Solutions of the practice final

1. solution -> drawback

(a) increase cache size drawback -> increase cache access time

(b) Loop unrolling with compiler -> increase code size

(c) Hierarchical CLA -> More hardware

(d) reduce register field -> more register spilling

(e) pipeline -> more hardware

2. a. 22 cycle: 4 (to fill pipeline) + 7 (#instruction) + 8 (stall by data hazard) + 3 (control hazard)

b. 18 cycles: 4 + 7 + 4(stall by data hazard) + 3 (control hazard)

3. Miss, Miss, Miss, Miss, Hit, Miss, Hit, Miss

Compulsory, Compulsory, Compulsory, Conflict, Conflict, Conflict

4. a. Physical Memory size / page size = 2^30 / 2^15 = 2^15

b. page table size = #entry * the size of an entry = 2^17 x 2B = 2^18 Bytes

#entry = 2^17 (size of VPN)

the size of an entry = the size of PPN + 1 (the extra bit)= 30-15+1 = 15+1 (bit) = 2 Bytes

c. #entry in TLB / total #page table = 64 * 8 / 2^17 = 2^9 / 2^17 = 1 / 2^8 = 1 / 256

d. TLB has 64 sets, so 6 bit needed for the index. The 15 right most bits are used for page offset.

So, the six bits are: bit 20 ~ bit 15

5. Here, the solution ignores the branch delay slot mentioned in the problem.

5. (1) Without loop unrolling = 4 + (6 + 2) x 200 = 1604

* To fill the 5 stage pipeline = 4 cycles, the branch penalty = 2 cycles

Cycle	1st Issue Slot (ALU or Branch)	2nd Issue Slot (LW or SW)
1	addi $s0, $so, 4	lw $t0, 0($s0)
2	NOP	NOP
3	NOP	lw $t1, 0($t0)
4	NOP	NOP
5	add $t1, $s1, $t1	NOP
6	bne $s0, $s2, Loop	sw $t1, 0($t0)
7		
8		
9		
10		

5. (2) With loop unrolling = 4 + (7 + 2) x 100 = 904

* To fill the 5 stage pipeline = 4 cycles, the branch penalty = 2 cycles

* Register renaming for loop unrolling: $t0 -> $t2, $t1 -> $t3

Cycle	1st Issue Slot (ALU or Branch)	2nd Issue Slot (LW or SW)
1	NOP	lw $t0, 0($s0)
2	addi $s0, $so, **8**	lw $t2, **4**($s0)
3	NOP	lw $t1, 0($t0)
4	NOP	lw $t3, 0($t2)
5	add $t1, $s1, $t1	NOP
6	add $t3, $s1, $t3	sw $t1, 0($t0)
7	bne $s0, $s2, Loop	sw $t3, 0($t2)
8		
9		
10		

6. (a) the prediction of 1K-entry predictor

PC	Branch outcome	1K index	State		Prediction
			Current	Next	
128	T	128	00	01	NT
640	NT	640	00	00	NT
1152	NT	128	01	00	NT
128	T	128	00	01	NT
640	T	640	00	01	NT
1152	NT	128	01	00	NT
128	T	128	00	01	NT
640	NT	640	01	00	NT
1152	NT	128	01	00	NT
128	T	128	00	01	NT
640	T	640	00	01	NT
1152	NT	128	01	00	NT

(b) the prediction of 512-entry predictor

PC	Branch outcome	512 index	State		Prediction
			Current	Next	
128	T	128	00	01	NT
640	NT	128	01	00	NT
1152	NT	128	00	00	NT
128	T	128	00	01	NT
640	T	128	01	11	NT
1152	NT	128	11	10	T
128	T	128	10	11	T
640	NT	128	11	10	T
1152	NT	128	10	00	T
128	T	128	00	01	NT
640	T	128	01	11	NT
1152	NT	128	11	10	T

(c) the prediction of 768-entry predictor

PC	Branch outcome	768 index	State		Prediction
			Current	Next	
128	T	128	00	01	NT
640	NT	640	00	00	NT
1152	NT	384	00	00	NT
128	T	128	01	11	NT
640	T	640	00	01	NT
1152	NT	384	00	00	NT
128	T	128	11	11	T
640	NT	640	01	00	NT
1152	NT	384	00	00	NT
128	T	128	11	11	T
640	T	640	00	01	NT
1152	NT	384	00	00	NT

Answer

PC	Branch outcome	Prediction				Correct?
		1K entry	512 entry	768 entry	Final	
128	T	NT	NT	NT	NT	N
640	NT	NT	NT	NT	NT	Y
1152	NT	NT	NT	NT	NT	Y
128	T	NT	NT	NT	NT	N
640	T	NT	NT	NT	NT	N
1152	NT	NT	T	NT	NT	Y
128	T	NT	T	T	T	Y
640	NT	NT	T	NT	NT	Y
1152	NT	NT	T	NT	NT	Y
128	T	NT	NT	T	NT	N
640	T	NT	NT	NT	NT	N
1152	NT	NT	T	NT	NT	Y

7.

a. TCPI = BCPI + MCPI = 3.725

BCPI = Peak CPI(=1) + Data Hazard CPI (DHCPI) + Control Hazard CPI (CHCPI) = 1+ 0.075 + 0.1

DHCPI = 0.3 x 0.25 x 1 = 0.075

CHCPI = 0.15 x 1/3 x 2 = 0.1

So, BCPI = 1 + DHCPI + CHCPI = 1.175

MCPI = I$CPI + D$CPI = 1.5 + 1.05 = 2.55

I$CPI = 0.1 x (10 + 0.05 x 100) = 1.5

D$CPI = (0.1 + 0.25) x 0.2 x (10 + 0.05 x 100) = 1.05

b. Wrong, lw -> lw dependency will stall cause a stall since the result of EX (ALU) is used as input address of Data memory.

c. Right, it removes all stall by data hazard. (But, control hazard still exists.)

d. data hazard: the distance is 1 cycle-> 3 stalls, 2 cycle -> 2 stalls, 3 cycle -> 1 stall

control hazard: bench penalty is 4 cycle

BCPI = Peak CPI + DHCPI + CHCPI = 1 + 0.375 + 0.2 = 1.575

DHCPI = 0.25 x (**0.3 x 3 + 0.2 x 2 + 0.2 x 1**) = 0.25 x (0.9 + 0.4 + 0.2) = 0.375

CHCPI = 0.15 x 1/3 x **4 = 0.2**

MCPI => cycle time is half **-> miss penalty in cycle is doubled**

Since memory access "time" is not changed, halved cycle time takes twice number of cycles to reach the same access time.

MCPI = I$CPI + D$CPI = 3 + 2.1 = 5.1

I$CPI = 0.1 x (**20** + 0.05 x **200**) = 3

D$CPI = (0.1 + 0.25) x 0.2 (**20** + 0.05 x **200**) = 2.1

TCPI = 1.575 + 5.1 = 6.675

e. ET = IC x CPI x CT = 10^6 x 6.675 x **100ps**

= 10^6 x 6.675 x 1 x 10^-12 sec = 6.675 x 10^-4 sec

cf. the ET of baseline (5 stage) = 10^6 x 3.725 x **200ps** = 7.45 x 10^-4 sec