CS 2410

Computer Architecture

Spring 2022

Distributed: Feb 14th, 2022

Due: 11:59pm Feb 28th, 2022

Points: 100

1. (10 pts) Suppose a 5-stage pipeline has the following delays for each stage

IF	ID	EX	MEM	WB
250ps	350ps	150ps	300ps	200ps

What is the clock cycle time in a pipelined and non-pipelined processor? Further assuming there are no stalls, what is the speedup achieved by pipelining a single-cycle datapath?

non-pipelined (single cycle): 250+350+150+300+200=1250

Pipelined: 350

Speedup: 1250/350

2. (60 pts) Assume a 7 stage, scalar (1-wide) in-order MIPS pipeline with stages: FT, FA, D, E, MT, MA, W. [Aside: assuming that each memory stage has been split in two for serialized tag check and data access in the I and D caches]
Assume full forwarding networks. Indicate stalled instruction occupancy in a pipeline stage with a lower case letter (ft,fa,d,e,mt,ma,w). Assume that all hazard detection and stall insertion logic is in decode (D). Indicate, by drawing an arrow when a value is forwarded from one instruction to another in the cycle that the forwarding occurs. Assume that all loads and stores are hits and that there are no exceptions. Assume that there are zero branch delay slots and perfect branch prediction.

Consider the following sequence of instructions (you may assume them to be the inner body of a FOR loop with the initialization code elided) scheduled on the above pipeline, assuming the next dynamic instance of the instruction at label I will be a taken branch.

```
A:    lw    $2, 40 ($6)
B:    lw    $2, 0 ($2)
C:    lw    $3, 40 ($7)
```

D: lw $3, 0 ($3)
E: add $3, $3, $2
F: sw $3, 0 ($2)
G: addi $6, $6, -64
H: addi $7, $7, -64
I: beq $6, $5, A

(a) Build a table where there is one row for each instruction and one column for each cycle, starting with the tag check on the fetch of the instruction at label A. Fill in the table with the stage currently occupied by the instruction in that row in that cycle until the instruction beq instruction reaches the writeback stage.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A: lw $2, 40 ($6)	F T	F A	D	E	M T	M A	W														
B: lw $2, 0 ($2)		F T	F A	d	d	D	E	M T	M A	W											
C: lw $3, 40 ($7)			F T	f a	f a	F A	D	E	M T	M A	W										
D: lw $3, 0 ($3)				f t	f t	F T	F A	d	d	D	E	M T	M A	W							
E: add $3, $3, $2							F T	f a	f a	F A	d	d	D	E	M T	M A	W				
F: sw $3, 0 ($2)								f t	f t	F T	f a	f a	F A	D	E	M T	M A	W			
G: addi $6, $6, -64											f t	f t	F T	F A	D	E	M T	M A	W		
H: addi $7, $7, -64														F T	F A	D	E	M T	M A	W	
I: beq $6, $5, A															F T	F A	D	E	M T	M A	W

(b) Assuming that all registers other than 0 (fixed), 1 (reserved), 5 (input), 6 (input), and 7(input) and 31(reserved) are freely available for use and that the above sequence can be unrolled by 2x, unroll the above loop and schedule 2 iterations on the same pipeline. Build a similar table ~~(compared to Question 6 in HW1)~~ where there is one row for each dynamic instruction and one column for each cycle. Fill in the table with the stage currently occupied by the instruction in that row in that cycle. What is the cycles/iteration?

CYCLE #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
A1: lw $2, 40 ($6)	F T	F A	D	E	M T	M A	W															
A2: lw $8, -24 ($6)		F T	F A	D	E	M T	M A	W														
G: addi $6, $6, -128			F T	F A	D	E	M T	M A	W													
B1: lw $2, 0 ($2)				F T	F A	D	E	M T	M A	W												
B2: lw $8, 0 ($8)					F T	F A	D	E	M T	M A	W											
C1: lw $3, 40 ($7)						F T	F A	D	E	M T	M A	W										
C2: lw $9, -24 ($7)							F T	F A	D	E	M T	M A	W									
H: addi $7, $7, -128								F T	F A	D	E	M T	M A	W								
D1: lw $3, 0 ($3)									F T	F A	D	E	M T	M A	W							
D2: lw $9, 0 ($9)										F T	F A	D	E	M T	M A	W						
E1: add $3, $3, $2											F T	F A	d	D	E	M T	M A	W				
E2: add $9, $9, $8												F T	f a	F A	D	E	M T	M A	W			
F1: sw $3, 0 ($2)													f t	F T	F A	D	E	M T	M A	W		
F2: sw $9, 0 ($8)															F T	F A	D	E	M T	M A	W	

I: beq $6, $5, A1															F T	F A	D	E	M T	M A	W

22 cycles / 2 iterations

For forwarding from E1 to F1, E2 to F2, it is also correct to forward from E->D, MT->E, or MA->MT.

(c) Consider a 3-wide VLIW pipeline with the following restrictions: Each bundle includes at most 1 control instruction, at most 2 ALU instructions, and at most 2 memory operations. Assume that the same register restrictions as in 2(a) apply. Assuming that the pipeline has the same stages, forwarding and hazard detection support as in 2 (above), generate a sequence of VLIW bundles from the unrolled code in 2b and schedule them. How do the cycles/iteration compare with 2(a)?

CYCLE #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A1: lw $2, 40 ($6) A2: lw $8, -24 ($6) G: addi $6, $6, -128	FT FT FT	FA FA FA	D D D	E E E	MT MT MT	MA MA MA	W W W								
C1: lw $3, 40 ($7) C2: lw $9, -24 ($7) H: addi $7, $7, -128		FT FT FT	FA FA FA	D D D	E E E	MT MT MT	MA MA MA	W W W							
B1: lw $2, 0 ($2) B2: lw $8, 0 ($8) NOP			FT FT FT	FA FA FA	d d d	D D D	E E E	MT MT MT	MA MA MA	W W W					
D1: lw $3, 0 ($3) D2: lw $9, 0 ($9) NOP				FT FT FT	fa fa fa	FA FA FA	D D D	E E E	MT MT MT	MA MA MA	W W W				
E1: add $3, $3, $2 E2: add $9, $8, $9 NOP					ft ft ft	FT FT FT	FA FA FA	d d d	d d d	D D D	E E E	MT MT MT	MA MA MA	W W W	
F1: sw $3, 0 ($2) F2: sw $9, 0 ($8) I: beq $6, $5, A1-A2-G							FT FT FT	fa fa fa	fa fa fa	FA FA FA	D D D	E E E	MT MT MT	MA MA MA	W W W

For forwarding from E1 to F1, E2 to F2, it is also correct to forward from E->E and MT->MT.

15 cycles/ 2 iterations

3. (30 pts) Branch prediction

Assume that you have a simple bimodal branch predictor (i.e., 2-bit dynamic branch predictor and table of 2-bit [N,n,t,T] state machines indexed by the PC) and that all predictors are initially in the 't' state.

a) If, for a given branch A, the sequence of (actual) directions is T, N, N, T, N, N, T, N, N
   i) For each dynamic instance of A, fill the table below.
   ii) What is the total hit count for the 9 predictions above? If A's pattern repeats forever, and no other branches alias to the same state machine, what will the final prediction rate converge to?

Actual-A	T	N	N	T	N	N	T	N	N
Predict	t(T)	T(T)	t(T)	n(N)	t(T)	n(N)	N(N)	n(N)	N(N)
Correct?	Yes	No	No	No	No	Yes	No	Yes	Yes

Accuracy: total hit 4, steady state converges to 2/3

b) Consider another branch B where the code containing B is only executed on the taken path of A, and B is consistently in the pattern T,T,T,N for each taken instance of A, and no dynamic instances of A will be seen within the code containing B (a.k.a., no recursive pattern).

i. If A and B have separate dedicated branch predictors and are the only two branches encountered, what is the overall branch prediction accuracy after the 9th dynamic instance of A has been predicted?

Actual-B	T	T	T	N	T	T	T	N	T	T	T	N
Predict	t(T)	T(T)	T(T)	t(T)	T(T)	T(T)	T(T)	t(T)	T(T)	T(T)	T(T)	t(T)
Correct?	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No

B hit rate is 9/12. So total hit rate combined A and B is 13/21 = (4+9)/(9+12)

ii. If A and B share the branch predictor and are the only two branches encountered, what is the overall branch prediction accuracy after the 9th dynamic instance of A has been predicted?

AB-Actual	T	T	T	T	N	N	N	T	T	T	T	N	N	N	T	T	T	T	N	N	N
Predict	t(T)	T(T)	T(T)	T(T)	T(T)	t(T)	n(N)	N(N)	n(N)	t(T)	T(T)	T(T)	t(T)	n(N)	N(N)	n(N)	t(T)	T(T)	T(T)	t(T)	n(N)
Correct?	Yes	Yes	Yes	Yes	No	No	Yes	No	No	Yes	Yes	No	No	Yes	No	No	Yes	Yes	No	No	Yes

Hit rate: 11/21