Date: 05/09/2020
To: CSS 422, Spring 2020
From: Yang Hu, Matt Johnson, Jun H Park
Subject: [Group 14] Report on the Initial Progress

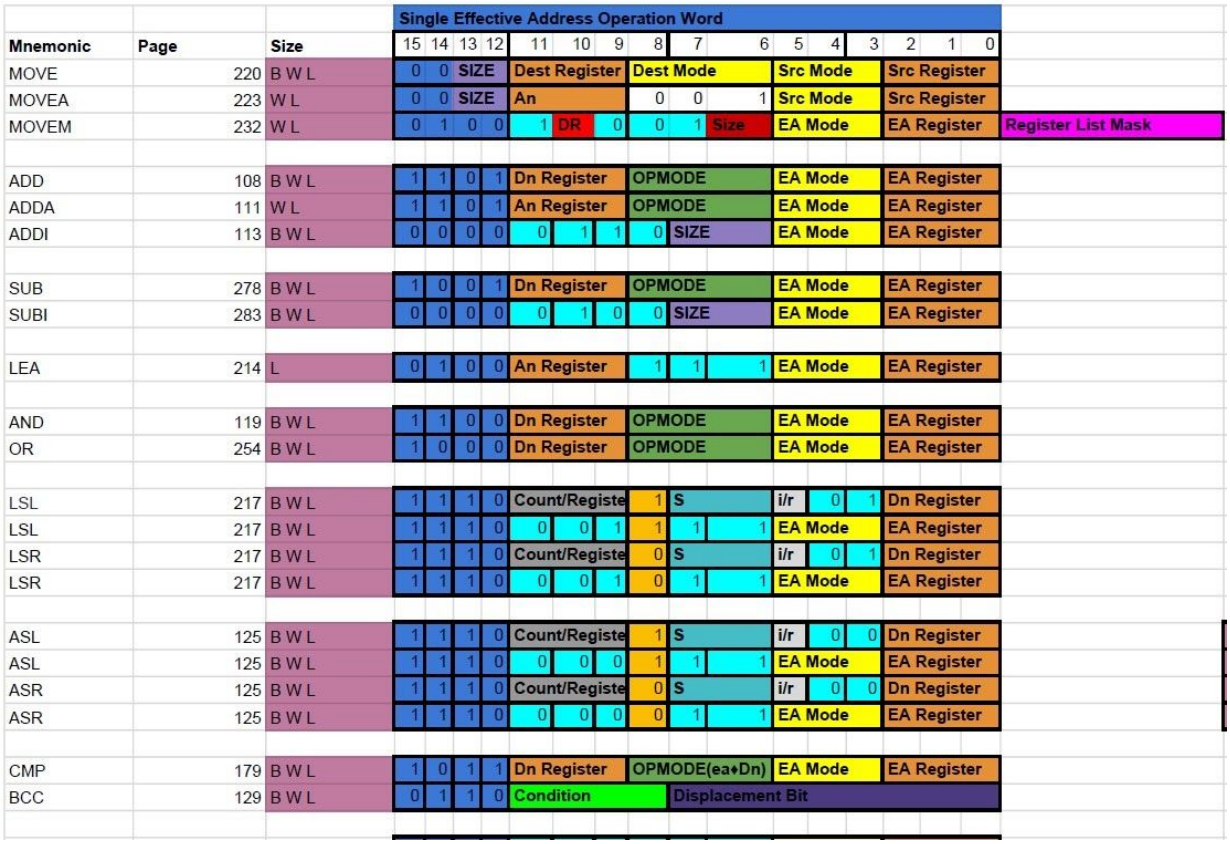
Meeting time and location:
We had a small meeting on Thursday at 5 pm for this assignment by Discord Voice Chat.
A further meeting has to be arranged.

Work Completed:
During the initial progress, group 14 had accomplished the following requirements.

Requirements:

1. The table that matches all the required opcode/EA with the machine code
2. Simple flow-chart

Figure 1.1: Opcodes



			Single Effective Address Operation Word															
Mnemonic	**Page**	**Size**	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOVE	220	B W L	0	0	SIZE		Dest Register			Dest Mode			Src Mode		Src Register			
MOVEA	223	W L	0	0	SIZE		An			0	0	1	Src Mode		Src Register			
MOVEM	232	W L	0	1	0	0	1	DR	0	0	1	Size	EA Mode		EA Register			Register List Mask
ADD	108	B W L	1	1	0	1	Dn Register			OPMODE			EA Mode		EA Register			
ADDA	111	W L	1	1	0	1	An Register			OPMODE			EA Mode		EA Register			
ADDI	113	B W L	0	0	0	0	0	1	1	0	SIZE		EA Mode		EA Register			
SUB	278	B W L	1	0	0	1	Dn Register			OPMODE			EA Mode		EA Register			
SUBI	283	B W L	0	0	0	0	0	1	0	0	SIZE		EA Mode		EA Register			
LEA	214	L	0	1	0	0	An Register			1	1	1	EA Mode		EA Register			
AND	119	B W L	1	1	0	0	Dn Register			OPMODE			EA Mode		EA Register			
OR	254	B W L	1	0	0	0	Dn Register			OPMODE			EA Mode		EA Register			
LSL	217	B W L	1	1	1	0	Count/Registe			1	S		i/r	0	1	Dn Register		
LSL	217	B W L	1	1	1	0	0	0	1	1	1	1	EA Mode		EA Register			
LSR	217	B W L	1	1	1	0	Count/Registe			0	S		i/r	0	1	Dn Register		
LSR	217	B W L	1	1	1	0	0	0	1	0	1	1	EA Mode		EA Register			
ASL	125	B W L	1	1	1	0	Count/Registe			1	S		i/r	0	0	Dn Register		
ASL	125	B W L	1	1	1	0	0	0	0	1	1	1	EA Mode		EA Register			
ASR	125	B W L	1	1	1	0	Count/Registe			0	S		i/r	0	0	Dn Register		
ASR	125	B W L	1	1	1	0	0	0	0	0	1	1	EA Mode		EA Register			
CMP	179	B W L	1	0	1	1	Dn Register			OPMODE(ea•Dn)			EA Mode		EA Register			
BCC	129	B W L	0	1	1	0	Condition			Displacement Bit								

Mode	Register List Mask															
Post-Increment	A7	A6	A5	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
Pre-decrement	D0	D1	D2	D3	D4	D5	D6	D7	A0	A1	A2	A3	A4	A5	A6	A7

Displacement Bit				
8-Bit Displace				
16-Bit Displace	$00			
32-Bit Displace	$FF			

Mnemonic	Page	Encoding	Condition
CC	90	0 1 1 0	Carry Clear
GT	90	1 1 1 0	Greater Than
LE	90	1 1 1 1	Less or Equal

	Operation	Size: Byte	Word	Long
	<ea> ♦ Dn -> Dn	0 0 0	0 0 1	0 1 0
	Dn ♦ <ea> -> <ea>	1 0 0	1 0 1	1 1 0

Direction	d	D
Right	R	0
Left	L	1

Operation Size	S
Byte	00
Word	01
Long	10

Direction of Transfer	DR	Operation Size	Size
Register to Memory	0	Word	0
Memory to Register	1	Long	1

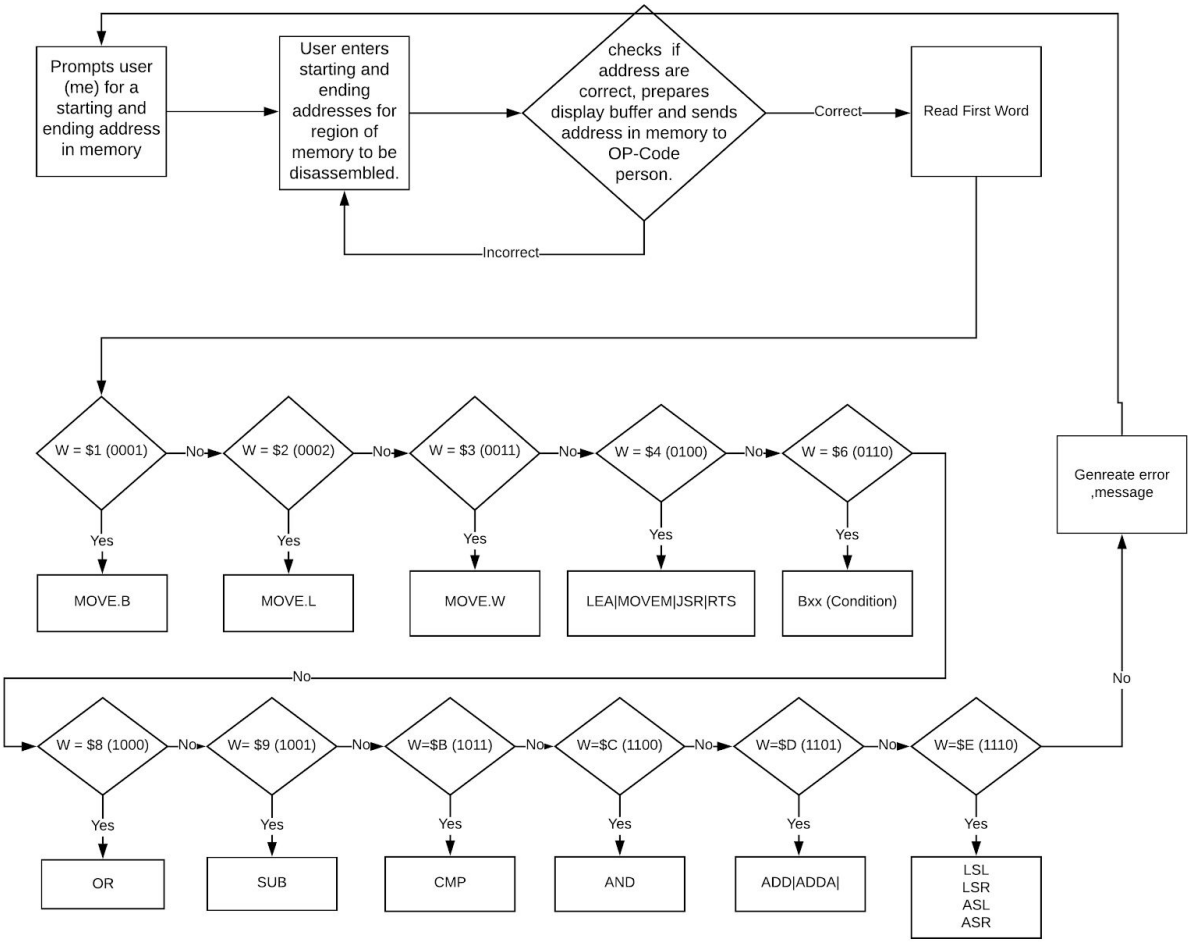
Operation Size	Suffix	SIZE
Byte B	.B	0 0
Word W	.W	1 1
Long L	.LW	1 0

Count/Register	i/r
Immediate	0
Register	1

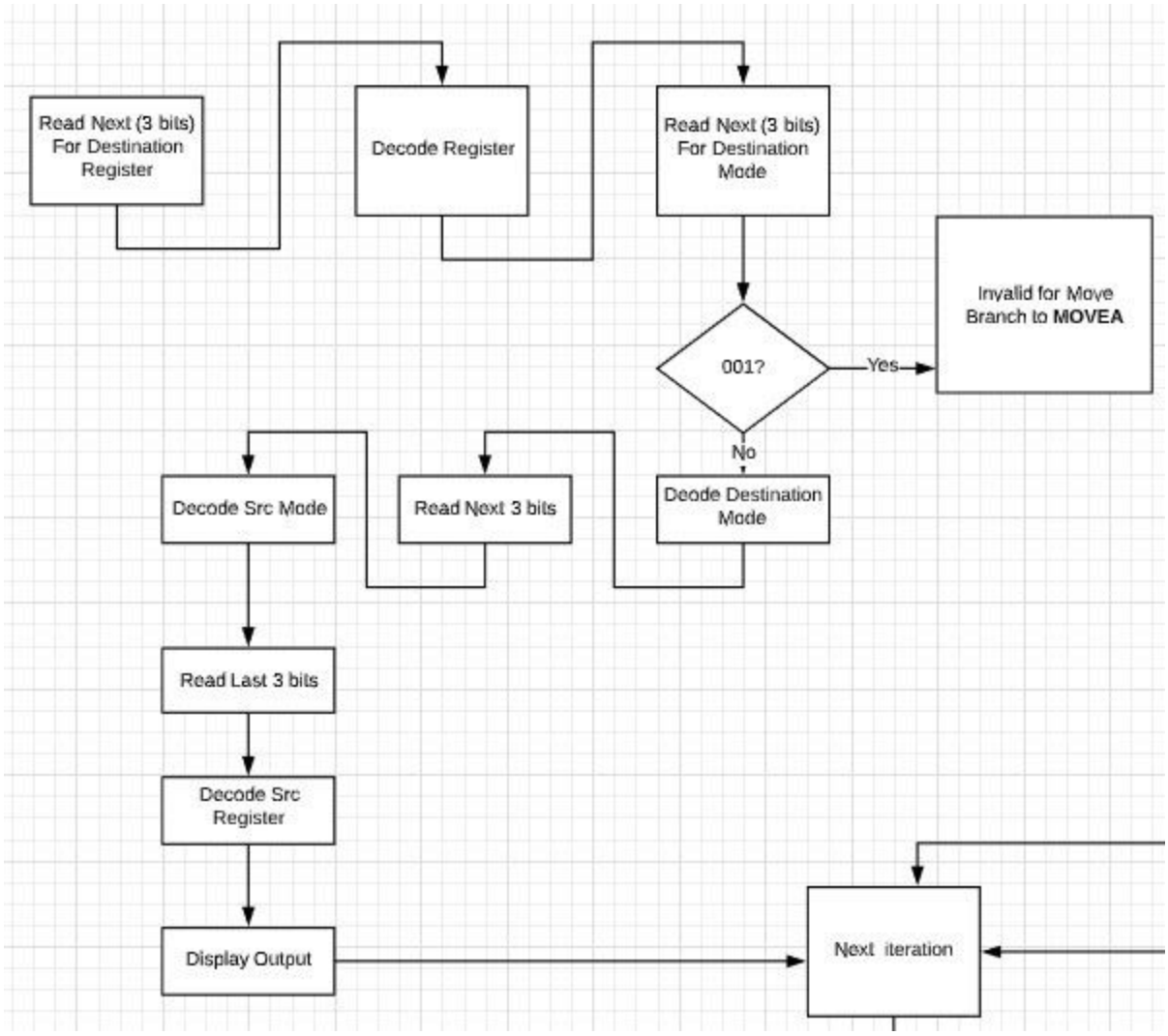
Figure 1.2: Effective Address

Addressing Mode		Format	Mode	Register
Direct Register	Data	Dn	0 0 0	reg
	Address	An	0 0 1	reg
Indrect Register	Address	(An)	0 1 0	reg
	Addr with Post Inc	(An)+	0 1 1	reg
	Addr with Pre Dec	-(An)	1 0 0	reg
Absolute Data Address	Short	(xxx).W	1 1 1	0 0 0
	Long	(xxx).L	1 1 1	0 0 1
Immediate		#<data>	1 1 1	1 0 0

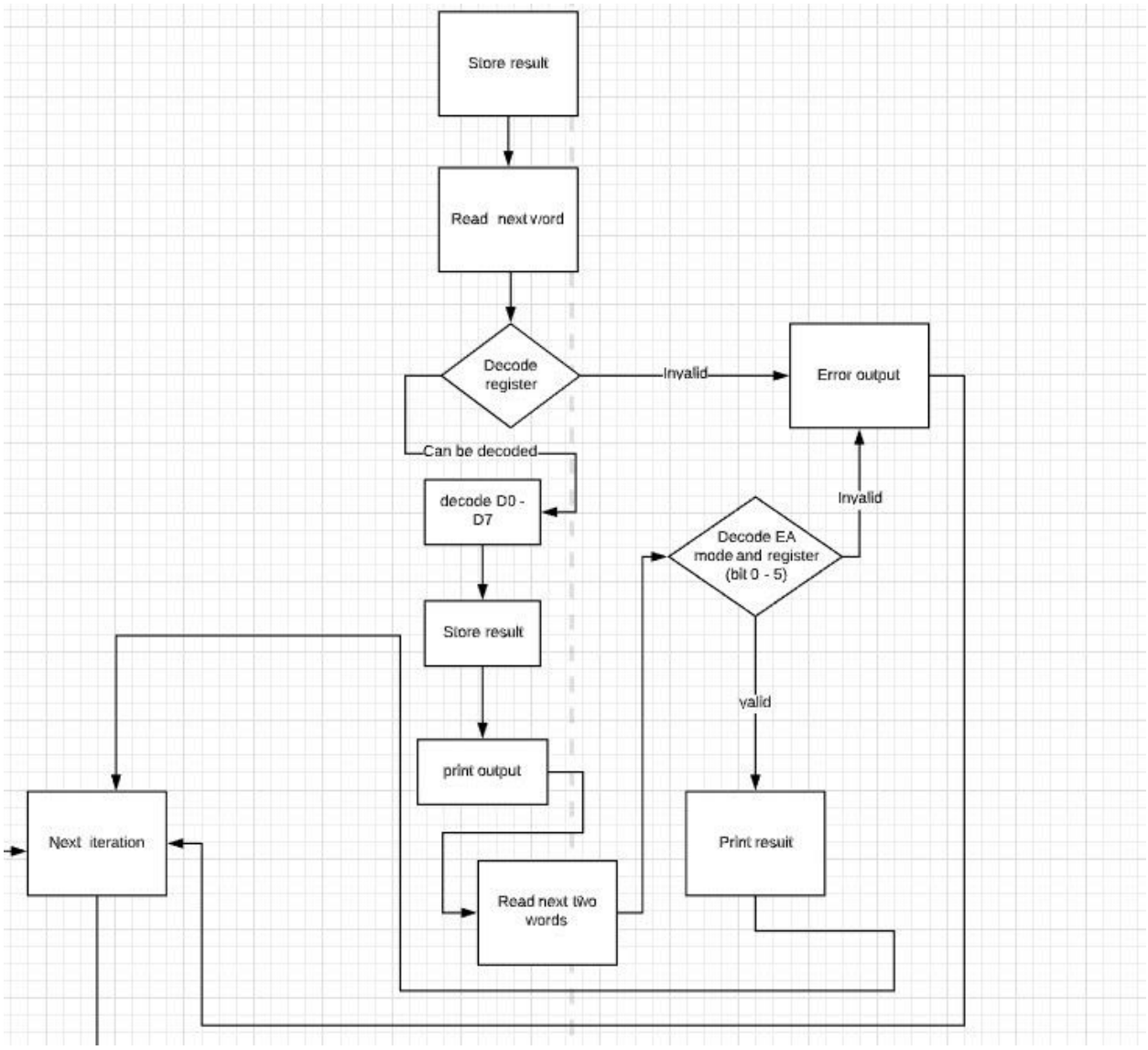
Figure 2.1: Flow Chart



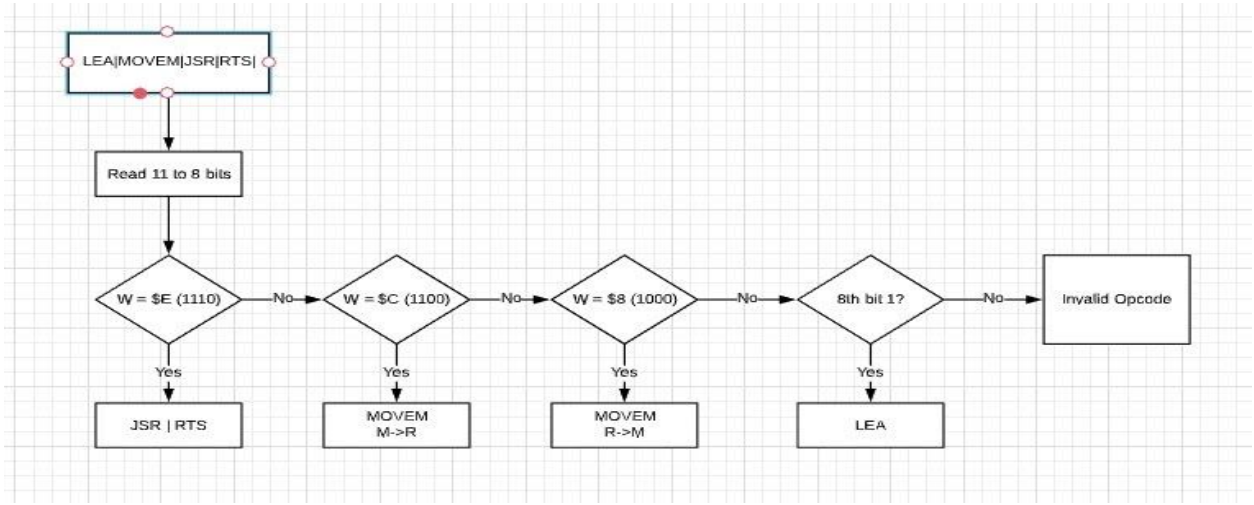
Flow chart for Move instruction



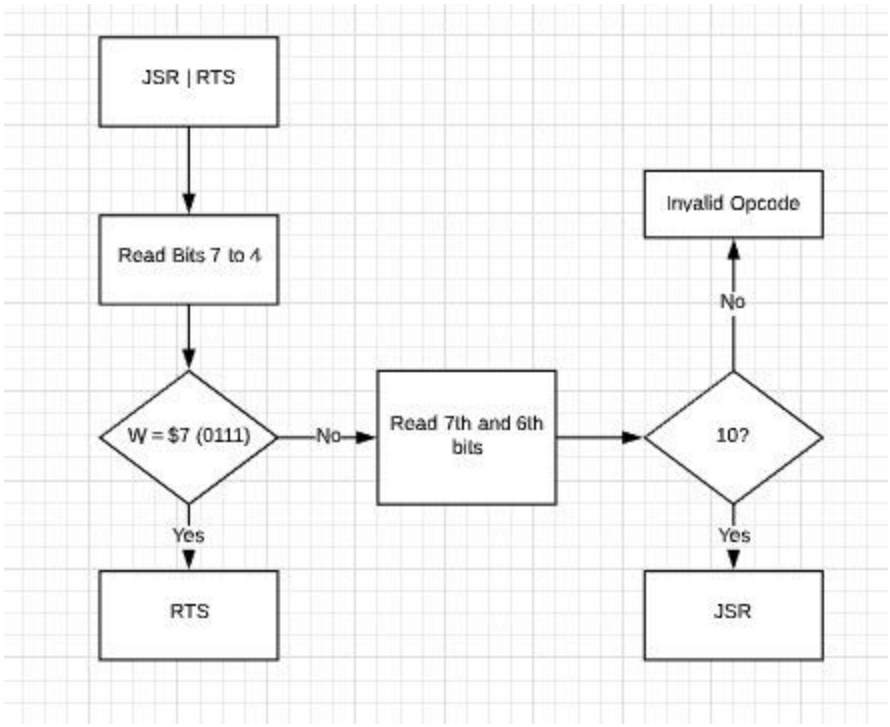
General Flow Chart



Flow chart for LEA, MOVEM, JSR|RTS (0100)



Flow chart for JSR | RTS



Problems:
Technical Part: Uncertainty of implementation. Many of these flow charts are created with initial guesses. It may have to re-structured.

Due to the weird situation (COVID 19), it is hard to communicate with team members.

Work Scheduled:
For the next iteration below things must be finished (Minimum).

1. Create a Repo.
2. Clear meeting time.
3. Input subroutine, which read user input as a string and converts to hexadecimal.
4. A flow chart for the subroutine.

Self Evaluation:
I think I am on track with the idea of the final project. But, lack of experience on 68k, I do not really know how to express my idea into assembly language.