Noel Ball
HW3

Question 1:

A.) If each step computed sequentially the throughput would be 1 instruction / 40 ns. However with pipelining each step can be its own stage, and once the pipeline is full a new instruction would be completed every 16 ns due to the longest stage, giving 1 instruction / 16 ns.

$$Speedup = \frac{Throughput\ Pipeline}{Throughput\ Sequence} = \frac{1/16}{1/40} = 2.5x$$

B.) If the longest step of 16 ns was split into two pipeline stages of 8 ns, the new longest step would be 10 ns, using 5 stages:
Stage A (6 ns) → Stage B (8 ns) → Stage $C_{1/2}$ (8 ns) → Stage $C_{2/2}$ (8 ns) → Stage D (10 ns).
By decreasing the pipeline's cycle time the throughput is heavily increased, and the latency is only slightly increased by adding one stage.
Using fewer stages would keep the throughput unnecessarily high from only one step being longer than the rest, however adding any more would increase the latency too much.

Question 2:

A.) RAW:
I_2 reads t_0 written by I_1
I_3 reads t_1 written by I_2 and t_0 written by I_1
I_5 reads t_0 written by I_3 and a_2 written by I_4

WAR:
I_3 writes to t_0 after I_2 reads t_0

WAW:
I_3 writes to t_0 after I_1 writes to t_0

B.) Stalls are highlighted, Eliminated Stalls are underlined

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I_1	F	D	X	M	W										
I_2		F	D	D	D	X	M	W							
I_3			F	D	D	D	D	D	X	M	W				
I_4				F	D	X	M	W							
I_5					F	D	D	D	D	D	D	X	M	W	

C.) Yes, 6 stall cycles would be eliminated.

D.) Stalls are highlighted

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I_1	F	D	X	M	W										
I_2		F	D	X	M	W									
I_3			F	D	X	M	W								
I_4				F	D	X	M	W							
I_5					F	D	X	M	W						

Question 3:

A.) Stalls are highlighted

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I_1	F	D	X	M	W										
I_2		F	D	X	M	W									
I_3			F	D	X	M	W								
I_4				F	D	X	M	W							
I_5					F	D	X	M	W						

Bi.) The instruction count stays the same without forwarding paths, because the program's algorithm and operations it needs to perform are unchanged.

Bii.) CPI increases because removing forwarding paths increases the number of pipeline stalls due to data hazards, raising the number of cycles required per instruction on average.

Biii.) Clock period may decrease because removing forwarding paths reduces the number of hardware circuits adding complexity and load to the processor, allowing for a slightly more efficient clock period.

Question 4:

A.) Stalls are highlighted

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I_1	F	D	X	M	W										
I_2		F	D	X	M	W									
I_3			F	D	X	M	W								
I_4				F	D	X	M	W							
I_5					F	D	X	M	W						

B.) Splitting the memory stage into two separate halves would worsen the CPI of this processor with this code since I_2 depends on t_0 from I_1, the increased delay from memory result not finishing until after the second memory stage will cause stalls.

C.) Yes, rescheduling the code would reduce the stall caused by the successive memory operations if a0 is incremented between ld and sd, sd has time to receive t_0 from ld's M2 stage:

loop:

ld t_0, 0(a_0)
addi a_0, a_0, 8
sd t_0, 0(a_1)
addi a_1, a_1, 8
bne a_0, a_2, loop