submitting by Rotem Kashani 209073352

Computer Architecture Exercise 8

Consider the following code fragment for the DLX CPU:

```
        LW R2, 1000(R0)
loop:   LW R4, 0(R2)
        SUBI R4, R4, #D0
        SW 0(R2), R4
        ADDI R2, R2, #4
        ADD R3, R2, R4
        BEQZ R3, loop
        AND R11, R8, R4
```

1. In its first iteration of the instruction **`BEQZ R3, loop`** the value in register **R3** is **1234**. Analyze the execution of the program for version 2 of the DLX, but completing the following tables:

Format 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LW R2, 1000(R0)	IF	ID	EX	MEM	WB									
LW R4, 0(R2)		IF	ID	ID	ID	EX	MEM	WB						
SUBI R4, R4, #D0			IF	IF	IF	ID	ID	ID	EX	MEM	WB			
SW 0(R2), R4						IF	IF	IF	ID	ID	ID	EX	MEM	WB
ADDI R2, R2, #4									IF	IF	IF	ID	ID	ID
ADD R3, R2, R4												IF	IF	IF
BEQZ R3, loop														
AND R11, R8, R4														

	15	16	17	18	19	20	21	22	23	24
LW R2, 1000(R0)										
LW R4, 0(R2)										
SUBI R4, R4, #D0										
SW 0(R2), R4										
ADDI R2, R2, #4	EX	MEM	WB							
ADD R3, R2, R4	ID	ID	ID	EX	MEM	WB				
BEQZ R3, loop	IF	IF	IF	ID	ID	ID	EX	MEM	WB	
AND R11, R8, R4				IF	IF	IF	ID	EX	MEM	WB

submitting by Rotem Kashani 209073352

Format 2

	IF	ID	EX	MEM	WB
1	LW R2, 1000(R0)				
2	LW R4, 0(R2)	LW R2, 1000(R0)			
3	SUBI R4, R4, #D0	LW R4, 0(R2)	LW R2, 1000(R0)		
4	SUBI R4, R4, #D0	LW R4, 0(R2)	Ø	LW R2, 1000(R0)	
5	SUBI R4, R4, #D0	LW R4, 0(R2)	Ø	Ø	LW R2, 1000(R0)
6	SW 0(R2), R4	SUBI R4, R4, #D0	LW R4, 0(R2)	Ø	Ø
7	SW 0(R2), R4	SUBI R4, R4, #D0	Ø	LW R4, 0(R2)	Ø
8	SW 0(R2), R4	SUBI R4, R4, #D0	Ø	Ø	LW R4, 0(R2)
9	ADDI R2, R2, #4	SW 0(R2), R4	SUBI R4, R4, #D0	Ø	Ø
10	ADDI R2, R2, #4	SW 0(R2), R4	Ø	SUBI R4, R4, #D0	Ø
11	ADDI R2, R2, #4	SW 0(R2), R4	Ø	Ø	SUBI R4, R4, #D0
12	ADD R3, R2, R4	ADDI R2, R2, #4	SW 0(R2), R4	Ø	Ø
13	ADD R3, R2, R4	ADDI R2, R2, #4	Ø	SW 0(R2), R4	Ø
14	ADD R3, R2, R4	ADDI R2, R2, #4	Ø	Ø	SW 0(R2), R4
15	BEQZ R3, loop	ADD R3, R2, R4	ADDI R2, R2, #4	Ø	Ø
16	BEQZ R3, loop	ADD R3, R2, R4	Ø	ADDI R2, R2, #4	Ø
17	BEQZ R3, loop	ADD R3, R2, R4	Ø	Ø	ADDI R2, R2, #4
18	AND R11, R8, R4	BEQZ R3, loop	ADD R3, R2, R4	Ø	Ø
19	AND R11, R8, R4	BEQZ R3, loop	Ø	ADD R3, R2, R4	Ø
20	AND R11, R8, R4	BEQZ R3, loop	Ø	Ø	ADD R3, R2, R4
21		AND R11, R8, R4	BEQZ R3, loop	Ø	Ø
22			AND R11, R8, R4	BEQZ R3, loop	Ø
23				AND R11, R8, R4	BEQZ R3, loop
24					AND R11, R8, R4

submitting by Rotem Kashani 209073352

2. Repeat the analysis in question 1 for version 6c of the DLX.
In format 1, include arrows to indicate the forwarding of intermediate values between registers.

Format 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LW R2, 1000(R0)	IF	ID	EX	MEM	WB									
LW R4, 0(R2)		IF	ID	ID	EX	MEM	WB							
SUBI R4, R4, #D0			IF	IF	ID	ID	EX	MEM	WB					
SW 0(R2), R4					IF	IF	ID	EX	MEM	WB				
ADDI R2, R2, #4							IF	ID	EX	MEM	WB			
ADD R3, R2, R4								IF	ID	EX	MEM	WB		
BEQZ R3, loop									IF	ID	EX	MEM	WB	
AND R11, R8, R4										IF	ID	EX	MEM	WB

Format 2

	IF	ID	EX	MEM	WB
1	LW R2, 1000(R0)				
2	LW R4, 0(R2)	LW R2, 1000(R0)			
3	SUBI R4, R4, #D0	LW R4, 0(R2)	LW R2, 1000(R0)		
4	SUBI R4, R4, #D0	LW R4, 0(R2)	Ø	LW R2, 1000(R0)	
5	SW 0(R2), R4	SUBI R4, R4, #D0	LW R4, 0(R2)	Ø	LW R2, 1000(R0)
6	SW 0(R2), R4	SUBI R4, R4, #D0	Ø	LW R4, 0(R2)	Ø
7	ADDI R2, R2, #4	SW 0(R2), R4	SUBI R4, R4, #D0	Ø	LW R4, 0(R2)
8	ADD R3, R2, R4	ADDI R2, R2, #4	SW 0(R2), R4	SUBI R4, R4, #D0	Ø
9	BEQZ R3, loop	ADD R3, R2, R4	ADDI R2, R2, #4	SW 0(R2), R4	SUBI R4, R4, #D0
10	AND R11, R8, R4	BEQZ R3, loop	ADD R3, R2, R4	ADDI R2, R2, #4	SW 0(R2), R4
11		AND R11, R8, R4	BEQZ R3, loop	ADD R3, R2, R4	ADDI R2, R2, #4
12			AND R11, R8, R4	BEQZ R3, loop	ADD R3, R2, R4
13				AND R11, R8, R4	BEQZ R3, loop
14					AND R11, R8, R4