I complied the assembler instruction into machine languages below one by one, depending on the translation tables listed in textbooks and video lectures. My process is more human oriented since I do not have embedded program to compile them. So, I started with sperate two instructions into a-instruction and c-instruction. Then, give binary numbers to all a-instructions for my ease. After that, I completed c-instructions’ comp part, dest part, and jump part one by one. As I am human being, it was hard for me to keep track of which bits are equal to which parts. Throughout this process, I was able to learn how simple what computer is doing is in the lowest layer of hardware. Based on these computations, we can watch YouTube, use AI to predict the future stock price and so on. It is just amazing.

The result of CPU emulation is attached below as well. As for the second the second program, it repeats the jump in the program, so the image is just the snapshot of one execution. Actually, the test with CPU emulator helped me to find one wrong binary, since those bits were not able to be stored in the emulator.

//Program 1 to convert to machine language

// Computes R0 = 2 + 3

@2

D=A

@3

D=D+A

@0

M=D

0000000000000010

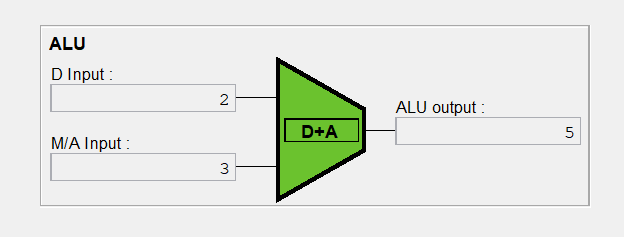
1110110000010000

0000000000000011

1110000010010000

0000000000000000

1110001100001000



// Program 2 to convert to machine language

// Symbol-less version of the Max.asm program.

@0

D=M

@1

D=D-M

@10

D;JGT

@1

D=M

@12

0;JMP

@0

D=M

@2

M=D

@14

0;JMP

0000000000000000

1111110000010000

0000000000000001

1111010011010000

0000000000001010

1110001100000001

0000000000000001

1111110000010000

0000000000001100

1110101010000111

0000000000000000

1111110000010000

0000000000000010

1110001100001000

0000000000001110

1110101010000111

