CMSC611: Advanced Computer Architecture
Homework 3

Question 1: (50 points)

a) See "Original code" on the attached pipeline chart for details.

b) See "Schedule code" on the attached pipeline chart for details.

c) Based on the initial value of R3, the loop iterates 4 times. Unroll the loop by replicating the loop body 4 times as:

Cycle with stalls	Instruction
1	LOOP: L.D F2, 0(R1)
3	MUL.D F4, F2, F0
4	L.D F6, 0(R2)
10	ADD.D F6, F4, F6
13	S.D 0(R2), F6
14	L.D F8, 8(R1)
16	MUL.D F10, F8, F0
17	L.D F12, 8(R2)
23	ADD.D F12, F10, F12
26	S.D 8(R2), F12
27	L.D F14, 16(R1)
29	MUL.D F16, F14, F0
30	L.D F18, 16(R2)
36	ADD.D F18, F16, F18
39	S.D 16(R2), F18
40	L.D F20, 24(R1)
42	MUL.D F22, F20, F0
43	L.D F24, 24(R2)
49	ADD.D F24, F22, F24
52	S.D 24(R2), F24
53	DADDIU R1, R1, #32
54	DADDIU R2, R2, #32
55	DSUBIU R3, R3, #64
57	BNEZ R3, LOOP

Schedule the unrolled code by reordering the instructions to reduce stalls as:

Clock Cycle	Instruction
1	LOOP: L.D F2, 0(R1)
2	L.D F8, 8(R1)
3	L.D F14, 16(R1)
4	L.D F20, 24(R1)
5	MUL.D F4, F2, F0
6	MUL.D F10, F8, F0
7	MUL.D F16, F14, F0
8	MUL.D F22, F20, F0
9	L.D F6, 0(R2)
10	L.D F12, 8(R2)
11	L.D F18, 16(R2)
12	L.D F24, 24(R2)
13	ADD.D F6, F4, F6
14	ADD.D F12, F10, F12
15	ADD.D F18, F16, F18
16	ADD.D F24, F22, F24
17	S.D 0(R2), F6
18	S.D 8(R2), F12
19	S.D 16(R2), F18
20	S.D 24(R2), F24
21	DSUBIU R3, R3, #64
22	DADDIU R1, R1, #32
23	BNEZ R3, LOOP
24	DADDIU R2, R2, #32

With the scheduling, the new code performs better than the unrolled code on clock cycles.

Question 2: (50 points)

a) Instruction status table:

Instruction	Issue	Execution	Write Result
L.D F0, 8(R1)	1	2-3	4
L.D F2, 8(R2)	2	3-4	5
ADD.D F4, F2, F4	3	6-9	10
MUL.D F8, F6, F4	4	11-18	19
SUB.D F6, F10, F0	5	6-9	11
DIV.D F12, F6, F2	6	12-41	42
S.D 16(R3), F12	7	43-44	45
ADD.D F12, F8, F2	10	20-23	24
S.D 16(R4), F12	11	25-26	27
DADDIU R1, R1, #8	27	28-29	30
DADDIU R2, R2, #8	30	31-32	33

b) Since there are only 2 reservation stations for Integer and Load/Store, we denote those reservation stations as Integer1 and Integer2 in this solution. If a load buffer is occupied, we mark "Yes" in column "Busy" for one of two Integer reservation stations, and we show the memory address in column "Vj"; if a store buffer is occupied, we mark "Yes" in column "Busy" for one of two Integer reservation stations, and we show the memory address in column "Vj" and the reservation station producing the source register (i.e., Qi for the store instruction) in column "Qk".

Reservation station and register result status tables on the 4th clock cycle:

Reservation Station							
Time	**Name**	**Busy**	**Op**	**S1 Vj**	**S2 Vk**	**RS for j Qj**	**RS for k Qk**
0	Integer1	No					
0	Integer2	Yes	L.D	M(8+R2)			
0	Add1	Yes	ADD.D		R(F4)	Integer2	
0	Add2	No					
0	Mult1	Yes	MUL.D	R(F6)			Add1
0	Mult2	No					
0	Div1	No					
0	Div2	No					

Register Result Status					
Clock		**F0**	**F2**	**F4**	**F8**
4	FU	M(8+R1)	Integer2	Add1	Mult1

Reservation station and register result status tables on the 6th clock cycle:

Reservation Station							
Time	**Name**	**Busy**	**Op**	**S1 Vj**	**S2 Vk**	**RS for j Qj**	**RS for k Qk**
0	Integer1	No					
0	Integer2	No					
3	Add1	Yes	ADD.D	M(8+R2)	R(F4)		
3	Add2	Yes	SUB.D	R(F10)	M(8+R1)		
0	Mult1	Yes	MUL.D	R(F6)			Add1
0	Mult2	No					
0	Div1	Yes	DIV.D		M(8+R2)	Add2	
0	Div2	No					

Register Result Status							
Clock		**F0**	**F2**	**F4**	**F6**	**F8**	**F12**
6	FU	M(8+R1)	M(8+R2)	Add1	Add2	Mult1	Div1

Reservation station and register result status tables on the 13th clock cycle:

Reservation Station							
Time	**Name**	**Busy**	**Op**	**S1 Vj**	**S2 Vk**	**RS for j Qj**	**RS for k Qk**
0	Integer1	Yes	S.D	M(16+R3)			Div1
0	Integer2	Yes	S.D	M(16+R4)			Add1
0	Add1	Yes	ADD.D		M(8+R2)	Mult1	
0	Add2	No					
5	Mult1	Yes	MUL.D	R(F6)	M(8+R2)+ R(F4)		
0	Mult2	No					
28	Div1	Yes	DIV.D	R(F10)- M(8+R1)	M(8+R2)		
0	Div2	No					

Register Result Status							
Clock		**F0**	**F2**	**F4**	**F6**	**F8**	**F12**
13	FU	M(8+R1)	M(8+R2)	M(8+R2)+R(F4)	R(F10)- M(8+R1)	Mult1	Add1