# WEEK 3 SCRIPT -

-- PIC 1 --

Welcome to the final week of the computer design and architecture course, in this week, we’re going to build the CPU and an assembler - which is a program which can take the assembly program and convert into binary instructions that a computer can understand.

Unlike the previous 2 weeks, this week is divided into just 3 parts, 2 which we’ll implement together and the last 1 is going to be the final project of this course.

-- PIC 2 --

This is arguably going to be the shortest video among the weekly course videos. Why? Because quite simply, it’s the hardest. The CPU is in simple terms, the combination of the ALU, RAM unit and the ROM unit. It’s the government of the computer, it decides what gets done when according to the input given by the user. Just like for the ALU, while building the CPU, you’ll be given a schematic like the one behind me and you’re going to have to implement it. And that’s it part 1 is done!

-- PIC 3 --

By now you’ve actually built a fully functioning computer, and you can run programs written in assembly language. But, if you would’ve observed, there’s a missing link, I haven’t taught you how to convert the assembly language instructions into the machine language (or 1s and 0s) that the computer can understand.

Which brings us to part 2, where I’m going to teach the algorithm on how to convert assembly code into simple 0s and 1s. Using this algorithm, you’ll be able to convert any program written in the hack assembly language into binary code, if you know programming, you can write a program to implement the algorithm in your prefered language. if you don’t know programming, you can simply do it by hand on a piece of paper!

-- PIC 4 --

For the final project we’re going to test your CPU for various different cases using the hardware simulator and if it passes, so do you.

-- PIC 5 --

Congratulations on completing the course! You now have a virtual computer at your disposal, I’d suggest you not to stop right here though and go ahead and make more programs and test the computer it’s limits. It might not work, but you never learn things, if you don’t break things.