Name: Cameron Peterso-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.asm
 1  ;Assembly Version of Code to Figure out if a number (positive 2's complement integer)
 2  ;is a power of 2 => only one 1 bit.
 3  ; Code starts @ x3000
 4  ;Input: positive two's complement number stored in x3050 (doesn't handle 0 case properly)
 5  ; Output: 1 @ x3051 = power of two; 0 @ x3051 = not a power of two
 6
 7          .ORIG   x3000
 8          AND     R2, R2, #0  ; R2<-0
 9          LDI     R0, INPUT   ;R0<-M[x3050] == R0<-(#)
10          BRz     #4          ; If R0 = 0, store right away (not a power of 2)
11          ADD     R1, R0, #-1 ; R1<-R0-#1
12          AND     R0, R0, R1  ; R0<-R0 AND R1
13          BRnp    #1          ; If R0 not 0, skip next line (not power of 2)
14          ADD     R2, R2, #1  ; R2<-R2+#1 == R2<-1 (is power of 2)
15          STI     R2, OUTPUT  ; M[x3051]<-R2
16          HALT
17
18          ;hardcoded values
19  INPUT   .FILL   x3050
20  OUTPUT  .FILL   x3051
21
22          .END
```

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

First test 8, which is a power of 2

Registers		Memory	
R0	xC4C7 50375	⚠ ▶ **x3050** x0008 8	
R1	xACDA 44250	⚠ ▶ **x3051** x71D3 29139	
R2	x9CCD 40141	⚠ ▶ **x3052** x3C5A 15450	
R3	xBD47 48455	⚠ ▶ **x3053** x74A8 29864	
R4	xA17E 41342	⚠ ▶ **x3054** xFD3E 64830	
R5	xB850 47184	⚠ ▶ **x3055** x6784 26500	
R6	xC8B5 51381	⚠ ▶ **x3056** xEB83 60291	
R7	x25E2 9698	⚠ ▶ **x3057** x436F 17263	
PSR	x8002 32770 CC: Z	⚠ ▶ **x3058** x7B41 31553	
PC	x3000 12288	⚠ ▶ **x3059** x1984 6532	
MCR	x0000 0	⚠ ▶ **x305A** x9D3D 40253	

R2 = 1 as we wanted.

Registers		Memory	
R0	x0000 0	⚠ ▶ **x3001** xA007 40967 *LDI R0, INPUT*	
R1	x0007 7	⚠ ▶ **x3002** x0404 1028 *BRz #4*	
R2	x0001 1	⚠ ▶ **x3003** x123F 4671 *ADD R1, R0, #-1*	
R3	xBD47 48455	⚠ ▶ **x3004** x5001 20481 *AND R0, R0, R1*	
R4	xA17E 41342	⚠ ▶ **x3005** x0A01 2561 *BRnp #1*	
R5	xB850 47184	⚠ ▶ **x3006** x14A1 5281 *ADD R2, R2, #1*	
R6	xC8B5 51381	⚠ ▶ **x3007** xB402 46082 *STI R2, OUTPUT*	
R7	x25E2 9698	⚠ ▶ **x3008** xF025 61477 *HALT*	
PSR	x8001 32769 CC: P	⚠ ▶ **x3009** x3050 12368 *INPUT .FILL x3050*	
PC	x3008 12296	⚠ ▶ **x300A** x3051 12369 *OUTPUT .FILL x3051*	
MCR	x0000 0	⚠ ▶ **x300B** xBFF7 49143	

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

Registers		Memory	
R0	x0000 0	⚠ ▶ **x3050** x0008 8	
R1	x0007 7	⚠ ▶ **x3051** x0001 1	
R2	x0001 1	⚠ ▶ **x3052** x3C5A 15450	
R3	xBD47 48455	⚠ ▶ **x3053** x74A8 29864	
R4	xA17E 41342	⚠ ▶ **x3054** xFD3E 64830	
R5	xB850 47184	⚠ ▶ **x3055** x6784 26500	
R6	xC8B5 51381	⚠ ▶ **x3056** xEB83 60291	
R7	x25E2 9698	⚠ ▶ **x3057** x436F 17263	
PSR	x8001 32769 CC: P	⚠ ▶ **x3058** x7B41 31553	
PC	x3008 12296	⚠ ▶ **x3059** x1984 6532	
MCR	x0000 0	⚠ ▶ **x305A** x9D3D 40253	

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

Next test 100, which is not a power of 2

Registers		Memory	
R0	x6337 25399	⚠ ▶ **x3050** x0064 100	
R1	x8BFA 35834	⚠ ▶ **x3051** x1D11 7441	
R2	x41B4 16820	⚠ ▶ **x3052** xB8F1 47345	
R3	xD73E 55102	⚠ ▶ **x3053** xE9B5 59829	
R4	x50A0 20640	⚠ ▶ **x3054** x0C1D 3101	
R5	xD2BE 53950	⚠ ▶ **x3055** x7DCA 32202	
R6	xD81C 55324	⚠ ▶ **x3056** xBD1A 48410	
R7	x66BB 26299	⚠ ▶ **x3057** x052D 1325	
PSR	x8002 32770 CC: Z	⚠ ▶ **x3058** x2322 8994	
PC	x3000 12288	⚠ ▶ **x3059** x33F9 13305	
MCR	x0000 0	⚠ ▶ **x305A** xF5BF 62911	

R2 = 0 as we wanted.

Registers		Memory	
R0	x0060 96	⚠ ▶ **x3001** xA007 40967	*LDI R0, INPUT*
R1	x0063 99	⚠ ▶ **x3002** x0404 1028	*BRz #4*
R2	x0000 0	⚠ ▶ **x3003** x123F 4671	*ADD R1, R0, #-1*
R3	xD73E 55102	⚠ ▶ **x3004** x5001 20481	*AND R0, R0, R1*
R4	x50A0 20640	⚠ ▶ **x3005** x0A01 2561	*BRnp #1*
R5	xD2BE 53950	⚠ ▶ **x3006** x14A1 5281	*ADD R2, R2, #1*
R6	xD81C 55324	⚠ ▶ **x3007** xB402 46082	*STI R2, OUTPUT*
R7	x66BB 26299	❗ ▶ **x3008** xF025 61477	*HALT*
PSR	x8001 32769 CC: P	⚠ ▶ **x3009** x3050 12368	*INPUT .FILL x3050*
PC	x3008 12296	⚠ ▶ **x300A** x3051 12369	*OUTPUT .FILL x3051*
MCR	x0000 0	⚠ ▶ **x300B** xE70A 59146	

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

Registers		Memory	
R0	x0060 96	⚠ ▶ **x3050** x0064 100	
R1	x0063 99	⚠ ▶ **x3051** x0000 0	
R2	x0000 0	⚠ ▶ **x3052** xB8F1 47345	
R3	xD73E 55102	⚠ ▶ **x3053** xE9B5 59829	
R4	x50A0 20640	⚠ ▶ **x3054** x0C1D 3101	
R5	xD2BE 53950	⚠ ▶ **x3055** x7DCA 32202	
R6	xD81C 55324	⚠ ▶ **x3056** xBD1A 48410	
R7	x66BB 26299	⚠ ▶ **x3057** x052D 1325	
PSR	x8001 32769 CC: P	⚠ ▶ **x3058** x2322 8994	
PC	x3008 12296	⚠ ▶ **x3059** x33F9 13305	
MCR	x0000 0	⚠ ▶ **x305A** xF5BF 62911	

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

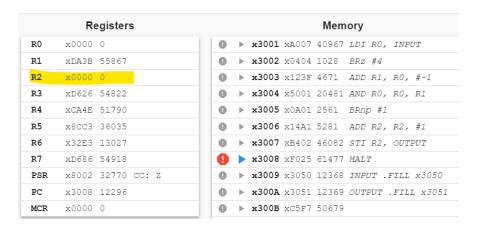| Registers		Memory	
R0	x2BDA 11226	⚠ ▶ **x3050** x0000 0	
R1	xDA3B 55867	⚠ ▶ **x3051** xC65F 50783	
R2	x6BD9 27609	⚠ ▶ **x3052** x510D 20749	
R3	xD626 54822	⚠ ▶ **x3053** xD38C 54156	
R4	xCA4E 51790	⚠ ▶ **x3054** x24B5 9397	
R5	x8CC3 36035	⚠ ▶ **x3055** xF34D 62285	
R6	x32E3 13027	⚠ ▶ **x3056** x3905 14597	
R7	xD686 54918	⚠ ▶ **x3057** x5AA1 23201	
PSR	x8002 32770 CC: Z	⚠ ▶ **x3058** x8568 34152	
PC	x3000 12288	⚠ ▶ **x3059** x35DC 13788	
MCR	x0000 0	⚠ ▶ **x305A** xBF2D 48941	

R2 = 0 as we wanted.

Registers		Memory	
R0	x0000 0	⚠ ▶ **x3001** xA007 40967 *LDI R0, INPUT*	
R1	xDA3B 55867	⚠ ▶ **x3002** x0404 1028 *BRz #4*	
R2	x0000 0	⚠ ▶ **x3003** x123F 4671 *ADD R1, R0, #-1*	
R3	xD626 54822	⚠ ▶ **x3004** x5001 20481 *AND R0, R0, R1*	
R4	xCA4E 51790	⚠ ▶ **x3005** x0A01 2561 *BRnp #1*	
R5	x8CC3 36035	⚠ ▶ **x3006** x14A1 5281 *ADD R2, R2, #1*	
R6	x32E3 13027	⚠ ▶ **x3007** xB402 46082 *STI R2, OUTPUT*	
R7	xD686 54918	❗ ▶ **x3008** xF025 61477 *HALT*	
PSR	x8002 32770 CC: Z	⚠ ▶ **x3009** x3050 12368 *INPUT .FILL x3050*	
PC	x3008 12296	⚠ ▶ **x300A** x3051 12369 *OUTPUT .FILL x3051*	
MCR	x0000 0	⚠ ▶ **x300B** xC5F7 50679	

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

Registers		Memory	
R0	x0000 0	⚠ ▶ **x3050** x0000 0	
R1	xDA3B 55867	⚠ ▶ **x3051** x0000 0	
R2	x0000 0	⚠ ▶ **x3052** x510D 20749	
R3	xD626 54822	⚠ ▶ **x3053** xD38C 54156	
R4	xCA4E 51790	⚠ ▶ **x3054** x24B5 9397	
R5	x8CC3 36035	⚠ ▶ **x3055** xF34D 62285	
R6	x32E3 13027	⚠ ▶ **x3056** x3905 14597	
R7	xD686 54918	⚠ ▶ **x3057** x5AA1 23201	
PSR	x8002 32770 CC: Z	⚠ ▶ **x3058** x8568 34152	
PC	x3008 12296	⚠ ▶ **x3059** x35DC 13788	
MCR	x0000 0	⚠ ▶ **x305A** xBF2D 48941	