**Program 2- B17**

**Submitted by Marcus Berger, Kevin Hilt, and Evan Hammer**

**Program Description:**

**Libraries used:**

Libraries used in this program are :  
 <iostream>, <iomanip>, <string>, <sstream>,<fstream>, and <cstdlib>

**Algorithm:**

1. Read in the object file and set up the memory array
2. Read in instruction counter and the number of instruction to be executed
3. For each instruction begin to execute instruction
   1. Decode the instruction and determine the address mode and instruction to be executed.
   2. Execute the correct instruction
4. Check the address mode to get the correct output
5. Output the information for executed instruction
6. Repeat steps 3-5 for each instruction
7. If an error is encountered the correct halting message will be printed

**Functions and Program Structure:**

**Compiling and Usage:**

Compile by typing “make b17” in Linux

Usage: b17 “object file name”

Sample object file:

50 1 000000  
c4 5 050404 200800 300800 102840 050c00  
101 2 300 9   
200 1 30   
300 1 10   
c4

Output file will contain memory address, instructions, the instruction executed, the address mode, that contains of the accumulator and the contains of registers x0-x3 for each instruction executed. When a halt instruction or halt error is encountered a correct message is printed and the program terminates.

Sample output file:  
0c4: 050404 LD IMM AC[000050] X0[000] X1[000] X2[000] X3[000]   
0c5: 200800 ADD 200 AC[000080] X0[000] X1[000] X2[000] X3[000]  
0c6: 300800 ADD 300 AC[000090] X0[000] X1[000] X2[000] X3[000]  
0c7: 102840 SUB 102 AC[000087] X0[000] X1[000] X2[000] X3[000]  
0c8: 050c00 J 050 AC[000087] X0[000] X1[000] X2[000] X3[000]   
050: 000000 HALT AC[000087] X0[000] X1[000] X2[000] X3[000]  
Machine Halted - HALT instruction executed

**Testing:**

Test 1: Test ADD, SUB, Jump, Load, and Halt instructions with immediate and direct addressing modes  
input:  
50 1 000000  
c4 5 050404 200800 300800 102840 050c00  
101 2 300 9  
200 1 30  
300 1 10  
c4  
Output:  
0c4: 050404 LD IMM AC[000050] X0[000] X1[000] X2[000] X3[000]  
0c5: 200800 ADD 200 AC[000080] X0[000] X1[000] X2[000] X3[000]  
0c6: 300800 ADD 300 AC[000090] X0[000] X1[000] X2[000] X3[000]  
0c7: 102840 SUB 102 AC[000087] X0[000] X1[000] X2[000] X3[000]  
0c8: 000000 J 50 AC[000087] X0[000] X1[000] X2[000] X3[000]  
050: 000000 HALT AC[000087] X0[000] X1[000] X2[000] X3[000]  
Machine Halted - HALT instruction executed

Text 2: Test illegal addressing mode  
input:  
50 1 000000  
100 5 200400 201840 202800 005844 050c84  
200 3 30 31 10  
100

Output:  
100: 200400 LD 200 AC[000030] X0[000] X1[000] X2[000] X3[000]  
101: 201840 SUB 201 AC[ffffff] X0[000] X1[000] X2[000] X3[000]  
102: 202800 ADD 202 AC[00000f] X0[000] X1[000] X2[000] X3[000]  
103: 005844 SUB IMM AC[00000a] X0[000] X1[000] X2[000] X3[000]  
104: 050c84 JN ??? AC[00000a] X0[000] X1[000] X2[000] X3[000]  
Machine Halted - illegal addressing mode.

Test 3: Test clear instruction and unimplemented op code  
Input:  
50 1 000000  
ff 7 075400 040804 077440 005884 076400 077800 030a04  
75 3 30 20 10  
ff

Output:  
0ff: 075400 LD 75 AC[000030] X0[000] X1[000] X2[000] X3[000]  
100: 040804 ADD IMM AC[000070] X0[000] X1[000] X2[000] X3[000]  
101: 077440 ST 77 AC[000070] X0[000] X1[000] X2[000] X3[000]  
102: 005884 CLR IMM AC[000000] X0[000] X1[000] X2[000] X3[000]  
103: 076400 LD 76 AC[000020] X0[000] X1[000] X2[000] X3[000]  
104: 077800 ADD 77 AC[000090] X0[000] X1[000] X2[000] X3[000]  
105: 030a04 ADDX IMM AC[000090] X0[000] X1[000] X2[000] X3[000]  
Machine Halted - unimplemented opcode

**What was submitted?**

b17.cpp – contains the main function for the b17 program

b17\_functions.cpp – contains the get\_instruction, get\_addressmode, hex\_to\_int, read\_memoy, and match\_instruction functions to prevent needing to repeat code,

b17.h – header file containing libraries used and function prototypes for the b17 program

makefile – complies the b17 program by typing “make b17”