DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

Computer Organization and Architecture (CS2600)

Total Time : 50 minutes **Class Test 6 (Scalar and Superscalar Processors)**
Max Marks : 30 (10x3) **03-05-2024**

---

1. Consider the following loop that executes on a 5 stage (IF, ID, EX, ME, WB) simple scalar pipeline processor.

```
LOOP: ld x10, 0(x13)
      ld x11, 8(x13)
      add x12, x10, x11
      subi x13, x13, 16
      bnez x12, LOOP
```

Assume that all instructions and data are in the cache memory and accessing the cache takes just one clock cycle. Further, assume that perfect branch prediction is used (no stalls due to control hazards), that there are no delay slots, that the pipeline has full forwarding support, and that branches are resolved in the EX (as opposed to the ID) stage.

- Show a pipeline execution diagram for the first two iterations of this loop.
- Compute the average cycles per instruction for the loop. Hint. Do not consider the initial clock cycles during which the pipeline is filling.

```
ld x10, 0(x13)        IF ID EX ME WB
ld x11, 8(x13)           IF ID EX ME WB
add x12, x10, x11           IF ID .. EX ME WB (forwarded x11)
addi x13, x13, -16             IF .. ID EX ME WB
bnez x12, LOOP                    .. IF ID EX ME WB
ld x10, 0(x13)                       IF ID EX ME WB
ld x11, 8(x13)                          IF ID EX ME WB
add x12, x10, x11                          IF ID .. EX ME WB
addi x13, x13, -16                            IF .. ID EX ME WB
bnez x12, LOOP                                   .. IF ID EX ME WB
```

After the pipeline is full, we need 12 cycles to execute 10 instructions, therefore average CPI = 12/10 = 1.2.

2. Consider an if-else statement in a while loop, where for each iteration of the loop the condition turns out to be T, F, T, T, F, where T is True and F is False. Branch is taken when the condition turns out True.

For this sequence of branches, what is the accuracy of a branch predictor that **(a)** always predicts `Taken`, **(b)** always predicts `not-Taken`, **(c)** is a 2-bit predictor with initial state `Strongly Not-Taken (SNT)`. **(d)** If the while loop runs forever (while(1)), how would you design a predictor that would provide the highest accuracy in prediction? Your predictor should be a sequential circuit with one output that provides a prediction (1 for taken, 0 for not taken) and no inputs other than the clock and the control signal that indicates that the instruction is a conditional branch. **(e)** If the sequence of branch outcomes inverts (i.e. FTFFT instead of TFTTF), what will be the outcome of the branch predictor designed in (d). **(f)** If in an iteration of the while loop, the branch outcomes are either FTFFT or TFTTF (picked randomly), how would you modify the branch predictor of (d) to maximize accuracy.

(a) The sequence of correct (C) and incorrect (I) predictions are: C I C C I. Therefore accuracy is 3/5 = 60%.

(b) The sequence of correct (C) and incorrect (I) predictions are: I C I I C. Therefore accuracy is 2/5 = 40%.

(c) The sequence and state of the 2-bit predictor is as follows:
T: (SNT) NT T ST → Predict NT → I;
F: SNT (NT) T ST → Predict NT → C;
T: (SNT) NT T ST → Predict NT → I;

T: SNT (NT) T ST → Predict NT → I;
F: SNT NT (T) ST → Predict T → I
Accuracy is 1/5 = 20%.

(d) Have a shift-register that is initialized to TFTTF. For each conditional branch, choose T or F as per the branch predictor sequence.

(e) the accuracy will be 0.

(f) Detect the first branch in the loop. If it is T, then set the shift registers to TFTTF. If it is false, then set the shift registers to FTFFT.

3. Simulate the execution of the following code snippet in a superscalar processor. Show the contents of the reservation station entries, register file busy, operation, and data fields for each cycle (make a copy of the table shown below that you simulate). Indicate which instruction is executing in each functional unit in each cycle. Also indicate any result forwarding across a common data bus by circling the producer and consumer and connecting them with an arrow.

```
i: R4 <- R0 + R8
j: R2 <- R0 * R4
k: R4 <- R4 + R8
1: R8 <- R4 * R2
```

Assume dual dispatch and a dual common data bus (CDB). Add latency is two cycles, and multiply latency is three cycles. An instruction can begin execution in the next cycle that it is dispatched, assuming all dependences are satisfied.

	1	2	3	4	5	6	7	8	9	10	11	12
i: R4 <- R0 + R8	IF	ID	DIS	EX	EX	WB						
j: R2 <- R0 * R4	IF	ID	DIS	-	-	EX	EX	EX	WB			
k: R4 <- R4 + R8		IF	ID	DIS	-	EX	EX	WB				
l: R8 <- R4 * R2		IF	ID	DIS	-	-	-	-	EX	EX	EX	WB

CLK3	Busy	Op	Vj	Vk	Qj	Qk	A
ALU 1	Y	ADD	R0	R8			
ALU 2	N						
MUL/DIV 1	Y	MUL	R0			ALU 1	
MUL/DIV 2	N						

CLK4	Busy	Op	Vj	Vk	Qj	Qk	A
ALU 1	Y	ADD	R0	R8			
ALU 2	Y	ADD		R2	ALU 1		
MUL/DIV 1	Y	MUL	R0			ALU 1	
MUL/DIV 2	Y	MUL			ALU 1	MUL 1	

CLK6	Busy	Op	Vj	Vk	Qj	Qk	A
ALU 1	N						
ALU 2	Y	ADD	R4	R8			
MUL/DIV 1	Y	MUL	R0	R4			
MUL/DIV 2	Y	MUL	R4			MUL 1	

CLK8	Busy	Op	Vj	Vk	Qj	Qk	A
ALU 1	N						
ALU 2	N						
MUL/DIV 1	Y	MUL	R0	R4			
MUL/DIV 2	N	MUL	R4			MUL1	

CLK9	Busy	Op	Vj	Vk	Qj	Qk	A
ALU 1	N						
ALU 2	N						
MUL/DIV 1	N						
MUL/DIV 2	Y	MUL	R4	R2			

CLK12	Busy	Op	Vj	Vk	Qj	Qk	A
ALU 1	N						
ALU 2	N						
MUL/DIV 1	N						
MUL/DIV 2	N						