1. (i) Data forwarding alone can't satisfy here because by the time the first ALU instruction completes the second ALU step, the second ALU instruction is already in the first ALU step and so it's too late. Considering this, a Nop between the 2 instructions, as well as data forwarding between the second ALU step and the decode step.

(ii)

	cc0	cc1	cc2	cc3	cc4	cc5	cc6	cc7	cc8
ADD X1, X2, X3	IF	ID	ALU1	ALU2	MEM	WB			
NOP									
ADD X5, X4, X1		IF	ID	ALU1	ALU2	MEM	WB		

2.

2.1. Hazards identified:

or X13, X12, X11

ld X10, 0(X13) EX to 1st RAW Hazard

ld X11, 8(X13) Ex to 2nd RAW Hazard

add X12, X10, X11 MEM to 1st RAW (load-use-data) & MEM to 2nd Hazards

subi X13, X12, 16 EX to 1st RAW Hazard

NOPs introduced to resolve Hazards

or X13, X12, X11

NOPs

NOPs

ld X10, 0(X13) EX to 1st RAW Hazard resolution with 2NOPs

ld X11, 8(X13) EX to 2nd RAW Hazard resolved as well from above 2NOPs

NOPs

NOPs.

2.2.1 (cont.)

add x12, x10, x11 MEM to 1st RAW[load-use-data] & MEM to 2nd Hazards resolved with 2 NOPs.
NOPs
NOPs

subi x13, x12, 16 EX to 1st only RAW Hazard resolved with 2 NOPs

2.2

	clock cycle	1	2	3	4	5	6	7	8	9	10
1	or	IF	ID	EX	MEM	WB					
2	ld		IF	ID	EX	MEM	WB				
3	ld			IF	ID	EX	MEM	WB			
4	NOP	mandatory NOP for which no forwarding solution possible: load-data-use									
5	add					IF	ID	EX	MEM	WB	
6	subi						IF	ID	EX	MEM	WB

(1) A = x B = x (no instruction in Ex stage)

(2) A = x B = x (no instruction in EX stage)

(3) A = 0 B = 0 (both operands of the or instruction: x11, x12 come from Reg File)

(4) A = 2 B = 0 (base (RS1) in first ld (x13) taken from EX/MEM of previous instruction)

(5) A = 1 B = 0 (base (RS1) in second ld (x13) taken from EX/MEM/WB of previous instruction)

(6) A = x B = x (no instruction EX stage because NOP introduced to resolve MEM to 1st)

(7) A = 0 B = 1 (RS2 in the add instruction is x11 which is forwarded from MEM/WB of second ld. the result of the first ld (x10) has already been written into Reg File in CC6. so, no forwarding is needed for first operand)

(8) A = 1 B = 0 (RS1 of subi instruction forwarded from EX/MEM of add instruction)

41

3.

Yuqi Mao
Page: 3
problem: 3.

Miss rate = 0.05
Block size = 2 words (8 bytes)
Frequency of memory operations from processor = 10^9
Frequency of writes from processor = 0.25×10^9

Bus transfers one word at a time
On average 30% blocks in cache have been modified
c must be written back in the case of the write back cache)
Cach is write allocate

read hits = $0.75 \times 0.95 = 0.7125$
read misses = $0.75 \times 0.05 = 0.0375$.
write hits = $0.25 \times 0.95 = 0.2375$
write misses = $0.25 \times 0.05 = 0.0125$.

3.1 write through cache

1. On a read hit, there is no memory access
2. On a read miss, memory must send 2 words to the cache
3. On a write hit, the cache must send a word to memory
4. On a write miss, memory must send 2 words to the cache, and then the cache must send a word to memory.

Average words transferred = $0.7125 \times 0 + 0.0375 \times 2 + 0.2375 \times 1 + 0.0125 \times 3$
= 0.35.
Average bandwidth used = 0.35×10^9.
Fraction of bandwidth used = 0.35.

3.2 write back cache

1. on a read hit there is no memory access
2. On a read miss: ① If replaced line is modified then Cache must send 2 words to memory and then memory must send 2words to the cache ② If the replaced line is clean then memory must send 2 words to cache
3. On a write hit there is no memory access
4. On a write miss: ① If replaced line is modified the Cache must send 2words to memory and then the memory must send 2 words to the cache ② if replaced line is clean then memory must send 2 words to cache.

Average words transferred = $0.7125 \times 0 + 0.0375 \times (0.7 \times 2 + 0.3 \times 4) + 0.2375 \times 0 + 0.0125 \times (0.7 \times 2 + 0.3 \times 4)$
= 0.13.
Average bandwidth used = 0.13×10^9 Fraction of bandwidth used = 0.13.

6. 4A.

1. The cache has 2MB/8KB = 250 < 256 rows.
There are 256/4 = 64 sets
So $\log_2 64$ = 6 bits Index = 6 bits

2. 8 KB/block → 2000 words/block < 2^{11}
Offset = 11 bits.

3. Tag = 26 - 6 - 11 = 9 bits.

4B

	compulsory misses	conflict misses	capacity misses
increasing number of sets	no effect block size is constant	decreases more sets are available for data, so there is less of a chance for two ops to collide and evict one another	decreases capacity increases
increasing number of ways.	no effect block size is constant	decreases there are more ways available for data to be placed into	decreases capacity increases.
increasing number of bytes per line	decreases more data is brought in on a given cache miss	no effect associativity and number of sets is constant	decreases capacity increases

+0.0375 x (0.7x2 + 0.3x4) + 0.2375 x 0 + 0.025 x (0.7x2 + 0.3x4)

Average bandwidth used = 0.13

= 0.13×10^9 Fraction of bandwidth used = 0.13.

Yuqi Mao
Page: 5
problem 5

5-

	Instructions / Program	CPI	Circuit complexity
A	① Decrease. The new instructions will replace any two-instruction sequence that accomplished the same such as ADD rs1,rs2,rd ADD rs3,rd,rd	② The same The ALU will perform the three-way addition in one cycle. Also a acceptable increase because more RAW	② Increase Three-operand ALU is more complex than a two-operand one.
B	① The same No difference to instructions	② Increase All instructions now use the same ALU ALU operations now have to stall	③ Decrease Now there is one less adder/ALU cycle
C	① The same or decrease if the more registers enable the compiler to avoid loads and stores, Decreases. Otherwise, the same	② The same Does not affect the actions of each instructions	③ Increase More registers complicate the register file.

Yuqi Mao
Page 6
Problem: 6.

6.1

I-Mem is read, two registers are read, and a register is written

$140 + 2 \times 70 + 60 = 340$ PJ

6.2 $140 + 2 \times 70 + 60 + 140 = 480$ PJ

6.3 $140 + 2 \times 70 = 280$ PJ.