Activity 2: Central Processing Unit (Part 3)

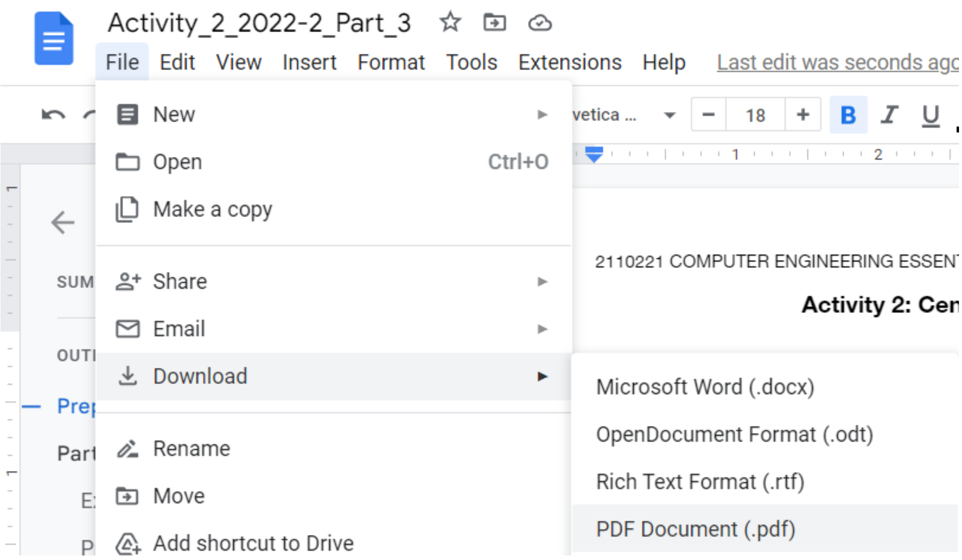
Group No : 3

Group Member :

1. Name ID Chandol Ngamcharoensathaporn
2. Name ID Chotpisit Adunsehawat
3. Name ID Chotiwit Fuengthanakul
4. Name ID Krittapas Rungsimantuchat

Preparation

- In part 1, use Activity 2 Reference: SML Instruction Set, which can be downloaded from myCourseVille or the link below: https://www.mycourseville.com/sites/all/modules/courseville/files/uploads/2016_1/2110221/materials/sml_instruction_set.333.1471674877.pdf
- In part 2 and 3, Use Brookshear Simple Machine Emulator to perform the indicated tasks https://www.mycourseville.com/sites/all/modules/courseville/files/uploads/2016_1/2110221/materials/bme.333.1471675276.htm
- Make a copy of this sheet. Answer the questions in the boxes given. After finishing, **save this file as a PDF** and **submit it to the assignment published on myCourseVille**.



Part 3 : Writing a Machine Language Program

Finally, after you've practiced on how to decode and understand basic machine instructions through examples in Part 1 and get familiar with how the emulator works in Part 2. Now it's time for you to write more complex machine language programs by yourself. xD

Using the Brookshear Emulator, each group will create 2 programs. To submit the results, students **must describe your solutions in a pseudo-code form** and **capture the screenshot of the memory from the emulator that contains the program after the program has finished execution.**

Once each question is complete, students must inform instructors or TAs for inspection.

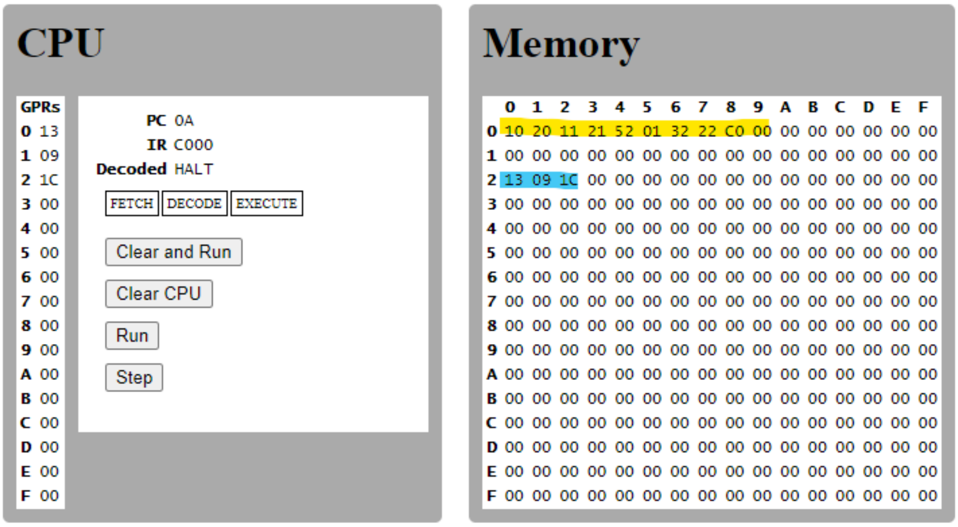
Example Program

Write a program in the machine language to ADD the contents of the memory cell at address 20 and 21. The result must be stored in the memory cell at address 22.

pseudo-code of the solution

```
1. Load the content of the memory cell address 20 into register 0.
2. Load the content of the memory cell address 21 into register 1.
3. Compute the sum of the values kept in register 0 and register 1, then place the result to
   register 2.
4. Store the value in register 2 into the memory cell address 22.
5. Halt execution.
```

screenshot of the memory from the emulator



GPRs	
0	13
1	09
2	1C
3	00
4	00
5	00
6	00
7	00
8	00
9	00
A	00
B	00
C	00
D	00
E	00
F	00

PC 0A
IR C000
Decoded HALT

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	10	20	11	21	52	01	32	22	C0	00	00	00	00	00	00	00
1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2	13	09	1C	00	00	00	00	00	00	00	00	00	00	00	00	00
3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
5	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
7	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Activity 2: Central Processing Unit (Part 3)

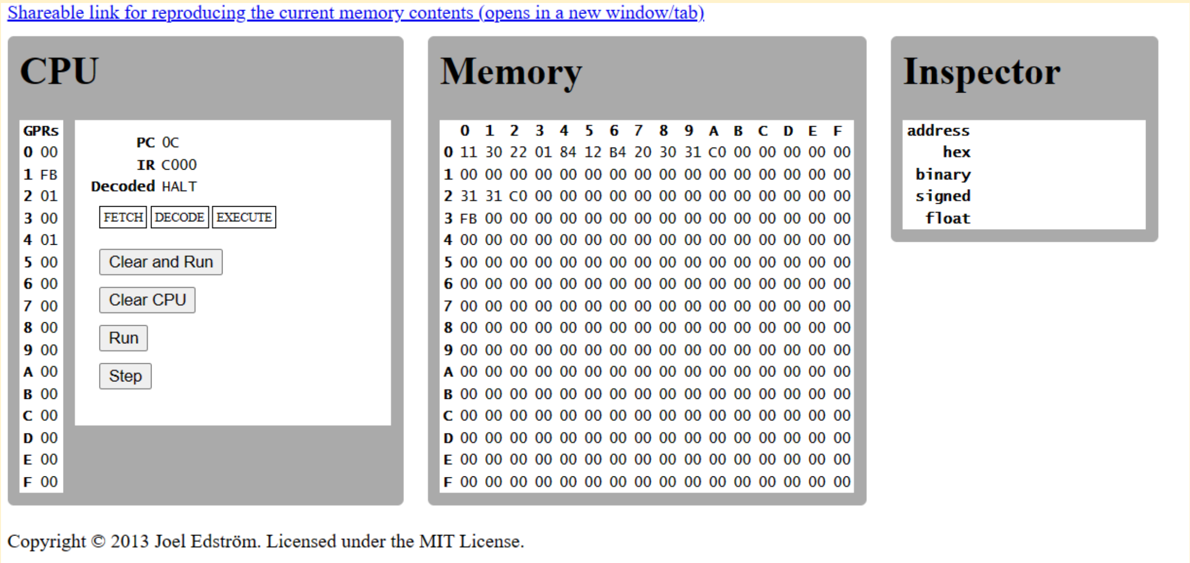
Program-A

Write a program in the machine language to check the contents of the memory cell at address 30. If the value is EVEN number, store the value of the memory cell at address 30 to the memory cell at address 31. Otherwise, store value 0 to the memory cell at address 31.

pseudo-code of the solution

1. LOAD the register 01 with the bit pattern found in the memory cell whose address is 30.
2. LOAD the register 02 with the bit pattern 01
3. AND the bit patterns in registers 01 and 02 and place the result in register 04.
4. Jump to Address 20 if Value in register 04 equals register 00 (00) to do case Even number. If equal, STORE the bit pattern found in register 01 in the memory cell whose address is 31 else STORE the bit pattern found in register 00 in the memory cell whose address is 31.

screenshot of the memory from the emulator



Shareable link for reproducing the current memory contents (opens in a new window/tab)

CPU

GPRs	
0	00
1	FB
2	01
3	00
4	01
5	00
6	00
7	00
8	00
9	00
A	00
B	00
C	00
D	00
E	00
F	00

PC 0C
IR C000
Decoded HALT

Memory

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	11	30	22	01	84	12	B4	20	30	31	C0	00	00	00	00	00
1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2	31	31	C0	00	00	00	00	00	00	00	00	00	00	00	00	00
3	FB	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
5	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
7	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Inspector

Program-B

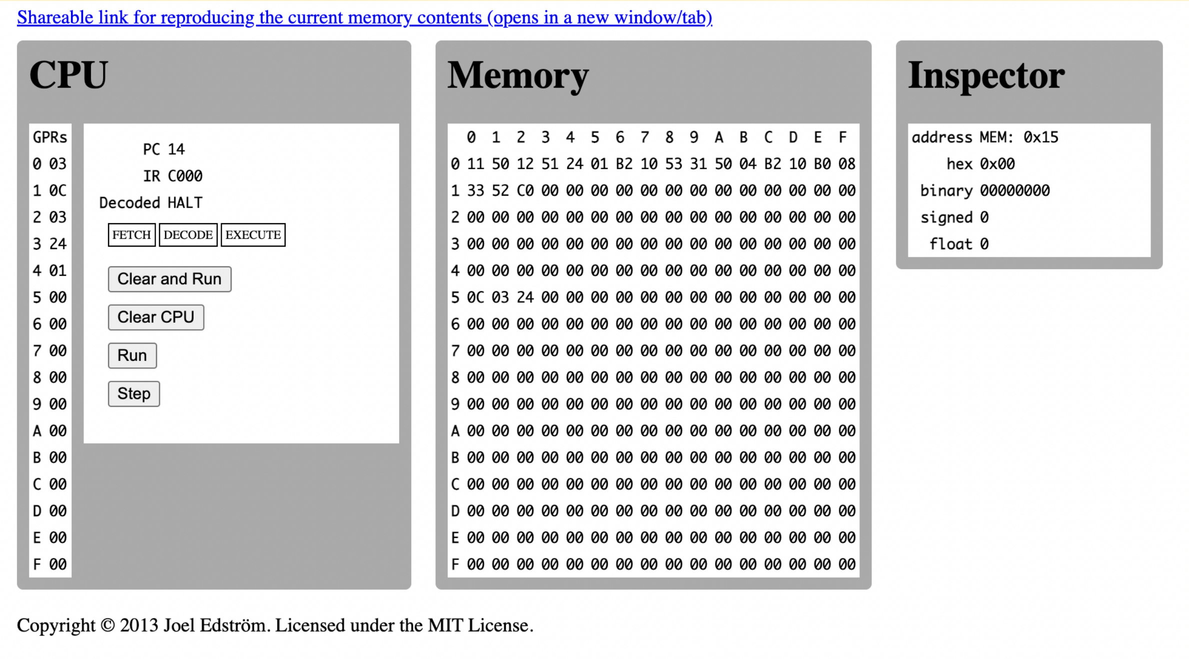
Write a program in the machine language to multiply the contents of the memory cell at address 50 to the contents of the memory cell at address 51 and store the result of multiplication to the memory cell at address 52. You can assume that both values are positive integers and the values are small enough such that the result of multiplication can be stored in 1 byte.

Activity 2: Central Processing Unit (Part 3)

pseudo-code of the solution

```
1.LOAD the register 01 with the bit pattern found in the memory cell whose address is 50.
2.LOAD the register 02 with the bit pattern found in the memory cell whose address is 51.
3.LOAD the register 04 with the bit pattern 01.
4.Check if multiplied by 00 or not.
5.ADD the bit patterns in registers 03 and 01 as though they were two’s complement representations and leave the result in register 03. (Result)
6.ADD the bit patterns in registers 00 and 04 as though they were two’s complement representations and leave the result in register 00. (Round)
7.Compare Register 00 with Register 02 ( Check that already add Every round)
8.If not, Make a loop by adding and comparing Compare and go to address 08.
9.STORE the bit pattern found in register 3 in the memory cell whose address is 52.
```

screenshot of the memory from the emulator



Shareable link for reproducing the current memory contents (opens in a new window/tab)

```
GPRs
0 03
1 0C
2 03
3 24
4 01
5 00
6 00
7 00
8 00
9 00
A 00
B 00
C 00
D 00
E 00
F 00
```

```
     PC 14
     IR C000
Decoded HALT
```

```
   0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0 11 50 12 51 24 01 B2 10 53 31 50 04 B2 10 B0 08
1 33 52 C0 00 00 00 00 00 00 00 00 00 00 00 00 00
2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5 0C 03 24 00 00 00 00 00 00 00 00 00 00 00 00 00
6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
address MEM: 0x15
    hex 0x00
 binary 00000000
 signed 0
  float 0
```

— THIS IS THE END OF PART 3 —

Activity 2: Central Processing Unit (Part 3)

A Chance to be "Outstanding"

Program-C and Program-D are used for "Outstanding Points". Each completed program is worth extra 0.5 points.

Once each question is complete, students must inform instructors or TAs for inspection.

Program-C

Write a program in the machine language to set the contents of the memory cell based on the contents of the memory cell at address F0, F1, and F2 as followed:

- The content of the memory cell at address F0 defines the value to be filled
- The content of the memory cell at address F1 defines the starting address to be filled
- The content of the memory cell at address F2 defines the last address to be filled

For example, if the contents of the memory cell at address F0, F1, and F2 are 0x01, 0xA2, and 0xA4, respectively. After execution, the program will fill in value 0x01 to memory cell at address A2, A3, and A4

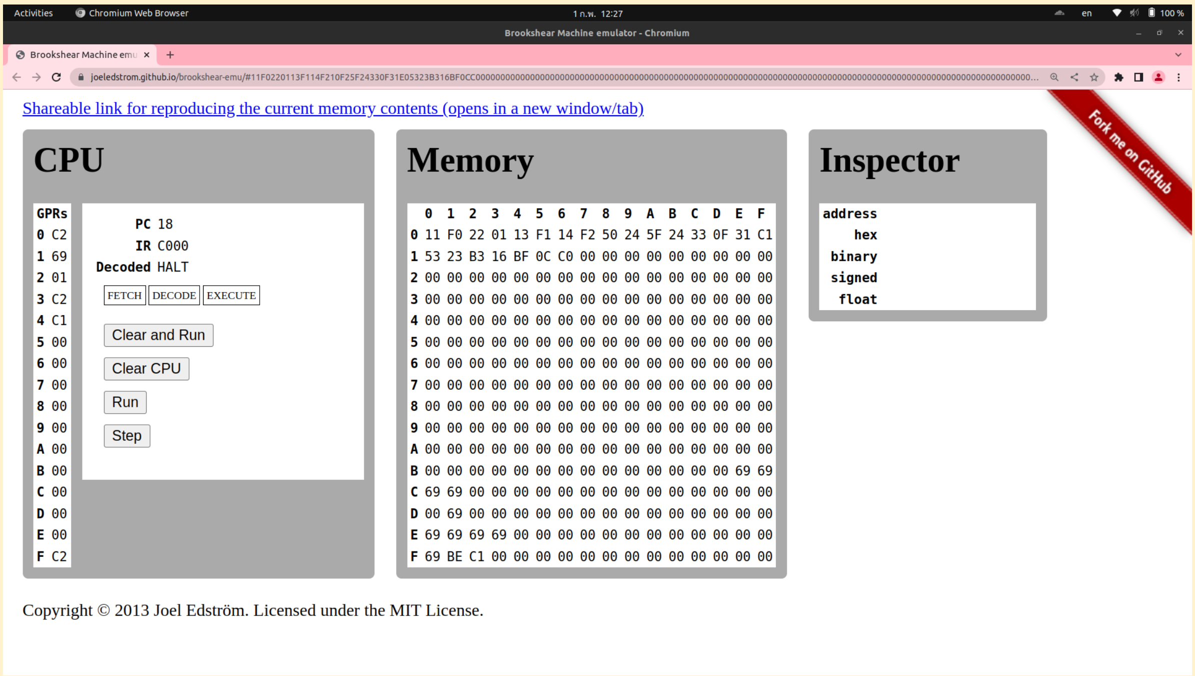
You can assume that the last address is always greater than or equal to the starting address.

pseudo-code of the solution

```
1. Load the register 1 with the bit pattern whose address is F0
2. Load the register 2 with the bit pattern 01
3. Load the register 3 with the bit pattern whose address is F1
4. Load the register 4 with the bit pattern whose address is F2
5. Add the bit pattern in register 2 and 4 and leave the result to register 0
6. Add the bit pattern in register 2 and 4 and leave the result to register F
7. Store th bit pattern found in register 3 in address 0F
8. Store the bit pattern found in register 1 in address <an address from step 7>
9. Add the bit pattern in register 2 and 3 and leave the result to register 3
10. Jump to 16 (Step 12) if the pattern is register 0 is equal to register 3's.
    Otherwise, continue
11. Jump to 0C (step 7) if the pattern in register 0 is equal to register F's (always
    true). Otherwise, continue.
12. Halt execution
```

screenshot of the memory from the emulator

Activity 2: Central Processing Unit (Part 3)


Activities
Chromium Web Browser
1 ก.พ. 12:27
en
100 %
Brookshear Machine emulator - Chromium
Brookshear Machine emu
joeledstrom.github.io/brookshear-emu/#11F0220113F114F210F25F24330F31E05323B316BF0CC0000000000000000000000000000000000000000000000000000000000000000000000000000000000...
Fork me on GitHub

Shareable link for reproducing the current memory contents (opens in a new window/tab)

CPU

```
GPRs
0 C2
1 69
2 01
3 C2
4 C1
5 00
6 00
7 00
8 00
9 00
A 00
B 00
C 00
D 00
E 00
F C2
```

```
      PC 18
      IR C000
 Decoded HALT
```

FETCH
DECODE
EXECUTE
Clear and Run
Clear CPU
Run
Step

Memory

```
  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0 11 F0 22 01 13 F1 14 F2 50 24 5F 24 33 0F 31 C1
1 53 23 B3 16 BF 0C C0 00 00 00 00 00 00 00 00 00
2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 69 69
C 69 69 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D 00 69 00 00 00 00 00 00 00 00 00 00 00 00 00 00
E 69 69 69 69 00 00 00 00 00 00 00 00 00 00 00 00
F 69 BE C1 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Inspector

```
address
    hex
 binary
 signed
  float
```

Activity 2: Central Processing Unit (Part 3)

Program-D

Write a program in the machine language to COMPARE the content of the memory cell at address 40 *(0x40) with the content of the memory cell at address 41 *(0x41) in decimal values. Assume that both values are represented in 8-bit Two's Complement.

- **If** *(0x40) < *(0x41), store value **"FF"** to the memory cell at address 42.
- **else if** *(0x40) > *(0x41), store value **"01"** to the memory cell at address 42.
- **otherwise**, store value "**00**" to the memory cell at address 42.

Example

Input		Output
*(0x40)	*(0x41)	*(0x42)
FD (-3)	05 (5)	FF
7F (127)	7E (126)	01
AF (-81)	AF (-81)	00

Hint You have already learned how to toggle the sign of the numbers represented in Two's Complement from the videos in the previous activity. Using EXCLUSIVE OR (XOR) operation may help you to solve this problem.

pseudo-code of the solution

screenshot of the memory from the emulator

— THIS IS THE END OF OUTSTANDING PART —