Activity 2-3 : Central Processing Unit

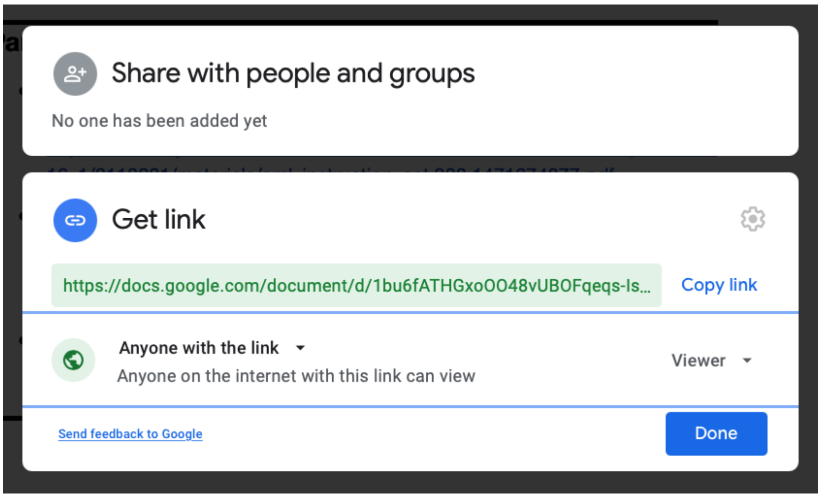
Group No : G27

Group Member :

1. Ittipat Yodprasit 6431349821
2. Anik Romyanon 6330571421
3. Waranthorn Chansawang 6432154921
4. Name ID

Part 0 : Preparation

- In part 1, use Activity 2 Reference: SML Instruction Set, which can be downloaded from CourseVille or 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 box given. Share this file with the permission for **anyone with link can view the document.** Submit the URL of this file to CourseVille.



Share with people and groups

No one has been added yet

Get link

https://docs.google.com/document/d/1bu6fATHGxoOO48vUBOFqeqs-ls...

Anyone with the link

Anyone on the internet with this link can view

Viewer

Part 3 : Writing A Machine Language Program (40 minutes)

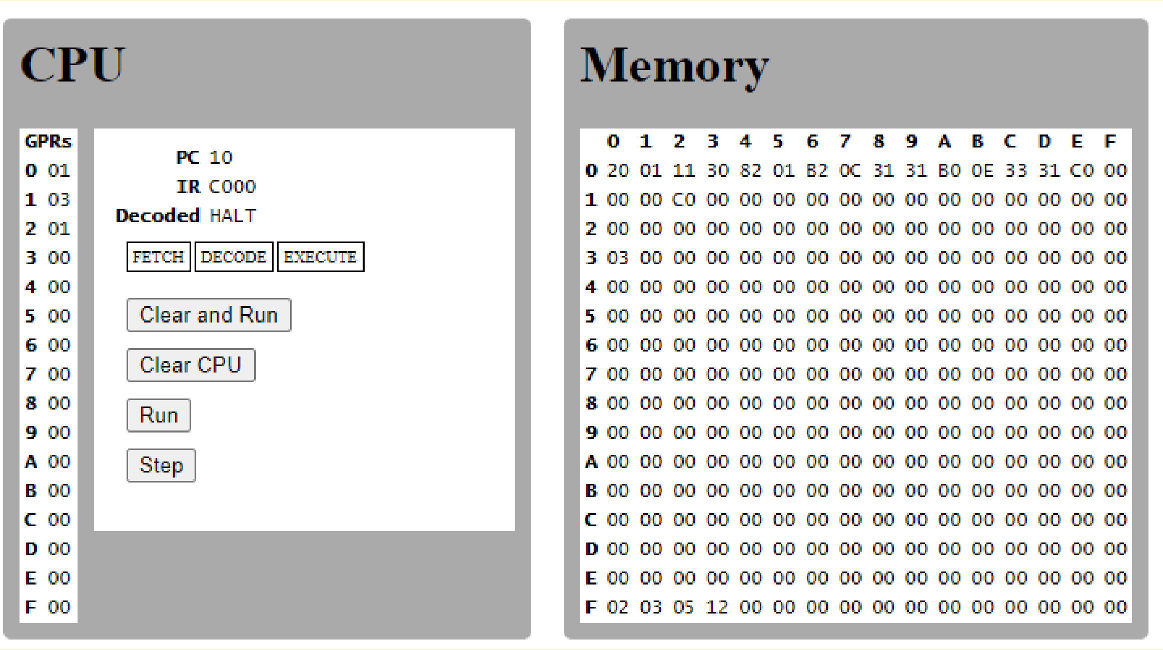
Using the Brookshear Emulator, each group will create 2 programs. To submit the results, students must describe their solutions in a pseudo-code and capture the screenshot of the memory from the emulator that contains the program

1. 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 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

```
Check if memory in 30 is even by AND with 01, if the result is 00,
result is even; put content of 30 to 31, if not put 00 in 31.
```

screenshot of memory in the emulator



CPU

GPRs	
0	01
1	03
2	01
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 10

IR C000

Decoded HALT

Memory

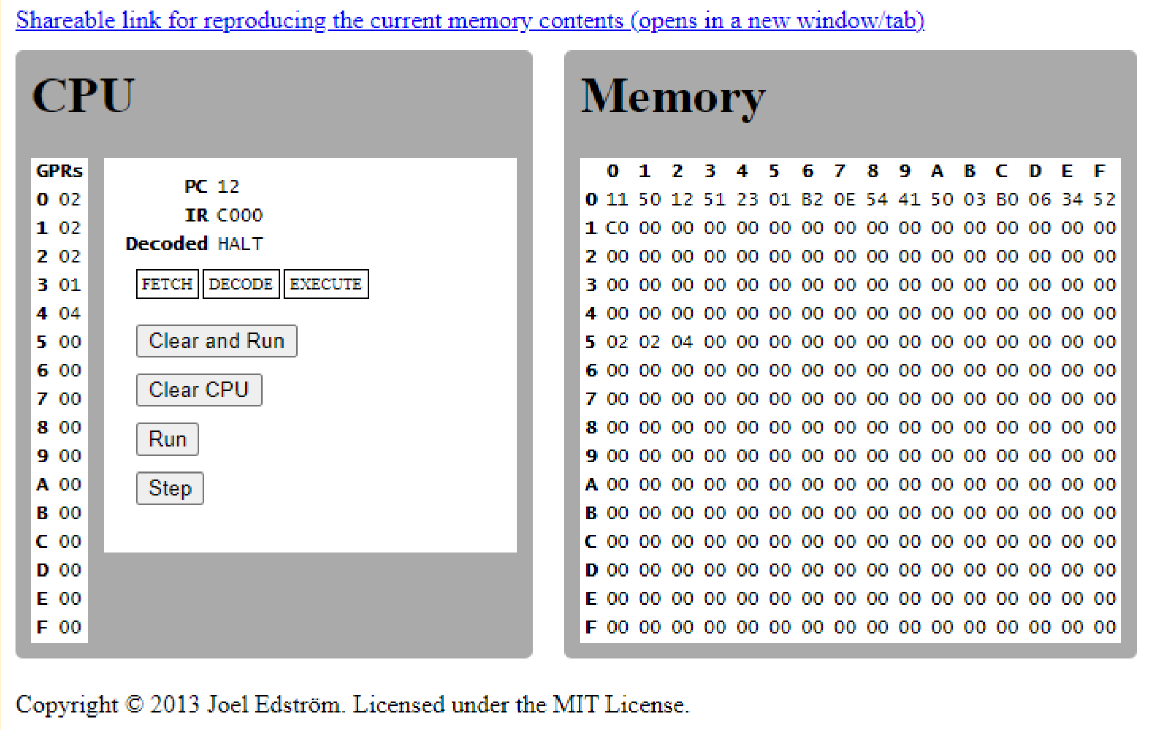
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	20	01	11	30	82	01	B2	0C	31	31	B0	0E	33	31	C0	00
1	00	00	C0	00	00	00	00	00	00	00	00	00	00	00	00	00
2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3	03	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
5	00	00	00	00	00	00	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	02	03	05	12	00	00	00	00	00	00	00	00	00	00	00	00

2. 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 stores the result of multiplication to the memory cell at address 52. You can assume that both values are positive integer and the values are small enough such that the result of multiplication can be stored in 1 byte.

pseudo-code of the solution

```
Store the content of 50 to register 1, and content of 51 to register 2, store 01 to register 3
If content of register 0 is equal to register 2: store content of register 4 to memory 52, then end the program.
If not, add value of register 4 by register 4 + register 1, add value of register 0 by register 3 + register 0, then jump back to check register 0 again.
```

screenshot of memory in the emulator



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

CPU

GPRs	
0	02
1	02
2	02
3	01
4	04
5	00
6	00
7	00
8	00
9	00
A	00
B	00
C	00
D	00
E	00
F	00

PC 12
IR C000
Decoded HALT

Memory

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	11	50	12	51	23	01	B2	0E	54	41	50	03	B0	06	34	52
1	C0	00	00	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	02	02	04	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

A Chance to be “Outstanding” (30 minutes)

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:

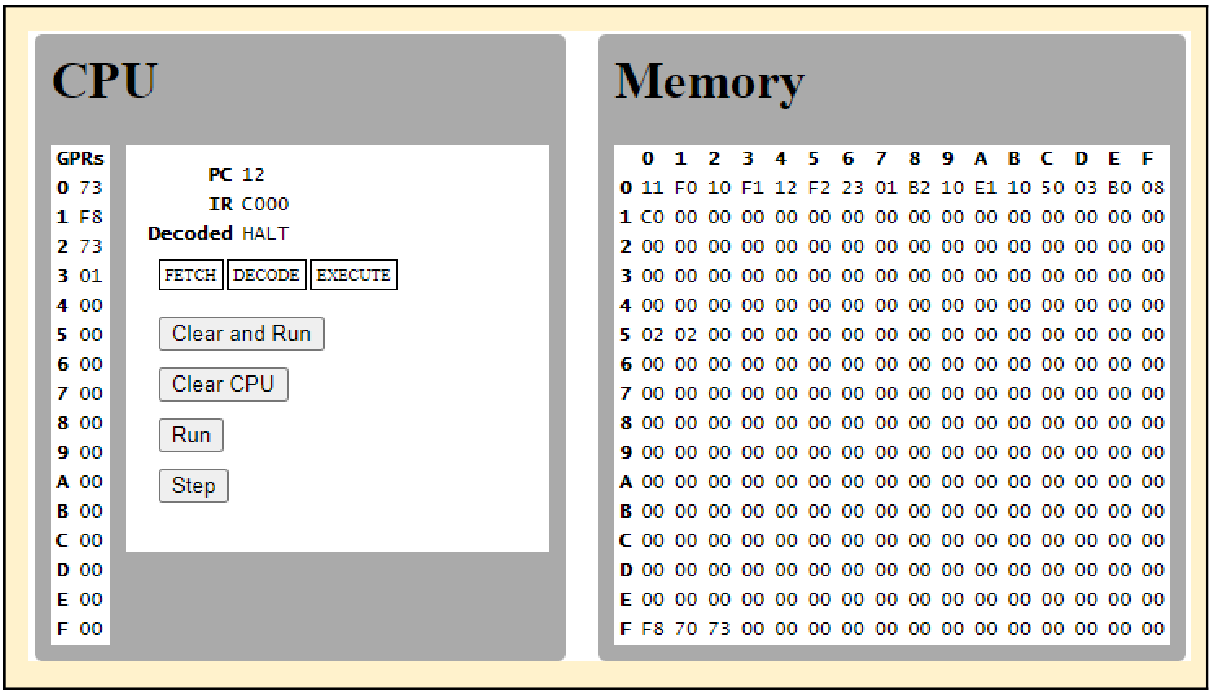
- The content of the memory cell at address F0 defines the value to be filled
- The content of the memory cell at address F1 defines the starting address to be filled
- The content of the memory cell at address F2 defines the last address to be filled

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

pseudo-code of the solution

```
Store f0,f1,f2 in registers then check if f1 == f2, if not store
f0 to address in register f1 then add f1 by 1, if f1 == f2, stop
the program.
```

screenshot of memory in the emulator



CPU

GPRs	
0	73
1	F8
2	73
3	01
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 12

IR C000

Decoded HALT

Memory

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	11	F0	10	F1	12	F2	23	01	B2	10	E1	10	50	03	B0	08
1	C0	00	00	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	02	02	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	F8	70	73	00	00	00	00	00	00	00	00	00	00	00	00	00