Here's a detailed explanation of each line:

DATA SEGMENT NO DW 5648H Z DW ? O DW ? DATA ENDS

This section defines the data segment of the program. It declares three 16-bit variables: NO, Z, and O. NO is initialized to 5648H (which is the decimal value 22056), while Z and O are left uninitialized.

CODE SEGMENT ASSUME CS:CODE, DS:DATA START:

This section defines the code segment of the program. It sets up the segment registers and defines the START label as the entry point of the program.

MOV AX, DATA MOV DS, AX

These instructions set up the data segment register (DS) with the address of the data segment.

MOV AX, NO MOV BX, 00H MOV CX, 10H MOV DX, 00H

These instructions load the value of NO into the AX register, and set up the BX, CX, and DX registers for the main loop.

UP: ROL AX,1 JC ONE INC BX JMP NXT

This is the main loop of the program. It uses the ROL (rotate left) instruction to shift the bits of AX to the left, one bit at a time. The JC (jump on carry) instruction checks whether the most significant bit of AX is set. If it is set, the program jumps to the ONE label. Otherwise, it increments the BX register and jumps to the NXT label.

ONE: INC DX

This label is reached when a 1 bit is encountered. It increments the DX register to count the number of 1 bits.

NXT: DEC CX JNZ UP

This label is reached when all 16 bits have been processed. It decrements the CX register and checks whether it has reached zero. If CX is not zero, the program jumps back to the UP label to process the next bit. If CX is zero, the loop exits.

MOV Z, BX MOV O, DX

These instructions store the final counts of zeros and ones in the Z and O variables.

INT 3 CODE ENDS END START

This instruction generates an interrupt to halt the program.