IIT Guwahati - Department of Computer Science & Engineering

CS 223- Computer Organization & Architecture

Practice Tutorial Sheet #2 [04.05.2020] – Solution Set

1. In a dynamically issued speculative superscalar processor the reservation station entries of two functional units (Mul and Add) at clock cycle T is as shown below. At T, none of the instructions waiting in the reservation station have started execution.

Name	Busy	Op	Vj	Vk	Qj	Qk	Dest	A
Mul	Yes	FMUL	8	5	0	#4	#5	40
Mul	Yes	FDIV	2	8	#6	0	#7	20

Name	Busy	Op	Vj	Vk	Qj	Qk	Dest	A
Add	Yes	FADD	8	5	#4	#5	#6	10

(A) Write any possible sequence of three instructions (in the order of issue) along with their corresponding register operands that could result in the reservation station entries shown above. Only F1, F2 and F3 can be used as the operands for the instructions.
Instruction syntax should be of the form: *Opcode DestReg, SrcReg1, SrcReg2*.

(B) State other mandatory conditions (if any) that should hold at the time of issue of these instructions to ensure the correctness of the above entries.

Solution: [**Note: Based on the assumptions made, there can be different arrangements of registers in the instructions.]

Name	**Busy**	**Op**	**Vj**	**Vk**	**Qj**	**Qk**	**Dest**	**A**
Mul	Yes	FMUL	8	5	**0**	#4	#5	40
Mul	Yes	FDIV	2	8	#6	**0**	#7	20
Add	Yes	FADD	8	5	#4	#5	#6	10

Dest field shows how the entry in the ReOrder Buffer (ROB).

Therefore, the order of instruction in program order is as follows. Instructions are getting ROB entry during issue and issue stage is in order.

N^{th} instruction = FMUL // assume 5^{th} instruction

$(N+1)^{th}$ instruction = FADD // assume 6^{th} instruction

$(N+2)^{th}$ instruction = FDIV // assume 7^{th} instruction

Name	**Busy**	**Op**	**Vj**	**Vk**	**Qj**	**Qk**	**Dest**	**A**
Mul	Yes	FMUL	8	5	**0**	#4	#5	40
Add	Yes	FADD	8	5	#4	#5	#6	10
Mul	Yes	FDIV	2	8	#6	**0**	#7	20

Given, in the question that only 3 registers ***F1, F2, and F3*** can be used as operands

and the instruction format is: ***Opcode DestReg, SrcReg1, SrcReg2.***

Qj = 0 in reservation station indicates that the value is ready.

For FMUL : 1st Src operand is ready and the value = 8. Let us assume that F1 is the register that holds the value = 8; 2nd Src operand is dependent on the 4th instruction in the ROB. Let the register be F2.

FMUL F3, F1 , F2; where F1 = 8;

For FADD: : 1st Src operand is dependent on 4th instruction. Hence, F2 is 1st operand;

2nd Src operand is dependent on the 5th instruction in the ROB i.e. FMUL . Hence, F3.

FADD F3, F2 , F3; // only F1, F2 and F3 be used. F1 is not used as it is used in next instruction.

For FDIV: : 1st Src operand is dependent on 6th instruction i.e. FADD Hence, F3 is 1st operand; 2nd Src operand is already ready and the same register holds the value = 8. Hence, F1.

FDIV F2, F3 , F1; where F1 = 8;

Therefore, the instructions are:

FMUL F3 , F1 , F2;

FADD F3 , F2 , F3;

FDIV F2 , F3 , F1;

*** the marked ones in same color should be same. Any combination of instructions satisfying these conditions are also correct.**

B) Mandatory Conditions to hold:

1) The value of F1 should not change because FDIV uses the same value as one of the operands.

2) As per this example: F2 is already present in the ROB (4th instruction).

2. Consider a dynamically scheduled single issue instruction pipeline with speculation using a branch predictor that always predicts the branch as taken. The pipeline has one FP-multiplier that takes 4 cycles for execution, one FP-adder that takes 2 cycles for execution, one integer unit that takes 1 cycle for execution and one memory address unit that takes 1 cycle to compute effective address for load and store operations. All the integer and FP units are fully pipelined and support operand forwarding. Memory access for a load operation takes place in the next cycle after the address computation. Store operation access memory only upon commit. This speculative pipeline is implemented with a 4-entry ROB. Assume that instruction issue will not be stalled due to availability of reservation stations. Consider the following code with 9 instructions (I1-I9).

Initial values of registers are as follows: F1← a, F2← b, F3← c, R2← 200, R3← 240

```
I1-lxx: L.D F4, 0(R2)        ; load X
I2-     MUL.D F5, F4, F4
I3-     MUL.D F5, F5, F1
I4-     MUL.D F4, F4, F2
I5-     ADD.D F6, F5, F4
I6-     ADD.D F6, F6, F3
I7-     S.D F6, 48(R2)       ; store Y
I8-     ADDI R2, R2, #8
I9-     BNE R2,R3, lxx       ; branch to lxx if R2!=R3
```

Fill up the time of issue, execution, memory access, CDB write and commit for one iteration of the loop in the table given below. Answer the questions given below the table.

	Instruction	Issue	Begin Execution	Memory Access	CDB Write	Commit
I1	L.D F4, 0(R2)	1	2	3	4	5
I2	MUL.D F5, F4, F4	2	5		9	10
I3	MUL.D F5, F5, F1	3	10		14	15
I4	MUL.D F4, F4, F2	4	6		10	16
I5	ADD.D F6, F5, F4	6	15		17	18
I6	ADD.D F6, F6, F3	11	18		20	21
I7	S.D F6, 48(R2)	16	17	22		22
I8	ADDI R2, R2, #8	17	18		19	23
I9	BNE R2,R3, lxx	19	20		21	24

// shaded colors indicate the wait to issue an instruction based on ROB commit of instruction.

Only at most 4 outstanding instructions possible.

(A) In which clock cycle I4 write result to CDB? - 10

I2 and I4 cannot start at 5, hence I4 will wait till 6 to begin execution.

(B) In which clock cycle I6 is issued? – 11 (ROB not free till 10, commit of I4)

(C) In which clock cycle I7 access memory?- 22. (Issue at 16, Commit inorder)

(D) In which clock cycle I8 starts execution?-18 (Issued at 17)

(E) What is the relation between X and Y in terms of a, b and c?

$Y = aX^2+bX+c$

(F) How many times the loop will be iterated? – 5 times. (R2 moves from 200 to 240 with an increment of 8)