High Performance Computer Architecture (CS60003)

Tutorial 3

Indian Institute of Technology Kharagpur

Exercises

1. Assume, **1.** there is 1 Reservation Station(1 entry per reservation station) and 1 Operational Unit for each of ADD/SUB operation and MUL/DIV operation; **2.** a Re-order Buffer is in place; **3.** 1 broadcast in bus is allowed per cycle; **4.** issue and commit should happen in-order (following the code sequence); **5.** the Reservation Station is considered to be freed on Write back; **6.** Addition takes 1 cycle, Multiplication takes 2 cycles and Division takes 4 cycles to finish. Fill the table with correct cycle of occurrence.

Instructions	Issue	Execute	Write	Commit
DIV R2 R3 R4	1	2		
MUL R1 R5 R6				
ADD R3 R7 R8				
MUL R1 R1 R2				
SUB R4 R2 R5				
ADD R1 R4 R3				

2. Consider the following set of instructions. Each `ADD/SUB` instruction takes 1 cycle and `MUL/DIV` takes 4 cycles to complete.

I_1: `ADD R2, R2, R1`
I_2: `SUB R1, R1, R3`
I_3: `ADD R3, R4, R5`
I_4: `DIV R3, R2, R3`
I_5: `ADD R1, R4, R4`
I_6: `ADD R3, R2, R2`

Suppose a 'divide-by-zero' exception occurs for the instruction I_4. The processor learns about the exception at the end of the 3^{rd} execution cycle of I_4. Complete the below table till 10^{th} clock cycle. Assume there are two independent execution units for `ADD/SUB` and `MUL/DIV` with a 3-entry reservation station per unit. The Reorder Buffer can hold upto 10 instructions. The instructions are issued and committed in order.

Cycle	Issue	Execute	Write	Commit
1	I_1			
2	I_2	I_1		
3	I_3	I_2	I_1	
..				
10				

3. (a) A five instruction sequence executes according to Tomasulo's algorithm. Each instruction is either of the two forms as mentioned below:

```
ADD <Dest Reg>, <Src Reg 1>, <Src Reg 1>
MUL <Dest Reg>, <Src Reg 1>, <Src Reg 1>
```

ADDs are pipelined and takes two cycles for execution, i.e. E1-E2. MULs are also pipelined and take 5 cycles for execution (E1-E5). There are only one ADD and one MUL unit present in the processor. An instruction must wait until its source registers are ready. For instance, if instruction 2 has a read-after-write dependence on instruction 1, instruction 2 can start executing in the next cycle after instruction 1 writes back. An example is shown below, where F, D, WB are fetch, decode and write-back respectively.

```
instruction 1  |F|D|E1|E2|WB|
instruction 2    |F|D |- |- |E1|......|WB|
```

The machine can fetch one instruction per cycle, and can decode one instruction per cycle. Consider the five-instruction per cycle. Consider the five-instruction program as follows.

```
ADD R7,R6,R7
ADD R3,R6,R7
MUL R0,R3,R6
MUL R2,R6,R6
ADD R2,R0,R2
```

The initial contents of the register file are provided below:

R0	R1	R2	R3	R4	R5	R6	R7
4	5	6	7	8	9	10	11

i. In each cycle, a single instruction is fetched and a single instruction is decoded. There is no same-cycle ISSUE->DISPATCH or CAPTURE->DISPATCH operations. Assume the reservation stations are all initially empty, which are shown below. Put each instruction into the next available reservation station. For example, the first ADD goes into RS0 and the first MUL goes into RS3. Instructions remain in the reservation stations until they are completed. Show the state of the reservation stations at each cycle till the end of cycle 10. [**Note:** when allocating source registers to reservation stations, please always have the higher numbered register be assigned to Source 2.]

	ADD	
RS	**Source 1**	**Source 2**
RS0		
RS1		
RS2		

	MUL	
RS	**Source 1**	**Source 2**
RS3		
RS4		
RS5		

ii. Show the state of the Register Alias Table at the end of cycle 10.

(b) Consider a program which consists of the following two instructions:

```
0x0040:  LD    R0, [0xABCD]
0x0044:  ADD   R1, R2, R3
```

Answer the following questions with the assumptions given below.

- An LD takes 3 cycles to execute and an ADD takes 1 cycle to execute.
- The LD and ADD can write their results to the Reorder Buffer (ROB) on the same cycle they are ready.
- It takes one cycle for the ROB to write data ready to be committed to architectural state to the architectural register file.
- The value at address 0xABCD is 1.

i. Compute the contents of the architectural register file and the ROB for each of the next six cycles for a processor which can issue 1 instruction per cycle. The initial state of the register file and ROB are given below.

Register File			
R0	R1	R2	R3
5	4	3	2

Reorder Buffer		
Result Register	Result Value	Done
–	—	–
–	—	–

ii. Assume that on a system, without an ROB, an exception occurs when `ADD` is executed. What will be the content of architectural register file at the time of exception? Comment if there is any issue with this scenario

4. Remember that Tomasulo's algorithm requires tag broadcast and comparison to enable wake-up of dependent instructions. In this question, we will calculate the number of tag comparators and size of tag storage required to implement Tomasulo's algorithm in a machine that has the following properties:

   - 8 functional units where each functional unit has a dedicated separate tag and data broadcast bus
   - 32 64-bit architectural registers
   - 16 reservation stations per functional unit
   - Each reservation station entry can have two source registers

   Answer the following questions.

   (a) What is the total number of tag comparators per reservation station entry?

   (b) What is the total number of tag comparators in the entire machine?

   (c) What is the (minimum possible) size of the tag?

   (d) What is the (minimum possible) size of the Register Alias Table (RAT) in bits? Assume there is a 'valid' bit for each entry in RAT.

   (e) What is the total (minimum possible) size of the tag storage in the entire machine in bits?

5. Suppose an out-of-order dispatch machine implementing Tomasulo's algorithm with the following characteristics:

   - The processor is fully pipelined with four stages: Fetch, Decode, Execute and Writeback.
   - The processor implements `ADD` and `MUL` instructions only.
   - For all instructions, fetch takes 1 cycle, decode takes 1 cycles and writeback takes 2 cycles.
   - For `ADD` instructions, execute takes 1 cycle. For `MUL` instructions, execute takes 3 cycles.
   - There are enough functional units to have 3 of each instructions executing at the same time.
   - A reservation station entry and a reorder buffer entry are allocated for an instruction at the end of its decode stage.
   - An instruction broadcasts its results at the end of its first writeback cycle. This updates the values in the reservation stations, RAT, and reorder buffer.
   - Writebacks of different instructions can happen concurrently.

   Suppose the pipeline is initially empty and the machine fetches exactly 5 instructions. 7 cycles after fetching the first instruction, the reservation stations and RAT are as follows:

ADD Reservation Station					
Tag	Busy	V_j	V_k	Q_j	Q_k
A	1	3	2	–	–
B	1	9	9	–	–
C	1	5	9	–	–

MUL Reservation Station					
Tag	Busy	V_j	V_k	Q_j	Q_k
X	1	2	–	5	A
Y	1	12	2	B	–
Z	0	–	–	–	–

	Valid	Tag	Data
R0	1	–	5
R1	1	–	2
R2	1	–	3
R3	0	C	12
R4	0	Y	1
R5	1	–	9
R6	1	–	0
R7	1	–	21

In the reservation stations, the youngest instruction is at the bottom and some instructions might have already been completed.

(a) What are the 5 instructions? Write them in program order. No credit will be given if the correct program order is not maintained.

(b) Now suppose this machine always raises a floating point exception after the last execute stage of MUL instructions (the engineer who designed the multiplication unit did not verify his design.) Please show the contents of the Reorder Buffer (ROB) (sample given below) right after the first floating point exception is raised. Oldest instruction should be on top and youngest at the bottom. After an instruction is retired, its reorder buffer entry should be marked as invalid. Use '?' to indicate any unknown information at that moment. Note that the ROB contains additional fields which needs to be filled up.

Valid	Tag	Opcode	Reg	Data Value	Value Ready?	Exception?