b. Improve the pipeline by adding branch prediction. Assume the branch is predicted correctly, and diagram below.

If we assume that the branch is correct, then we have no stalls in our pipeline

we predict branch will or won't happen so we don't stall for $t1

Cycle

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	22	21	22
lw $t1, 0($a0)	IF	RF	E	M	WB									IF	RF	E	M	WB					
beq $t1, $0, targ1		IF	RF	E	M	WB									IF	RF	E	M	WB				
addi $a0, $a0, 4			IF	RF	E	M	WB									IF	RF	E	M	WB			
add $t1, $t1, $t3				IF	X	RF	E	M	WB								IF	X	RF	E	M	WB	
lw $t1, 0($t1)						IF	X	X	X	RF	E	M	WB						IF	X	X	X	RF
add $v0, $v0, $t1										IF	X	X	X	RF	I	M	WB						IF

No Stall

Instructions following branch are flushed, so we don't stall

still depends!

X = Stalls

Still a dependency because we need WB from the LW instruction

c. Further improve the pipeline by adding register forwarding. Diagram below.

still have branch prediction

$t1 is computed in E stage

LW now comes from memory

Cycle

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	22	21	22
lw $t1, 0($a0)	IF	RF	E	M	WB				IF	RF	E	M	WB				IF	RF	E	M	WB		
beq $t1, $0, targ1		IF	RF	E	M	WB				IF	RF	E	M	WB				IF	RF	E	M	WB	
addi $a0, $a0, 4			IF	RF	E	M	WB				IF	RF	E	M	WB				IF	RF	E	M	WB
add $t1, $t1, $t3				IF	RF	E	M	WB				IF	RF	E	M	WB				IF	RF	E	M
lw $t1, 0($t1)					IF	X	RF	E	M	WB			IF	X	RF	E	M	WB			IF	X	RF
add $v0, $v0, $t1							IF	X	RF	E	M	WB			IF	X	RF	E	M	WB			IF

take branch flushed

data haz

data haz

Stall, we depend on lw

registers are found in execute stages