HY425 – Computer Architecture
Chris Papastamos | csd4569
Assignment 2

Problem 1

i. For the execution of one itteration of the given loop I will calculate each instruction delay as their respective delay +1 for their instruction fetch.

Instruction					Cycles
Loop:	LD	F2,	0(Rx)		1+4=5
I0:	DIVD	F8,	F2,	F0	1+8=9
I1:	MULTD	F2,	F6,	F2	1+3=4
I2:	LD	F4,	0(Ry)		1+4=5
I3:	ADDD	F4,	F0,	F4	1+2=3
I4:	ADDD	F10,	F8,	F2	1+2=3
I5:	ADDI	Rx,	Rx,	8	1+0=1
I6:	ADDI	Ry,	Ry,	8	1+0=1
I7:	SD	F4,	0(Ry)		1+1=2
I8:	SD	F2,	0(Rx)		1+1=2
I9:	SUB	R20,	R4,	Rx	1+0=1
I10:	BNZ	R20,	Loop		1+1=2

Total:38 Cycles

ii.

Instruction					Strart Cycle	Ending Cycle	
Loop:	LD	F2,	0(Rx)		1	6	
I0:	DIVD	F8,	F2,	F0	7	16	<stall>
I1:	MULTD	F2,	F6,	F2	8	12	
I2:	LD	F4,	0(Ry)		9	14	
I3:	ADDD	F4,	F0,	F4	15	18	<stall>
I4:	ADDD	F10,	F8,	F2	17	20	<stall>
I5:	ADDI	Rx,	Rx,	8	18	21	
I6:	ADDI	Ry,	Ry,	8	19	22	
I7:	SD	F4,	0(Ry)		23	25	<stall>
I8:	SD	F2,	0(Rx)		24	26	
I9:	SUB	R20,	R4,	Rx	25	26	
I10:	BNZ	R20,	Loop		27	29	<stall>

The pipeline will run for **29 Cycles**!

iii.

Instruction					Strart Cycle	Ending Cycle	
Loop:	LD	F2,	0(Rx)		1	6	
I0:	LD	F4,	0(Ry)		2	7	
I1:	ADDI	Rx,	Rx,	8	3	6	
I2:	ADDI	Ry,	Ry,	8	4	7	
I3:	DIVD	F8,	F2,	F0	7	16	<stall>
I4:	ADDD	F4,	F0,	F4	8	11	
I5:	MULTD	F2,	F6,	F2	12	15	
I6:	SUB	R20,	R4,	Rx	13	14	
I7:	SD	F4,	0(Ry)		15	17	
I8:	SD	F2,	0(Rx)		16	18	
I9:	ADDD	F10,	F8,	F2	17	21	
I10:	BNZ	R20,	Loop		18	19	

The reordered code will run for **21 Cycles**!

iv.

Instruction					Strart Cycle	Ending Cycle	
Loop:	LD	F2,	0(Rx)		1	6	
I0:	LD	F4,	0(Ry)		2	7	
I1:	LD	F2a,	8(Rx)		3	8	
I2:	LD	F4a,	8(Ry)		4	9	
I3:	DIVD	F8,	F2,	F0	7	16	<stall>
I4:	ADDD	F4,	F0,	F4	8	9	
I5:	MULTD	F2,	F6,	F2	9	13	
I6:	SUB	R20,	R4,	Rx	10	11	
I7:	DIVD	F8a,	F2a,	F0	11	20	
I8:	ADDD	F4a,	F0,	F4a	12	15	
I9:	MULTD	F2,	F6,	F2a	13	17	
I10:	SUB	R20a,	R20,	Rx	14	15	
I11:	SD	F4,	8(Ry)		15	17	
I12:	SD	F2,	8(Rx)		16	18	
I13:	SD	F4a,	16(Ry)		17	19	
I14:	SD	F2a,	16(Rx)		18	20	
I15:	ADDI	Rx,	Rx,	16	19	20	
I16:	ADDI	Ry,	Ry,	16	20	21	
I17:	ADDD	F10,	F8,	F2	21	24	
I18:	ADDD	F10a,	F8a,	F2a	22	25	
I19:	BNZ	R20a,	Loop		23	24	

For two itterations unrolled the pipeline will use 24 Clock Cycles.

Problem 2

In left table we can see the instructions before register renaming. The colored registers need to be renamed in order to achieve maximum efficiency. In the table on the right is the loop using the renamed registers

Instruction				
Loop:	LD	F2,	0(Rx)	
I0:	DIVD	F8,	F2,	F0
I1:	MULTD	F2,	F6,	F2
I2:	LD	F4,	0(Ry)	
I3:	ADDD	F4,	F0,	F4
I4:	ADDD	F10,	F8,	F2
I5:	ADDI	Rx,	Rx,	8
I6:	ADDI	Ry,	Ry,	8
I7:	SD	F4,	0(Ry)	
I8:	SD	F2,	0(Rx)	
I9:	SUB	R20,	R4,	Rx
I10:	BNZ	R20,	Loop	

Instruction				
Loop:	LD	F2,	0(Rx)	
I0:	DIVD	F8,	F2,	F0
I1:	MULTD	F2a,	F6,	F2
I2:	LD	F4,	0(Ry)	
I3:	ADDD	F4a,	F0,	F4
I4:	ADDD	F10,	F8,	F2a
I5:	ADDI	Rxa,	Rx,	8
I6:	ADDI	Rya,	Ry,	8
I7:	SD	F4a,	0(Rya)	
I8:	SD	F2a,	0(Rxa)	
I9:	SUB	R20,	R4,	Rxa
I10:	BNZ	R20,	Loop	

The new registers used are: F2a, F4a, Rxa and Rya

ii. Lets now look at the execution of this code snippet with renamed registers:

Instruction					FU	Start Clock	Ending Clock
Loop:	LD	F2,	0(Rx)		LD/SD	1	6
I0:	DIVD	F8,	F2,	F0	ALU0	7	16
I1:	MULTD	F2a,	F6,	F2	ALU1	7	10
I2:	LD	F4,	0(Ry)		LD/SD	7	11
I3:	ADDD	F4a,	F0,	F4	ALU1	11	14
I4:	ADDD	F10,	F8,	F2a	ALU1	14	17
I5:	ADDI	Rxa,	Rx,	8	ALU0	1	2
I6:	ADDI	Rya,	Ry,	8	ALU1	1	2
I7:	SD	F4a,	0(Rya)		LD/SD	15	17
I8:	SD	F2a,	0(Rxa)		LD/SD	12	14
I9:	SUB	R20,	R4,	Rxa	ALU0	3	4
I10:	BNZ	R20,	Loop		ALU1	3	4

Following the execution above the micro architecture will execute the program in **17 Cycles**!

iii.

Instruction					IF Cycle	FU	Start Clock	Ending Clock
Loop:	LD	F2,	0(Rx)		0	LD/SD	1	6
I0:	DIVD	F8,	F2,	F0	0	ALU0	8	17
I1:	MULTD	F2a,	F6,	F2	1	ALU1	8	11
I2:	LD	F4,	0(Ry)		1	LD/SD	7	11
I3:	ADDD	F4a,	F0,	F4	2	ALU1	12	15
I4:	ADDD	F10,	F8,	F2a	2	ALU0	18	21
I5:	ADDI	Rxa,	Rx,	8	3	ALU0	4	5
I6:	ADDI	Rya,	Ry,	8	3	ALU1	4	5
I7:	SD	F4a,	0(Rya)		4	LD/SD	16	18
I8:	SD	F2a,	0(Rxa)		4	LD/SD	21	22
I9:	SUB	R20,	R4,	Rxa	5	ALU0	6	7
I10:	BNZ	R20,	Loop		5	ALU1	6	7

Using IF by each cycle, the micro architecture will execute the program in **22 Cycles**.