Pipeline 0 Mips - simulation table

Home work 7

4.

Cycle	Instruction memory		Register File				ALU				Memory			PC Src Mux		
	Addr	Instr	Read Reg 1	Read data 1	Write Reg	Reg write	A	B	Op	ALU Result	L	O	S	1	0	S
1	0x00000010	lui	x	x	x	x	x	x	x	x	x	x	x	0xXXXX XXXX	0x0000 0014	0
2	0x00000014	addi	x	x	x	x	x	x	x	x	1	x	x	0xXXXX XXXX	0x0000 0018	0
3	0x00000018	sub	0x000	0x0000 0000	x	x	x	0x0000 1001	lui	0x1001 0000	x	x	x	0xXXXX XXXX	0x0000 001C	0
11	0x0000001C	xor	0x004	0x0000 0003	x	x	0x0	0x0000002A	add	0x0000002A	x	0x 18	0	0xXXXX XXXX	0x0000 0020	0
5	0x00000020	ori	0x006	0x0000 03FF	x	x	0x0000 0003	0x00006400	sub	0xFFFFFFFB	x	0x 08	0	0x0000 0100	0x0000 0024	0
6	0x00006024	beq	0x00	0x1001 0000	0x10	1	0x0000 03FF	0xFFFFFFFF	xor	0x10000400	x	0x 04	x	0xXXXX XXXX	0x0000 0024	0
7	0x00000028	addi	0x08	0x0000 002A	0x09	1	0x1001 0000	0x00000040	or	0x0000 007B	x	0x 0A	0	0xXXXX XXXX	0x0000 002C	0
8	0x0000002C	sll	0x00	0x6000 0000	0x08	1	0x0000 002A	0xFFFF FFFC	sub	0x0000 0000	x	0x 10	0	0xXXXX XXXX	0x0000 0030	0
9	0x00000030	sw	0x00	0x6000 0000	0x0A	1	0x0000 0040	0x0000 0000	add	0x0000 0000	x	x	0	0xXXXX XXXX	0x0000 0034	0
10	0x00000034		0x10	0x1001 0040	0x10	0	0x0000 0000	x	sll	0x1001 0040	x	0x 00	0	0xXXXX XXXX	0x0000 0038	0
11	0x00000038				0xXX	1	0x1001 0040	0x0000 0000	add		x	0x 00	0	0xXXXX XXXX	0x0000 003C	0
12	0x0000003C				0x00	1					x	x	x	0xXXXX XXXX	0x0000 0040	0
13	0x00000040				0xXX	0								0xXXXX XXXX	0x0000 1044	0

lui $s0, 0x1001 $s0 = 4097

addi $t0, $zero, 42 $t0 = 42

sub $t1, $a0, $a1 $t1 = 3 - 1024 = (-1021)

xor $t2, $t2, $s3 $t2 = 1023

or $s0, $s0, 0x0040

beq $t0, $s3, Exit 42 =/= -1

addi $t4, $zero, 0

sll $zero, $zero, 0

sw $10, 0($s0)

2. Frodo:

```
lw   $t0, 0($a1)      WB
                      <- 2 nops
add  $t1, $t0, $a2    read
                      <- 2 nops
lw   $t0, 0($a0)      WB
add  $t0, $t0, $t1    WB read
                      <- 2 nops
sw   $t0, 0($a0)      WB
sub  $a3, $a3, 1      read, write
addi $a0, $a0, 4      ^
addi $a1, $a1, 4      |
bne  $a3, $0, Frodo.  write
```

6 stalls! 6 nops

3. Is W3 a Data hazard?

Add R1, R2, R3
Add R4, R1, R5

If so explain why, and (if there it can be explain how many stalls are needed for it, if

Solution: yes! It is a data hazard, because it is reading before it can finish writing. This requires 2 stalls to move the ID down the pipeline along with 2 additional cycles.