

ACTIVITY 2 PART3

SML Instruction Set ([with example](#))

Op-code	Operand	Description
1	RXY	LOAD the register R with the bit pattern found in the memory cell whose address is XY.
2	RXY	LOAD the register R with the bit pattern XY.
3	RXY	STORE the bit pattern found in register R in the memory cell whose address is XY.
4	ORS	MOVE the bit pattern found in register R to register S.
5	RST	ADD the bit patterns in registers S and T as though they were two's complement representations and leave the result in register R.
6	RST	ADD the bit patterns in registers S and T as though they represented values in floating-point notation and leave the floating point result in register R.
7	RST	OR the bit patterns in registers S and T and place the result in register R.
8	RST	AND the bit patterns in registers S and T and place the result in register R.
9	RST	EXCLUSIVE OR the bit patterns in registers S and T and place the result in register R.
A	ROX	ROTATE the bit pattern in register R one bit to the right X times. Each time place the bit that started at the low-order end at the high-order end.
B	RXY	JUMP to the instruction located in the memory cell at address XY if the bit pattern in register R is equal to the bit pattern in register number 0. Otherwise, continue with the normal sequence of execution. (The jump is implemented by copying XY into the program counter during the execute phase.)
C	000	HALT execution.

Part 1 and Part 2 will be done and graded on MyCourseVille

Part 1 answer

1. LOAD the register 2 with the bit pattern 36
2. AND the bit patterns in registers 5 and 2 and place the result in register 3
3. C000
4. LOAD the register 5 with the bit pattern 0F.
5. 00
6. Nothing (the program will execute the instruction after B12A)
7. 31
8. 11

Part 2 answer

1. 02, 10F0, 02
2. 04, 11F1, 03
3. 06, 12F2, 05
4. 08, 2301, 01
5. 0A, 5403, 03
6. 0E
7. 16
8. $\text{Result} = [(*0xF0) + (*0xF1) + (*0xF2) + 1] * 2$

Part 3 and Outstanding:

There are 17 problems in this section (difficulty code: 1-easy, 2-medium, 3-hard)

Part 3 score : easy and medium difficulty

Outstanding score : comes from hard difficulty

All TEST CASE and it's output are in [HERE \(YAML format\)](#)

EASY: compute values

ส่วนนี้จริงๆอ่าน Instructions set เป็นก็น่าจะทำได้ทุกคน แต่มีบางข้อที่น้องอาจจะถาม

Description: swap the left hexadecimal value with the right hexadecimal value of the content of the memory cell at address 0x20.

ข้อนี้น้องต้องใช้ ROTATE (AR04) โดยจะ Rotate Register ที่ R 4 ครั้ง

Description: flip the binary bits (swap 1s and 0s in the same significance) of the memory cell contents at address 0x20.

ข้อนี้น้องต้องเอา value ใน @0x20 มา XOR กับ 1111 1111

MEDIUM: compare contents

ส่วนนี้จะเล่นกับ instruction *JUMP* เป็นหลัก
(เทียบค่า ใน Register *x* กับ Register 0 ถ้าเท่ากัน Jump ไป address นั้นๆ)
ข้อที่น้องน่าจะถาม

*Description: check If the value is **EVEN** number.*

-ข้อนี้ เอาค่าของน้อง มา AND กับ 0000 0001 (เหมือนเราสนใจจะเทียบแค่หลักแรก) แล้วเอาค่าหลัง AND มาเทียบกับ 0000 0000 ถ้าเท่า Even, ไม่เท่า Odd

*Description: check If the value is a **positive or zero** number.*

-ข้อนี้ เอาค่าของน้อง มา AND กับ 1000 0000 (สนใจจะเทียบแค่หลัก sign bit) แล้วเอาค่าหลัง AND มาเทียบกับ 0000 0000 ถ้าเท่า Zero or Pos num, ไม่เท่า Negative num

*Description: check if @0x30 is **exactly double the value** of the @0x31.*

-ข้อนี้เอา @0x30 มาคูณ 2 (บวกด้วยตัวเอง) แล้ว check equal(jump) กับ @0x31

*Description: check if the **first and last bits** of two memory cells **are equal***

-ข้อนี้ เอาค่าของน้อง มา AND กับ 1000 0001 ทั้ง2ตัว(สนใจแค่หลักแรกและสุดท้าย) แล้วเอาค่าหลัง AND มาเทียบกันเอง

*Description: check if the **right hexadecimal value** of 2 memory cells **are equal**.*

-ข้อนี้ เอาค่าของน้อง มา AND กับ 0000 1111 ทั้ง2ตัว(สนใจแค่ 4 bit ขวา) แล้วเอาค่าหลัง AND มาเทียบกันเอง

HARD: loop computation (OUTSTANDING)

เห็น JUMP จนกว่าจะผ่าน condition อันนี้มีโค้ดตัวอย่างให้TAดู เพราะยากอยู่

Description: multiply the contents of the memory cell at address 0x50 to the contents of the memory cell at address 0x51.

IDEA: สร้าง counter บวกตัวตั้ง ซ้ำๆ จนกว่าจะมีค่าเท่ากับตัวคูณ

FOR GRADER : [LINK](#)

Description: count bit with value 1 in the content of memory cell

IDEA: สร้าง counter กับ Mask 0000 0001(AND เพื่อดูแค่ bit แรก) แล้ว rotate check หา 1 จนครบ 8 bits

FOR GRADER : [LINK](#)

THIS LINK IS EASY TO READ : [SOLUTION LINK](#)

Description: rotate the bit of the binary value stored in the memory cell at address 0x50 until the first bit from the right (least significant bit) is '1' in binary.

IDEA: สร้าง mask 0000 0001(AND เพื่อดูแค่ bit แรก) แล้ว rotate จนกว่า จะเจอ 1 ใน mask

FOR GRADER : [LINK](#)

THIS LINK IS EASY TO READ : [SOLUTION LINK](#)

Description: calculate the summation of $X + i$ for values of i from 0 to n (sum of $X+0, X+1, X+2, \dots, X+n$), where the initial value X is stored in the memory cell at address 0x50, and the value of (n) is stored in the memory cell at address 0x51.

IDEA: สร้าง counter บวกตัวตั้งทีละหนึ่งและหาผลรวม ซ้ำๆ จนครบ n รอบ

FOR GRADER : [LINK](#)

THIS LINK IS EASY TO READ : [SOLUTION LINK](#)

Description: divide the value stored in the memory cell at address 0x50 by the value in the memory cell at address 0x51.

IDEA: สร้าง counter เก็บผลลัพธ์ แล้วบวกตัวหาร ซ้ำๆ จนกว่าจะมีค่าเท่ากับตัวตั้ง แล้ว spit ค่าใน counter ออกมา

FOR GRADER : [LINK](#)

THIS LINK IS EASY TO READ : [SOLUTION LINK](#)