

### 4.3

The name program counter is misleading because the program counter doesn't count anything at all.

The value in the program counter is the address of the next instruction to be processed, which means its content "point to" the next instruction. Hence we say the name instruction pointer is more insightful.

### 4.5

*a.* Location 3 contains 0000 0000 0000 0000. Location 6 contains 1111 1110 1101 0011.

*b.*

(1) As 2's complementary integers, the value of location 0 is 7747 and the value of location 1 is -4059.

(2) As an ASCII value, the value of location 4 is 101, which represents "e".

(3) The complete floating point number is 0000 0110 1101 1001 1111 1110 1101 0011 and the value is  $1.1011001111111011010011 \times 2^{-114}$ .

(4) As unsigned integers, the value of location 0 is 7747 and the value of location 1 is 61477.

*c.* 0001 1110 0100 0011 represents "ADD R7<-R1+R3", which means

to add the value of R1 to the value of R3 and store the result in R7.

*d.* It refers to location 6. Location 6 contains 1111 1110 1101 0011.

4.8

*a.* 8 bits.

*b.* 7 bits.

*c.*  $32 - 8 - 7 \times 3 = 3$ . The maximum number of unused bits is 3.

4.9

It's loading the address of the next instruction into the PC.

5.2

The MDR is 64-bit, but it tells nothing about the size of the MAR.

5.7

15, which is also  $01111_{(2)}$ .