BNE R2,R4, Loop
```

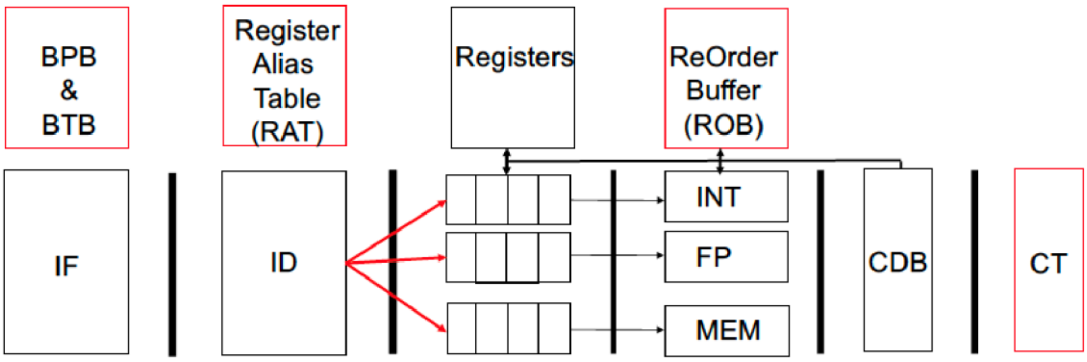
Show the code for the kernel in a software pipelined version of the code that maximizes instruction level parallelism. (3 points)

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 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).



3A)

Consider the following content of the Register Alias Table (RAT):

Register	Tag	Status
R0	--	Committed
R1	--	Committed
R2	ROB 0	Completed
…	…	…
R31	ROB 1	Completed
F0	ROB 10	Pending
F1	ROB 11	Pending
F2	--	Committed
…	…	…
F31	--	Committed

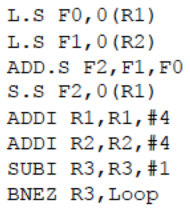
i)

i) What does the status Committed, Completed and Pending mean?

ii) Where can the values of registers R0, R1 and R2 be found given the content of the RAT? (4 points)

3B)

Consider the following example program:



```
L.S F0,0(R1)
L.S F1,0(R2)
ADD.S F2,F1,F0
S.S F2,0(R1)
ADDI R1,R1,#4
ADDI R2,R2,#4
SUBI R3,R3,#1
BNEZ R3,Loop
```

Let's pay attention to how the Tomasulo algorithm resolves the RAW hazards between the two load (L.S) instructions and the add (ADD.S) instruction. Explain in detail the actions taken when the load and the add instructions access the RAT table and how that resolves the RAW hazard. You must explain in detail what information is bookkept in the RAT when the add instruction is inserted in the reservation station in front of the FP unit. (**4 points)**

3C) Assume that the branch predictor is initialized to Not-Taken and that the loop in 3B) is executed 1000 times. What is the fraction of correct branch predictions in percent? (**4 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 all miss-status-holding registers are allocated and a new miss is encountered

ii) A miss-status-holding register is allocated only when a secondary miss is encountered

iii) A miss-status-holding register is allocated for both primary and secondary misses

iv) A secondary miss will generate a miss request sent to memory

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

4B)
The trace through basic blocks A, B and C (below) is found to be more common than through basic blocks A, D and C. Schedule and optimize the trace to minimize the number of instructions in the most common trace and also provide fix-up code in case the less likely trace of block A, D and C is executed. (**4 points**)

```
        LW R4,0(R1)
        ADDI R6,R4,#1
/* block A*/
        BEQ R5,R4,LAB
        LW R6,0(R2)
/*block B*/
/*block D* empty/
  LAB: SW R6,0(R1)
/*block C*/
…………
```

4C) Explain, by applying the concept of predication to a load instruction of the type LW R*i*, DISP(R*j*), what the predicated instruction does and under what situation and what additional information is needed in the instruction. (**2 points)**

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

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

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 with a write-back policy. $X_i=R_i$ 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:

$W_1=5$
$W_1=6$
R_1
R_2
$W_1=2$
R_2

i) What is returned by each read operation?

ii) What values should be returned by the load instructions for a memory system that is coherent?

iii) Which of the load instructions return the correct value and which do not? (6 points)

5B) Consider the same sequence of reads and writes as in 5A). **What bus transactions will be triggered by an MSI protocol to maintain cache coherence. (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 the coarse-grain (or blocked) multithreading technique when a cache miss is encountered**. **(2 points).**

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

Solutions to the exam in DAT105/DIT 051 2024-10-28

ASSIGNMENT 1

The following data has been collected.

A)

Geom mean (SP) = $(SP_{P1} \times SP_{P2} \times SP_{P3})^{1/3} \Leftrightarrow SP_{P3} = SP^3/(SP_{P1} \times SP_{P2})$. Hence, For A SP_{P3}=4 and for B SP_{P3}=8. Hence for A T_{P3}=16s and for B T_{P3}=8s

B)

For A Arith mean = (2 + 2 + 16)/3 = 6.7 and for B Arith mean = (4 + 4 + 8)/3 = 5.3. The reason that B is the fastest is outliers get higher weight in arithmetic means than in geometric means and P3 is an outlier on A (16s) substantially slower than on B (8s).

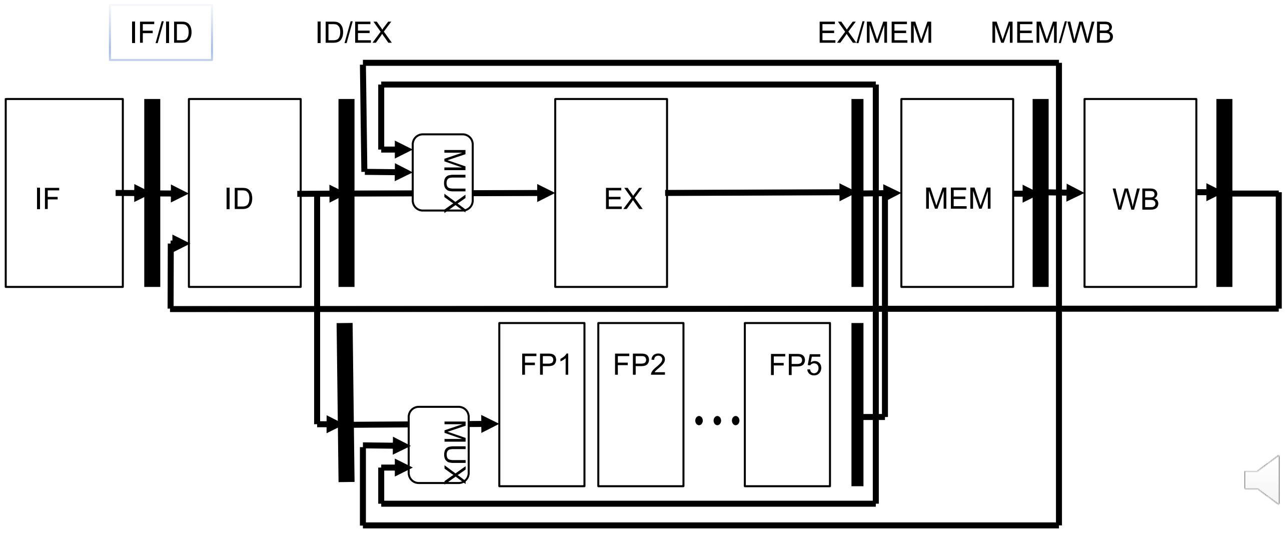
C)

According to Amdahl's Law $SP_{max} = 1/(1\text{-}f)$ where f is the fraction that can get parallelized. Hence, f 1-1/200 = 99.5% . SP(100) =1/(1-f + f/100) =1/(0.005+ 0.995/200) ~ 100.

D) T = IC x CPI/f disregarding instruction cache misses. Hence, T = 10^9 x 2 x 0.5 x 10^{-9} = 1 s. But the execution time is 2s so 1/2 = 50% of the time is spent servicing cache misses.

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 back to the input of the execution stage.



2A) Consider the following instruction sequence:

I1: ADD F3,F1,F2
I2: ADD F6,F4,F5
I3: ADD F8,F6,F7

No pipelining of floating-point unit.

When the floating-point unit is not pipelined, the three instructions will be serialized through the fowing point unit:

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

	C 15	C16	C17	C18
I1				
I2				
I3	FP3	FP4	FP5	MEM

I3 will enter the MEM stage in clock 18.

With pipelined floating-point unit.

There is no RAW hazard between any of the three instructions so they can be fully pipelined through the floating-point unit:

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
I1	IF	ID	FP1	FP2	FP3	FP4	FP5	MEM	WB						
I2		IF	ID	FP1	FP2	FP3	FP4	FP5	MEM	WB					
I3			IF	ID	X	X	X	X	FP1	FP2	FP3	FP4	FP5	MEM	WB

Here, I3 enters the MEM stage in clock 14.

2B)

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14
I1	IF1	IF2	ID	EX1	EX2	ME1	ME2	WB						
I2		IF1	IF2	ID	EX1	EX2	ME1	ME2	WB					
I3			IF1	IF2	ID	ID	ID	ID	FP1	FP2	FP3	FP4	FP5	ME1

	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17

I3	IF2	ID	ID	ID	ID	FP1	FP2	FP3	FP4	FP5	ME1	ME2	WB	
I4	IF1	IF2	IF2	IF2	IF2	ID	FP1	FP2	FP3	FP4	FP5	ME1	ME2	WB
I5		IF1	IF1	IF1	IF1	IF2	ID	EX1	EX2	ME1	ME2	WB		
I6						IF1	IF2	ID	EX1	EX2	ME1	ME2	WB	

I6 will reach the ME1 stage in clock 14.

2C)

See schedule in Table 3.26 on page 150 in the textbook.

ASSIGNMENT 3

3A)

i) Committed means that the instruction has been executed and is not speculative anymore. This means that its destination operand is in the registerfile. Status Completed means that the instruction has been executed but the instruction is still in speculative mode. Hence, the value of the register is in the ROB. Finally, pending means that the instruction has been issued but has not completed. It has an entry in the ROB but the value is not available.

ii) R0 and R1 have status Committed and hence their values are available in the registerfile. R2 has status Completed and its value is in ROB entry 0.

3B)

When the two load instructions come to the ID stage, they will be inserted in the Reorder Buffer and receives a tag corresponding to in what entry they will be inserted. The destination registers (F0 and F1) will be also marked in the RAT as pending and the ROB entry numbers are recorded there as well. When the ADD arrives in the ID stage it will be recorded in the ROB as well and it will try to access the source operands (F0 and F1) from the RAT. As they are pending, the ROB entry numbers will be used. These ROB numbers will be broadcast on the CDB when the loads are completed. The ADD will then grab operand data corresponding to them. As the loads are speculatively executed, the results will not be written back to the registers. Instead, they will be temporarily stored in the ROB until the loads are committed.

3C)

Assuming that the 2-bit predictor is in the **Non-taken** state initially. At entry, it will take two mispredictions until the predictor correctly predicts the branch as taken. In the **Taken** state. At exit, the branch predictor will mispredict. Hence, there are three mispredictions and the prediction accuracy is (1000-3)/1000 x 100% = 99.7%

ASSIGNMENT 4

4A)

i) It blocks if all miss-status-holding registers are allocated and a new miss is encountered. **TRUE**

ii) A miss-status-holding register is allocated only when a secondary miss is encountered. **FALSE,** it is allocated for primary as well as secondary misses.

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

iv) A secondary miss will generate a miss request sent to memory. **FALSE.** Only primary misses will generate miss requests.

4B)

```
        LW R4,0(R1)
        ADDI R6,R4,#1
/* block A*/
        BEQ R5,R4,LAB
        LW R6,0(R2)
/*block B*/
/*block D* empty/
  LAB:  SW R6,0(R1)
```

```
        LW R4,0(R1)
...........LW R6, (R2)
        BEQ R5,R4,LAB1
LAB2:   SW R6,0(R2)
…..
LAB1:   ADDI R6,R4,#1
        J LAB2
```

Trace including block A and B: 4
Trace including block A and D: 6

4C)

CLW R*k*, R*i*, DISP(R*j*)

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

4D)
Number of cold misses (the number of unique blocks): 10 misses

Capacity misses:

```
Victim.              1 4         2 3   0 6
Block access 0 1 2 3 4 5 2 3 6 7 0 9 10
```

Total number of misses: 10
Capacity misses: 10-10 = 0

Conflict misses:

```
Victim               0 1         2 3 4 5 6
Block access 0 1 2 3 4 5 2 3 6 7 0 9 10
```

Total misses: 11
Conflict misses = Total misses – Capacity misses – Cold misses = 11-0-10 = 1

ASSIGNMENT 5

5A)

i)

W_1=5
W_1=6
R_1 Returns 6 because this is the value returned from P1's cache
R_2 Returns 0 because this is the value returned from memory servicing the miss.
W_1=2
R_2 Returns 0 because this is the value returned from P2's cache.

ii)

W_1=5
W_1=6
R_1 Returns 6 which is expected
R_2 Returns 0 but should return 6 which is the value of the latest write.
W_1=2
R_2 Returns 0 but should return 2 which is the value of the latest write.

iii)
Only the first one.

5B)

W_1=5 **BusRdx**
W_1=6 No bus transaction
R_1 No bus transaction
R_2 **BusRd**
W_1=6 **BusUpgrade**
R_2 **BusRd**

5C)

When a high-latency operation such as a cache miss is encountered, all instructions in the pipeline that are stages before the stage where the high-latency operation is detected will be flushed. A switch to a new thread will then happen and instructions from that thread will be fetched.