4DM4 Assignment 1
RISC Scheduling of the Chacha20 Stream Cipher

Ashpan Raskar raskara 400185326
Ahnaf Bhuiyan bhuiya3 400198359

October 24, 2022

Contents

Part A

A1

	Clock Cycle										
Instruction	1	2	3	4	5	6	7	8	9	10	
LW R2,0(R0)	**F1**	**F2**	**D**	**EX**	**M1**	**M2**	**WB**				
ADD R2, R2, R3		**F1**	**F2**	**D**	**STL**	**STL**	**EX**	**M1**	**M2**	**WB**	

A2

	Clock Cycle											
Instruction	1	2	3	4	5	6	7	8	9	10	11	12
BNEZ, R0, loop	**F1**	**F2**	**D**	**EX**	**M1**	**M2**	**WB**					
NO-OP		**F1**	**F2**	**D**	**EX**	**M1**	**M2**	**WB**				
NO-OP			**F1**	**F2**	**D**	**EX**	**M1**	**M2**	**WB**			
Next Iteration, LD				**F1**	**F2**	**D**	**EX**	**M1**	**M2**	**WB**		
Case2												
ADD R0,R0,R31	**F1**	**F2**	**D**	**EX**	**M1**	**M2**	**WB**					
BNEZ, R0, loop		**F1**	**F2**	**STL**	**D**	**EX**	**M1**	**M2**	**WB**			
NO-OP			**F1**	**STL**	**D**	**EX**	**M1**	**M2**	**WB**			
NO-OP					**F1**	**F2**	**D**	**EX**	**M1**	**M2**	**WB**	
Next Iteration, LD						**F1**	**F2**	**D**	**EX**	**M1**	**M2**	**WB**

Part B

B1

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
LOAD R1,0(R0)	F1	F2	D	EX	M1	M2	WB																							
LOAD R2,32(R0)		F1	F2	D	EX	M1	M2	WB																						
ADD R1, R1, R2			F1	F2	D	STL	STL	EX	M1	M2	WB																			
SW 0(R0), R1				F1	F2	STL	STL	D	EX	M1	M2	WB																		
LOAD R3,64(R0)					F1	STL	STL	F2	D	EX	M1	M2	WB																	
LOAD R5,0(R0)								F1	F2	D	EX	M1	M2	WB																
XOR R3, R3, R1									F1	F2	D	STL	STL	EX	M1	M2	WB													
SW 64(R0), R3										F1	F2	STL	STL	D	EX	M1	M2	WB												
LOAD R3,64(R0)											F1	STL	STL	F2	D	EX	M1	M2	WB											
ROT.L R3,16														F1	F2	STL	STL	D	EX	M1	M2	WB								
SW 64(R0), R3															F1	STL	STL	F2	D	EX	M1	M2	WB							

B2

Instruction	Number of Stalls	Comments
LOAD R1,0(R0)	0	Loading 'a' into register R1 from address 0+R0
LOAD R2,32(R0)	0	Loading 'b' into register R2 from address 32+R0
ADD R1, R1, R2	2	Adding 'a' to 'b' and storing the result back to R1
SW 0(R0), R1	2	Saving 'a' into adress 0+R0
LOAD R3,64(R0)	2	Loading 'd' into register R3 from address 64+R0
LOAD R4,0(R0)	0	Loading 'a' into register R4 from address 0+R0
XOR R3, R3, R4	2	XORing 'a' to 'd' and storing the result back to R3
SW 64(R0), R3	2	Saving 'd' into adress 64+R0
LOAD R5,64(R0)	2	Loading 'd' into register R5 from address 64+R0
ROT.L R5,16	2	Performing a cyclic bit shift on 'd' by 16 bits
SW 64(R0), R5	2	Saving 'd' into adress 64+R0
		End of first quarter round
LOAD R1,96(R0)	0	Loading 'c' into register R1 from address 96+R0
LOAD R2,64(R0)	0	Loading 'd' into register R2 from address 64+R0
ADD R1, R1, R2	2	Adding 'c' to 'd' and storing the result back to R1
SW 96(R0), R1	2	Saving 'c' into adress 96+R0
LOAD R3,32(R0)	2	Loading 'b' into register R3 from address 32+R0
LOAD R4,96(R0)	0	Loading 'c' into register R4 from address 96+R0
XOR R3, R3, R4	2	XORing 'b' to 'c' and storing the result back to R3
SW 32(R0), R3	2	Saving 'b' into adress 32+R0
LOAD R5,32(R0)	2	Loading 'b' into register R5 from address 32+R0
ROT.L R5,12	2	Performing a cyclic bit shift on 'b' by 12 bits
SW 32(R0), R5	2	Saving 'b' into adress 32+R0
		End of second quarter round
LOAD R1,0(R0)	0	Loading 'a' into register R1 from address 0+R0
LOAD R2,32(R0)	0	Loading 'b' into register R2 from address 32+R0
ADD R1, R1, R2	2	Adding 'a' to 'b' and storing the result back to R1
SW 0(R0), R1	2	Saving 'a' into adress 0+R0
LOAD R3,64(R0)	2	Loading 'd' into register R3 from address 64+R0
LOAD R4,0(R0)	0	Loading 'a' into register R4 from address 0+R0
XOR R3, R3, R4	2	XORing 'a' to 'd' and storing the result back to R3
SW 64(R0), R3	2	Saving 'd' into adress 64+R0
LOAD R5,64(R0)	2	Loading 'd' into register R5 from address 64+R0
ROT.L R5,16	2	Performing a cyclic bit shift on 'd' by 16 bits
SW 64(R0), R5	2	Saving 'd' into adress 64+R0
		End of third quarter round
LOAD R1,96(R0)	0	Loading 'c' into register R1 from address 96+R0
LOAD R2,64(R0)	0	Loading 'd' into register R2 from address 64+R0
ADD R1, R1, R2	2	Adding 'c' to 'd' and storing the result back to R1
SW 96(R0), R1	2	Saving 'c' into adress 96+R0
LOAD R3,32(R0)	2	Loading 'b' into register R3 from address 32+R0
LOAD R4,96(R0)	0	Loading 'c' into register R4 from address 96+R0
XOR R3, R3, R4	2	XORing 'b' to 'c' and storing the result back to R3
SW 32(R0), R3	2	Saving 'b' into adress 32+R0
LOAD R5,32(R0)	2	Loading 'b' into register R5 from address 32+R0
ROT.L R5,12	2	Performing a cyclic bit shift on 'b' by 12 bits
SW 32(R0), R5	2	Saving 'b' into adress 32+R0
		End of fourth quarter round

B3

It takes **92 clock cycles** to complete the quarter round operation of the Chacha20 stream cipher. The number of clock cycles is calculated by adding the number of clock cycles required to complete each of the four operations in the quarter round. Since one of the operations takes 23 clock cycles, the total number of clock cycles is $23 + 23 + 23 + 23 = 92$.

B4

Assumptions

- Assume 'a' is in R2
- Assume 'b' is in R31
- Assume 'd' is in R30

	Clock Cycle																									
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
LOAD R1,0(R0)	F1	F2	D	EX	M1	M2	WB																			
LOAD R2,32(R0)		F1	F2	D	EX	M1	M2	WB																		
LOAD R3,64(R0)			F1	F2	D	EX	M1	M2	WB																	
LOAD R4,96(R0)				F1	F2	D	EX	M1	M2	WB																
ADD R1, R1, R2					F1	F2	D	EX	M1	M2	WB															
XOR R3, R3, R1						F1	F2	D	EX	M1	M2	WB														
ROT.L R3,16							F1	F2	D	EX	M1	M2	WB													
ADD R4, R4, R3								F1	F2	D	EX	M1	M2	WB												
XOR R2, R2, R4									F1	F2	D	EX	M1	M2	WB											
ROT.L R2, 12										F1	F2	D	EX	M1	M2	WB										
ADD R1, R1, R2											F1	F2	D	EX	M1	M2	WB									
XOR R3, R3, R1												F1	F2	D	EX	M1	M2	WB								
ROT.L R3, 8													F1	F2	D	EX	M1	M2	WB							
ADD R4, R4, R3														F1	F2	D	EX	M1	M2	WB						
XOR R2, R2, R4															F1	F2	D	EX	M1	M2	WB					
ROT.L R2, 7																F1	F2	D	EX	M1	M2	WB				
SW 0(R0), R1																	F1	F2	D	EX	M1	M2	WB			
SW 32(R0), R2																		F1	F2	D	EX	M1	M2	WB		
SW 64(R0), R3																			F1	F2	D	EX	M1	M2	WB	
SW 96(R0), R4																				F1	F2	D	EX	M1	M2	WB

As seen in the above table, it takes **26 clock cycles** to complete the quarter round operation. The number of clock cycles is calculated by adding the number of clock cycles required to complete each of the four operations in the quarter round, the loading and saving.

B5

Block of Code	Clock Cycles	Comment
Optimized Quarter Round code block from B4. A, B, D, C are loaded at the beginning of the round, and are manipulated as need. Unlike in B4, they are not saved in this round. Words E, F, H, G are then loaded QR (0(R0),16(R0),32(R0),48(R0))	Clock Cycles:26 Stall Cycles: 0	A, B, D, C are words in the first column E, F, H, G are words in the second column
Optimized Quarter Round code where E, F, H, G are manipulated as need. They are not saved in this round. Words I, J, L, K are then loaded QR (4(R0),20(R0),36(R0),52(R0))	Clock Cycles:22 Stall Cycles: 0	I, J, L, K are words in the third column
Optimized Quarter Round code where I, J, L, K are manipulated as need. They are not saved in this round. Words M, N, P, O are then loaded QR (8(R0),24(R0),40(R0),56(R0))	Clock Cycles:22 Stall Cycles: 0	M, N, P, O are words in the fourth column
Optimized Quarter Round code where M, N, P, O are manipulated as need. At the end of this round, all previous words (A-O) are saved to the alloted memory locations QR (12(R0),28(R0),44(R0),60(R0))	Clock Cycles:28 Stall Cycles: 0	The addresses which these words are stored too are the same ones from the previous rounds, which the words are initially read from to avoid confusion and misplacement

98 clock cycles. 0 stalls.

B6

Block of Code	Stall Cycles	Comment
Outer-loop begins: LW #1023	0	Beginning of the outer loop, setting a counter to run 1023 times
Inner-loop begins: LW #9	0	Beginning of the inner loop, setting a counter to run 10 times
Odd Round code from B5	0	Using columns of the key-stream block
Even Round code from B5	0	Using diagonals of the key-stream block
BNEZ, R0, inner-loop	2*10 = 20	Ending of the inner loop. 2 stalls per loop cycle from NO-OP's
NO-OP		
NO-OP		
Next Iteration, LD		
BNEZ, R0, outer-loop	2*1024 = 2048	Ending of the outer loop. 2 stalls per loop cycle from NO-OP's
NO-OP		
NO-OP		
Next Iteration, LD		

Total clock cycles: 2131988

B7

Instruction Clock Cycles

104 CC — $\frac{1}{2}$ quarter round

208 CC — quarter round

2080 CC — double round

2080 CC × 1024 — 1024 blocks

= 2129920 CC

Stall clock Cycles

2 × 10 — inner loop

+2 × 1024 — outer loop

= 2068

Total CC = 2131988

2.5 GHz

$= \frac{2.5 \times 10^9 \text{ CC}}{1 \text{ second}}$

$2129920 \text{ CC} \times \frac{1 \text{ second}}{2.5 \times 10^9 \text{ CC}}$

$= 8.528 \times 10^{-4} = 85.28 \times 10^{-3}$ sec

$= 85.28$ ms