John Mangold

Andrew Housh

Jiasong Yan

CSC 317 – Prog2 – B\_17

5/1/2015

b17.cpp

// description of program

This program is a simulation of a b17 assembly processor. It will read in a file and perform the necessary action. It will recognize improper addressing modes and halt the program.

// description of the algorithms and libraries used

**<iostream> //**Used for output to the console

**<string> //**Used to handle and process inputs from file.

**<vector>** //Used to simulate memory.

**<fstream>** //To handle input files.

**<stdio.h>** //Used for printf

**<sstream>** //Used to convert decimal to hex string.

**<iomanip> //Used for hex format print out**

**<climits> //Call INT\_MAX from this library**

// description of functions and program structure

**Int main( int argc, char\*\* argv )**

Handles necessary function calls

**string parse\_file( ifstream &fin, vector<string> &mem );**

Reads through input file one line at a time and places information into a vector used to simulate memory.

**string hex\_to\_bin( string hex );**

Changes a hex number string to a binary number string.

**string bin\_to\_hex( string bin );**

Changes a binary number string to a hex number string.

**void split\_instruction( string instruction, string &hex, string &lastThree );**

Takes a string and splits it into two separate strings.

**void get\_op\_and\_am( string lastThree, string &opcode, string &addressMode );**

Changes three digit hex opcode to binary then splits it into the operator and address mode.

**void print\_line( string address, string instruction, string op\_name, string address\_mode, string accumulator\_hex );**

Prints a properly formatted line to the screen.

**void action( vector<string> &memory, stirng op, string &op\_name, string address\_mode, string &accumulator, string operand\_address, string &address, string mem\_address );**

This function will the actual simulation of the b17 assembly processor. It will read each opcode and execute them appropriately and use whatever information that is stored in memory.

// how to compile and use the program

**Compile:**

g++ -std=c++11 –c -Wall

**Usage:**

Typing ./b17 and a input file will invoke the program. No other input is necessary to simulate the assembly processor.

// description of the testing and verification process

Used all test files that was provide by Dr. Karlsson to simulate the b17 assembly processor to make sure it matches the output.

// description of Makefile

The makefile provides the basic routine to create an executable for b17.cpp.