Suppose R# is an 8-bit register. The data stored in R# is treated as an unsigned binary number. Explain what "arithmetic" operation the following instruction performs.

INV R2, R1

(Notice that "INV" performs bitwise inversion for input R1 and stores the result to R2.)

$$R_2 = \overline{R_1}$$

The following shows all the bits of R_2.

$$R_1 = x_7 x_6 \dots x_0$$

$$R_2 = \overline{x_7 x_6} \dots \overline{x_0}$$

If you add the two, you get

$$R_1 + R_2 = 11 \dots 1 = 255$$

Thus, the "arithmetic" operation it performs is

$$R_2 = 255 - R_1$$

Suppose R# is an 8-bit register. The data stored in R# is treated as a two's complement binary number. Explain what "arithmetic" operation the following instruction performs for input R1.

AND R2, R1, 0x7F

If R1 is zero or positive, R2 = R1.
If R1 is negative,

$$R_1 = 1x_6x_5 \dots x_0$$
$$R_2 = 0x_6x_5 \dots x_0$$
$$|R_1| = 0\overline{x_6x_5} \dots \overline{x_0} + 1$$
$$R_2 + |R_1| = 01111111_2 + 1 = 128$$
$$R_2 = 128 - |R_1|$$

It performs $128 - |R_1|$ (for a negative number) just by one logical instruction.

Suppose R# is an 8-bit register. The data stored in R# is treated as an unsigned binary number. Draw a graph for the following instruction. The x-axis should be the value stored in R1 and the y-axis should be the value stored in R2.

EOR R2, R1, 0x01

(Notice that "EOR" performs bitwise exclusive-OR between input R1 and 0x01 and stores the result to R2.)

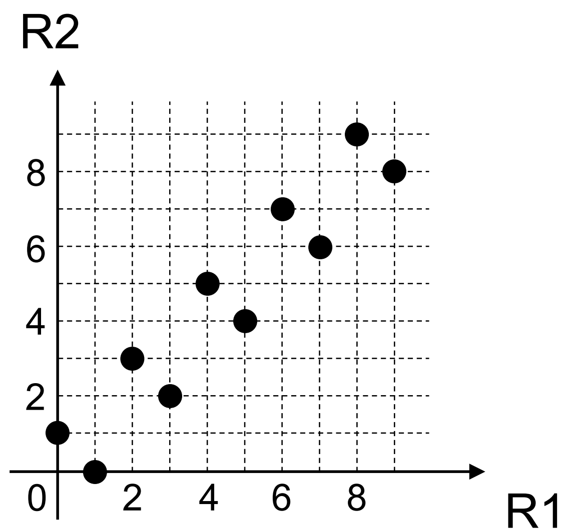
$$R_1 = x_7 x_6 \dots x_0$$
$$R_2 = x_7 x_6 \dots \overline{x_0}$$

If R_1 is even ($x_0 = 0$), $R_2 = R_1 + 1$ ($y = x + 1$).
If R_1 is odd ($x_0 = 1$), $R_2 = R_1 - 1$ ($y = x - 1$).

Well, you might express it as

$$R_2 = R_1 + (-1)^{R_1}$$

but the goal is to draw the graph, so you don't need to express it like that.



Suppose R# is an 8-bit register. The data stored in R# is treated as an unsigned binary number. Draw a graph for the following instruction. The x-axis should be the value stored in R1 and the y-axis should be the value stored in R2.

EOR R2, R1, 0x02

(Notice that "EOR" performs bitwise exclusive-OR between input R1 and 0x01 and stores the result to R2.)

$$R_1 = x_7 x_6 \dots x_1 x_0$$
$$R_2 = x_7 x_6 \dots \overline{x_1} x_0$$

If x_1 is 0 (then, R_1 is $4k$ or $4k+1$), $R_2 = R_1 + 2$ ($y = x + 2$).
If x_1 is 1 (then, R_1 is $4k+2$ or $4k+3$), $R_2 = R_1 - 2$ ($y = x - 2$).

Well, you might express it as

$$R_2 = R_1 + (-1)^{(R_1\ DIV\ 2)} \cdot 2$$

where (DIV calculates the quotient of the integer division), but the goal is to draw the graph, so you don't need to express it like that.