Computer Architecture Lab 01 - Q1

Yihang Qiu, 520030910155, 2021/11

1 Trial

Run the program with 5 (stored at `x3014`), **the result** (the value stored at `x3015`) **is "3"**.

Results corresponding to different value stored at `x3014` is as follows.

x3014	1	2	3	4	5	6	7	8
x3015	0	0	1	2	3	5	8	13

2 Expected Function of the Code

The expected function of the code is to output the n-th term of Fibonacci Sequence with 0 as the 1st term and 1 as the 2nd term. (n is the value stored at `x3014`.)

3 Bugs of the Code and Possible Solution

The first problem is the initialization of values of registers. **Using ADD instructions to reset the value of registers to 0 or 1 is faulty**. Chances are that before execution, the value of registers are not zero. Also, instructions adding nothing to a register with no following branches is meaningless.

The second problem is that when the input is 1 or 2, the output is always 0. **In other words, a branch is expected for $n \leq 1$ and $n \geq 2$.**

A possible solution is as follows.

1) Change the "ADD" instruction to "AND" instruction. Specially, to set R3 to be 1, we can use "ADD R3, R2, #1". (These changes are underlined in the corrected code.)

2) Considering when $n \leq 1$, the result should be 0 and the condition code at `x3011` is negative, change the initialization of R4 to "ADD R4, R2, R3" and change the STI instruction into a branch. **(These changes are in bold type.)**

* Note that in this case, a new line is introduced and thus we need to change some of the immediate numbers and the value stored in the memory used in STI instruction. (These collateral changes are marked in red.)

* The input is now stored at `x3017` and the result is now stored at `x3018`.

The corrected machine code is as follows.

3000	**Comments:**		
E216	x3000	LEA;	R1←MEM[x3017]
6240	x3001	LDR;	R1←MEM[R1+0]
54A0	x3002	AND;	R2←R2 AND 0
16A1	x3003	ADD;	R3←R2+1
1883	x3004	**ADD;**	**R4←R2+R3**
5B60	x3005	AND;	R5←R5 AND 0
127E	x3006	ADD;	R1←R1+(-2)
0C09	x3007	BRnz;	BRnz [x3011]
9C7F	x3008	NOT;	R6←NOT R1
1DA1	x3009	ADD;	R6←R6+1
1D85	x300a	ADD;	R6←R6+R5
0405	x300b	BRz ;	BRz [x3011]
1883	x300c	ADD;	R4←R2+R3
1B61	x300d	ADD;	R5←R5+1
14E0	x300e	ADD;	R2←R3+0
1720	x300f	ADD;	R3←R4+0
0FF7	x3010	BRnzp;	BRnzp [x3008]
0802	x3011	**BRn ;**	**BRn [x3014]**
B803	x3012	**STI;**	**R4→MEM[MEM[x3016]]**
F025	x3013	**TRAP;**	**HALT**
B401	x3014	**STI;**	**R2→MEM[MEM[x3016]]**
F025	x3015	TRAP;	HALT
3018	x3016		
0005	x3017		
0000	x3018		