Cycle	*Instruction Retired*	*Reason*
1	NOP	No instruction retired yet.
2	NOP	No instruction retired yet.
3	NOP	No instruction retired yet.
4	NOP	No instruction retired yet.
5	lbi r0, 0	Instruction in Write Back stage.
6	lbi, r5, 43	Instruction in Write Back stage.
7	lbi, r6, 43	Instruction in Write Back stage.
8	lbi, r7, 43	Instruction in Write Back stage.
9	ld r1, r0, 0	Instruction in Write Back stage.
10	NOP	st r5, r1, 0 stalls at its Decode stage to wait for the value of r1 that will be written in by last instruction ld r1, r0, 0 (RAW on r1)
11	NOP	st r5, r1, 0 stalls at its Decode stage to wait for the value of r1 that will be written in by last instruction ld r1, r0, 0 (RAW on r1)
12	st r5, r1, 0	Instruction in Write Back stage.
13	ld r1, r0, 2	Instruction in Write Back stage.
14	NOP	st r6, r1, 1 stalls at its Decode stage to wait for the value of r1 that will be written in by last instruction ld r1, r0, 2 (RAW on r1)
15	NOP	st r6, r1, 1 stalls at its Decode stage to wait for the value of r1 that will be written in by last instruction ld r1, r0, 2 (RAW on r1)
16	st r6, r1, 1	Instruction in Write Back stage.
17	ld r1, r0, 4	Instruction in Write Back stage.
18	NOP	st r7, r1, 1 stalls ate its Decode stage to wait for the value of r1 that will be written in by last instruction ld r1, r0, 4 (RAW on r1)
19	NOP	st r7, r1, 1 stalls ate its Decode stage to wait for the value of r1 that will be written in by last instruction ld r1, r0, 4 (RAW on r1)
20	st r7, r1, 1	Instruction in Write Back stage.
	Halt	The program halts.

*We are using a bypassing register file, so there are usually two stalls for a data dependency.

* In our design, the processor will detect Halt in the Halt instruction's MEM stage and then end the program. Thus, the Halt instruction will not enter Write Back stage. That's why we put nothing in the cycle number column for Halt in the above table as there is no such cycle that Halt is in Write Back stage.