Name: Cameron Peterson-Zopf

PROGRAMMING ASSIGNMENT 2

Discussion on a high level with your colleagues is encouraged. Make sure the work submitted is your own. When in doubt, ask a TA or the instructor. If you are not sure what constitutes academic dishonesty, please refer to the AISC web site: https://aisc.uci.edu/.

You can fill out your answers below in text, paste screenshots, and/or include images (make sure the image is right side up & legible).

This homework covers:

- LC-3 Machine Language to Assembly – Part 1

AISC

Please initial here to indicate you understand UCI's Academic Integrity Policy and confirm that this is your own work you are submitting (this counts for points): CPZ

UPDATED CODE SCREENSHOT
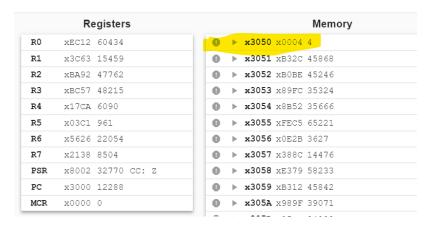
```
                                      power2.bin

 1  ; Purpose: Figure out if a number (positive 2's complement integer) is a power of 2 => only one 1 bit.
 2  ; Code starts @ x3000
 3  ; Input: positive two's complement number stored in x3050 (doesn't handle 0 case properly)
 4  ; Output: 1 @ x3051 = power of two; 0 @ x3051 = not a power of two
 5  ;
 6
 7  0011 0000 0000 0000      ;starting address of x3000
 8
 9  0101 010 010 1 00000     ;AND, DR=R2, SR=R2, #0     // R2<-0
10  1010 000 000000111       ;LDI, DR=R0, offset=#7     // R0<-M[M[x3009]] == R0<-M[x3050] == R0<-(#)
11  0000 010 000000100       ;BR, NZP=010, off=#4       // If R0 = 0, store right away (not a power of 2)
12  0001 001 000 1 11111     ;ADD, DR=R1, SR=R0, -#1    // R1<-R0-#1
13  0101 000 000 0 00 001    ;AND, DR=R0, SR=R0, SR=R1  // R0<-R0 AND R1
14  0000 101 000000001       ;BR, NZP=101, off=#1       // If R0 not 0, skip next line (not power of 2)
15  0001 010 010 1 00001     ;ADD, DR=R2, SR=R2, #1     // R2<-R2+#1 == R2<-1 (is power of 2)
16  1011 010 000000010       ;STI, SR=R2, off=#2        // M[M[x300A]]<-R2 == M[x3051]<-R2
17  1111 0000 0010 0101      ;TRAP, x25                 // HALT
18
19  ;hardcoded values
20  0011 0000 0101 0000      ;Address x3009: value x3050
21  0011 0000 0101 0001      ;Address x300A: value x3051
```

BE SURE TO SHOW VALUES @ X3050 & X3051 AS WELL AS REGISTERS JUST BEFORE HALT
INSTRUCTION

First test 4, which is a power of 2

Registers		Memory	
R0	xEC12 60434	▶ **x3050** x0004 4	
R1	x3C63 15459	▶ **x3051** xB32C 45868	
R2	xBA92 47762	▶ **x3052** xB0BE 45246	
R3	xBC57 48215	▶ **x3053** x89FC 35324	
R4	x17CA 6090	▶ **x3054** x8B52 35666	
R5	x03C1 961	▶ **x3055** xFEC5 65221	
R6	x5626 22054	▶ **x3056** x0E2B 3627	
R7	x2138 8504	▶ **x3057** x388C 14476	
PSR	x8002 32770 CC: Z	▶ **x3058** xE379 58233	
PC	x3000 12288	▶ **x3059** xB312 45842	
MCR	x0000 0	▶ **x305A** x989F 39071	

R2 = 1 as we wanted.

Registers		Memory	
R0	x0000 0	▶ **x3000** x54A0 21664	0101010010100000
R1	x0003 3	▶ **x3001** xA007 40967	1010000000000111
R2	x0001 1	▶ **x3002** x0402 1026	0000010000000010
R3	xBC57 48215	▶ **x3003** x123F 4671	0001001000111111
R4	x17CA 6090	▶ **x3004** x5001 20481	0101000000000001
R5	x03C1 961	▶ **x3005** x0A01 2561	0000101000000001
R6	x5626 22054	▶ **x3006** x14A1 5281	0001010010100001
R7	x2138 8504	▶ **x3007** xB402 46082	1011010000000010
PSR	x8001 32769 CC: P	▶ **x3008** xF025 61477	1111000000100101
PC	x3008 12296	▶ **x3009** x3050 12368	0011000001010000
MCR	x0000 0	▶ **x300A** x3051 12369	0011000001010001

The value stored in x3051 is 1, indicating it is a power of 2.

Registers		Memory	
R0	x0000 0	▶ **x3050** x0004 4	
R1	x0003 3	▶ **x3051** x0001 1	
R2	x0001 1	▶ **x3052** xB0BE 45246	
R3	xBC57 48215	▶ **x3053** x89FC 35324	
R4	x17CA 6090	▶ **x3054** x8B52 35666	
R5	x03C1 961	▶ **x3055** xFEC5 65221	
R6	x5626 22054	▶ **x3056** x0E2B 3627	
R7	x2138 8504	▶ **x3057** x388C 14476	
PSR	x8001 32769 CC: P	▶ **x3058** xE379 58233	
PC	x3008 12296	▶ **x3059** xB312 45842	
MCR	x0000 0	▶ **x305A** x989F 39071	

BE SURE TO SHOW VALUES @ X3050 & X3051 AS WELL AS REGISTERS JUST BEFORE HALT INSTRUCTION

Next test 5, which is not a power of 2

R2 = 0 as we wanted.

The value stored in x3051 is 0, indicating it is not a power of 2.

BE SURE TO SHOW VALUES @ X3050 & X3051 AS WELL AS REGISTERS JUST BEFORE HALT INSTRUCTION

Finally test 0, which is not a power of 2
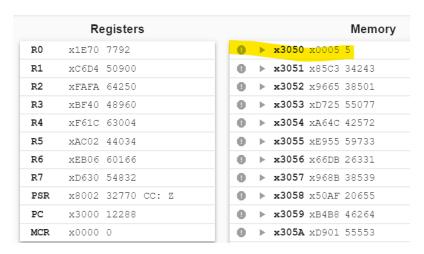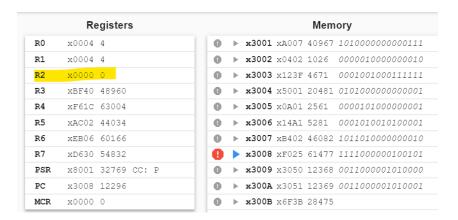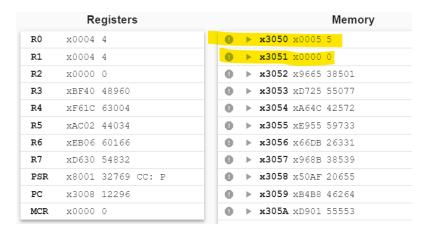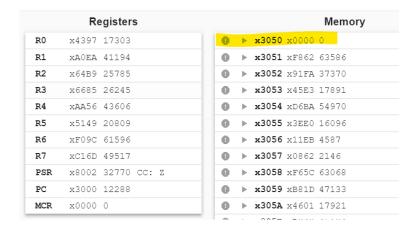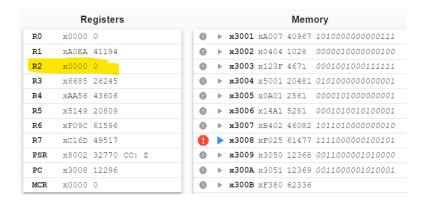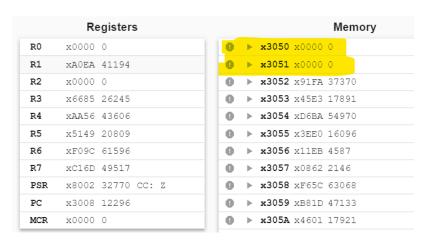
Registers		Memory	
R0	x4397 17303	x3050 x0000 0	
R1	xA0EA 41194	x3051 xF862 63586	
R2	x64B9 25785	x3052 x91FA 37370	
R3	x6685 26245	x3053 x45E3 17891	
R4	xAA56 43606	x3054 xD6BA 54970	
R5	x5149 20809	x3055 x3EE0 16096	
R6	xF09C 61596	x3056 x11EB 4587	
R7	xC16D 49517	x3057 x0862 2146	
PSR	x8002 32770 CC: Z	x3058 xF65C 63068	
PC	x3000 12288	x3059 xB81D 47133	
MCR	x0000 0	x305A x4601 17921	

R2 = 0 as we wanted.

Registers		Memory	
R0	x0000 0	x3001 xA007 40967 1010000000000111	
R1	xA0EA 41194	x3002 x0404 1028 0000010000000100	
R2	x0000 0	x3003 x123F 4671 0001001000111111	
R3	x6685 26245	x3004 x5001 20481 0101000000000001	
R4	xAA56 43606	x3005 x0A01 2561 0000101000000001	
R5	x5149 20809	x3006 x14A1 5281 0001010010100001	
R6	xF09C 61596	x3007 xB402 46082 1011010000000010	
R7	xC16D 49517	x3008 xF025 61477 1111000000100101	
PSR	x8002 32770 CC: Z	x3009 x3050 12368 0011000001010000	
PC	x3008 12296	x300A x3051 12369 0011000001010001	
MCR	x0000 0	x300B xF380 62336	

The value stored in x3051 is 0, indicating it is not a power of 2.

Registers		Memory	
R0	x0000 0	x3050 x0000 0	
R1	xA0EA 41194	x3051 x0000 0	
R2	x0000 0	x3052 x91FA 37370	
R3	x6685 26245	x3053 x45E3 17891	
R4	xAA56 43606	x3054 xD6BA 54970	
R5	x5149 20809	x3055 x3EE0 16096	
R6	xF09C 61596	x3056 x11EB 4587	
R7	xC16D 49517	x3057 x0862 2146	
PSR	x8002 32770 CC: Z	x3058 xF65C 63068	
PC	x3008 12296	x3059 xB81D 47133	
MCR	x0000 0	x305A x4601 17921	