Disassembler Project

UWB - CSS 422 – Winter 2019

By: Amy Meyers, Mariah Files, Mustafa Majeed

--

Project Description

This program is written to scan sections of memory and attempt to convert the contents in memory to a valid string that can be used as assembly language code for the 68K chip.

The section of memory is determined by the contents of a .cfg file that will be placed in the same directory as the program. Please see the cfg file specifications on page 2 for formatting.

The program parses the op-code word of the instruction at a specified space in memory and then decides how many additional words of memory need to be read in order to complete the instruction, then proceeds to read everything that is needed to extract a valid 68K instruction.

For each valid 68K machine code instruction that resides in memory, this program will print out a complete instruction in an ASCII readable format.

The printed string will follow the following format for instructions recognized by the program:

At: XXXX (CORRECT 68K INSTRUCTION) (CORRECT E/A) YYYY

XXXX = Memory Address

YYYY = Additional Word or Long Absolute Memory Location

If the instruction is not recognized by the program, the printed string will be formatted as follows:

At: XXXX DATA YYYY

XXXX = Memory Address

YYYY = Hex value of the unrecognizable instruction

LIST OF RECOGNIZABLE OPCODES AND EA

OP codes

MOVE, MOVEA, MOVEM	OR, ORI	ROL, ROR
ADD, ADDA	NEG	BCLR
SUB, SUBQ	EOR	CMP, CMPI
MULS, DIVS	LSR, LSL	Bcc (BCS, BGE, BLT, BVC)
LEA	ASR, ASL	BRA, JSR, RTS

EA Modes

Data Register Direct	Dn	Address Register Indirect w/ Post Increment	(An)+
Address Register Direct	An	Address Register Indirect w/ Pre Decrement	-(An)
Address Register Indirect	(An)	Absolute Long Address	(xxx).L
Immediate Addressing	#xxx	Absolute Word Address	(xxx).W

CFG FILE SPECIFICATIONS

The .cfg file that will contain the starting and ending address for this program to scan must be name "Config.cfg". It must follow DOS/Windows machine line endings (\r\n) and contain Long addresses for both start and end.

Program Flow Chart