Team Avengers 68K Disassembler Progress Report

Date: May 7th, 2020

To: CSS 422, Spring 2020

From: Group Avengers [Mohammed Ali, Dj Wadhwa, Tarcisius Hartanto]

Subject: 68k Disassembler Progress Report 2

Work Completed:

We started by splitting up the assignment based on the Opcodes, IO, and Effective addresses. Mohammed worked on reading the input from the user, translating it to HEX values that can be read by the program. Additionally, he set up the environment so the program starts reading the code linearly. DJ worked on creating subroutines that would read the MOVE, MOVEA, ADD, and SUB instruction. This was done by bit shifting and masking values in memory and detecting which opcode belongs to the machine code that is read. This was accomplished by implementing a switch case equivalent code in assembly using BCC statements. Tarcisius began working on reading and understanding whether the code provided was an opcode or an effective address. If it was an effective address his subroutine would relay that back upwards, so the opcode translation can be finished completely.

Opcode Machine Code Diagram:

Opcode	Machine code
NOP	%0100111001110001 = $2771
MOVE	%00(byte: 01, word: 11, long: 10)(Destination address)(Destination Mode)(Source Mode)(Source Address)
MOVEA	%00(byte: 01, word: 11, long: 10)(Destination address register)001(Source Mode)(Source Address)

MOVEM	%01001(Reg → mem: 0, mem → reg: 1)001(word: 0, long: 1) (Source mode) (source address)
MOVEQ	%0111 (Dn) 0 (DATA)
ADD	%1101 (Dn) (Opmode) (mode) (Xn)
ADDA	%1101 (An) (Opmode) (mode) (Xn)
ADDQ	%0101 (DATA) 0 (size) (mode) (Xn)
SUB	%1001 (Destination Data Register)(Dn * ea → Dn: 0, ea * Dn → es: 1) (byte: 00, word: 01, long: 10) (Source mode) (Source address)
LEA	%0100 (An) 111 (mode) (Xn)
AND	(Size 01, 11, or 10) %1100
OR	%1000(Destination Data Register)(Dn * ea → Dn: 0, ea * Dn → es: 1) (byte: 00, word: 01, long: 10) (Source mode) (Source address)
NOT	%01000110(byte: 00, word: 01, long: 10)(Source mode) (Source address)
LSR	Assuming data registers: %1110 (Rotation amount)0(byte: 00, word: 01, long: 10)(Immediate:0, register:1)01(Destination data register)
LSL	Assuming data registers: %1110 (Rotation amount)1(byte: 00, word: 01, long: 10)(Immediate:0, register:1)01(Destination data register)
ASR	Assuming data registers: %1110 (Rotation)0(byte: 00, word: 01, long: 10)(Immediate:0, register:1)00(Destination data register)
ASL	Assuming data registers: %1110 (Rotation)1(byte: 00, word: 01, long: 10)(Immediate:0, register:1)00(Destination data register)
Bcc (BGT,	BGT:%01101110 (Displacement: upto 8 bits)

BLE, BEQ)	BLE: %01101111(Displacement: upto 8 bits) BEQ: %01100111 (Displacement: upto 8 bits)
JSR	&0100111010(Source mode) (Source address)
RTS	&0100111001110101 = $4E75
BRA	&01100000 (Displacement: upto 8 bits)

Problems:

So far there have not been any huge problems. We have started with the I/O section which is good because it means we have something started. We really have to focus and double down on getting everything else done. We have not seriously begun the implementation of the project, but now that we have some familiarity with assembly we can delve into the more serious parts of this project.

Work Scheduled:

1. Create more detailed models for processes involved in the project **(in progress)**
2. Begin I/O section **(in progress)**
    a. Define ASCII ID's **(in progress)**
    b. Implement ASCII processing (string to hex, hex to string) **(in progress)**
3. Implementation of the opcode **(in progress)**
4. Finish I/O section **(Deadline 5/15/2020)**
5. Implementation of effective addressing **(in progress)**
6. Merge all portions together **(5/27)**
7. Testing and Debugging **(5/27-6/01)**
8. Documentation **(5/27 - 6/01)**
9. Create Presentation **(6/02)**

Evaluation:

Now that midterms are over we feel like we can finally get started on this project. We also feel as if we now have a deeper knowledge and a better understanding of the process to complete the disassembler. We still feel as if there are some essential things to learn but we now have a grasp on the foundational topics to get started.