5 stage MIPS pipeline.

a) No Forwarding Branch optimization

Instr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LD	F	D	X	M	W															
DADDI		F	D	s	s	X	M	W												
SD			F	s	s	D	s	s	X	M	W									
DADDI						F	s	s	D	X	M	W								
DSUB									F	D	s	s	X	M	W					
BNEZ										F	s	s	D	s	s	X	M	W		
LD (loop start)													F	s	s	F	D	X	M	W

- First DADDI waits till the Writeback stage of LD to obtain value of R1.
- (No Forwarding) SD waits till computation by the first DADDI to get the value of R1 and consequent Writeback Stage.
- DSUB waits for second DADDI to compute value of R2 and Consequent Writeback Stage.
- BNEZ waits until DSUB computes value of R4 and reaches the Writeback stage.
- From the next Loop/iteration, LD is re-fetched after branch resolution.

Loop executes 99 iterations (396/4)from 0 to 98. Iteration i begins in cycle $1 + (i * 15)$ (see figure). The last iteration takes 18 cycles to complete. Total number of cycles for the whole loop is calculated as $(98 * 15) + 18 = 1488$ cycles.

b) Forward and Branch Predict not taken

Instr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LD	F	D	X	M	W									
DADDI		F	D	s	X	M	W							
SD			F	s	D	X	M	W						
DADDI					F	D	X	M	W					
DSUB						F	D	X	M	W				
BNEZ							F	D	s	X	M	W		
LD (loop start)								F	s	F	D	X	M	W

- DADDI waits until LD gets to Writeback to obtain Value of R1 and stall is implemented now in the Execute stage through interlocking mechanism and value is directly forwarded into execution stage
- SD now waits until the first DADDI computes the value of R1 and forwards the value from Execution stage to Memory stage.

- DSUB obtains the value of R2 forwarded by the second DADDI and does not need to wait
- BNEZ now gets the value of R4 forwarded by sub but needs to wait till the sub reaches Execution stage.
- LD from the next iteration is re-fetched after the branch is resolved as mispredicted (in the branch ID stage).

Iteration i now begins in cycle $1 + (i * 9)$ and the last iteration takes only 12 cycles to complete. So, the total number of cycles for the whole loop is $(98 * 9) + 12 = 894$ cycles.

c) Simple instruction scheduling and branch delay slot

- Optimization 1: move the second DADDI to the load delay slot
- Optimization 2: move DSUB up, to after the second DADDI.
- Optimization 3: move SD to the branch delay slot, adjusting the offset (SD -4 (R2), R1).

Instr.	1	2	3	4	5	6	7	8	9	10	11
LD R1,0(R2)	F	D	X	M	W						
DADDI R2, R2, 4		F	D	X	M	W					
DSUB R4,R3,R2			F	D	X	M	W				
DADDI R1,R1,1				F	D	X	M	W			
BNEZ R3,loop					F	D	X	M	W		
SD -4(R2), R1						F	D	X	M	W	
LD (loop start)							F	D	X	M	W

- First DADDI does not wait as it is now separated by two cycles from LD.
- BNEZ now does not wait as it is separated by one cycle from DSUB.
- SD now fills the branch delay slot.
- LD from the next iteration is properly fetched as the branch is resolved in time.

Iteration i now begins in cycle $1 + (i * 6)$ and the last iteration takes only 10 cycles to complete. So, the total number of cycles for the whole loop is $(98 * 6) + 10 = 598$ cycles.