Quiz 5 Solution
EE309 - Microprocessors

Q1)

I1: lw $r1, 40($r6)

I2: add $r6, $r2, $r2

I3: sw $r6, 50($r1)

(a) RAW: I3 and I1, I3 and I2;

RAR: I1 and I3

WAR: I2 and I1

WAW: None

(b) With no data forwarding

		1	2	3	4	5	6	7	8	9
I1	lw	F	D	E	M	W				
I2	add		F	D	E	M	W			
I3	sw			stall	stall	F	D	E	M	W

(c) With data forwarding

		1	2	3	4	5	6	7
I1	lw	F	D	E	M	W		
I2	add		F	D	E	M	W	
I3	sw			F	D	E	M	W

Q 2 A)

```
         xor     $r1, $r1, $r1
         addi    $r1, $zero, 1024
loop:    lw      $r2, 1023($r1)
         sw      $r2, 2047($r1)
         subi    $r1, $r1, 1
         bne     $r1, $zero, loop
```

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
I1	Xor	F	D	E	M	W													
I2	Addi		F	D	E	M	W												
I3	Lw			■	■	F	D	E	M	W									
I4	Sw						■	■	F	D	E	M	W						
I5	Subi									F	D	E	M	W					
I6	Bne										■	■	F	D	E	M	W		
I7														■	F	D	E	M	W
						1	2	3	4	5	6	7	8	9	🟥	🟥	🟥		

Total no. of cycles = 4 + 9*1024 + 3 = 9223

Q 2 b (forwarding and delay slot)

		1	2	3	4	5	6	7	8	9	10	11	12
I1	Xor	F	D	E	M	W							
I2	Addi		F	D	E	M	W						

I3	Lw			F	D	E	M	W					
I4	Sw				F	D	E	M	W				
I5	Subi					F	D	E	M	W			
I6	Bne						■	F	D	E	M	W	
I7									F	D	E	M	W
				1	2	3	4	5	6	🟥	🟥	🟥	🟥

Total no. of cycles = 2 + 6*1024 + 4 = 6150

Improved code: (modifications in red)

```
        xor     $r1, $r1, $r1
        addi    $r1, $zero, 1024
loop:   lw      $r2, 1023($r1)
        subi    $r1, $r1, 1
        bne     $r1, $zero, loop
        sw      $r2, 2048($r1)
```

(complete the clock-cycle vs instruction diagram !)

Improved code: (modifications in red). This does more than re-ordering and may not be acceptable if the question had specified only re-order.

```
        xor     $r1, $r1, $r1
        addi    $r1, $zero, 1023
loop:   lw      $r2, 1024($r1)
        sw      $r2, 2048($r1)
        bne     $r1, $zero, loop
        subi    $r1, $r1, 1
        addi    $r1, $r1, 1
```

		1	2	3	4	5	6	7	8	9	10	11
I1	Xor	F	D	E	M	W						
I2	Addi		F	D	E	M	W					
I3	Lw			F	D	E	M	W				
I4	Sw				F	D	E	M	W			
I5	Bne					F	D	E	M	W		
I6	subi						F	D	E	M	W	
I7	addi							F	D	E	M	W
				1	2	3	4	5	🟥	🟥	🟥	🟥

Total no. of cycles = 2 + 5*1024 + 4 = 5126

Q3.

(a). Cache size 64 KB = 2^{16} bytes; Block size 64B = 2^{6} bytes; 4-way implies 4 blocks per set

Block size = 2^{6} bytes => 6 bits for block offset

No. of sets = 2^{16} / (4 * 2^{6}) = 2^{8} => 8 bits for set index

2GB physical memory => 2^{31} bytes => 31 bits for address

Hence, Tag – 17 bits, Set Index – 8 bits, Block offset – 6

(b) Double the cache size:
No. of sets = 2^{17} / (4 * 2^{6}) = 2^{9} => 9 bits for set index
Hence, Tag – 16 bits, Set index – 9 bits, Block offset – 6
Change => -1, +1,,0

Direct mapped cache:
No. of blocks = 2^{16} / (2^{6}) = 2^{10}
Hence, Tag – 15 bits, Block index – 10 bits, Block offset – 6
Change => -2, +2, 0

Fully associative:
Except block offset, all bits are Tag
Hence, Tag – 25 bits, Block offset – 6 bits
Change => +8, -8, 0