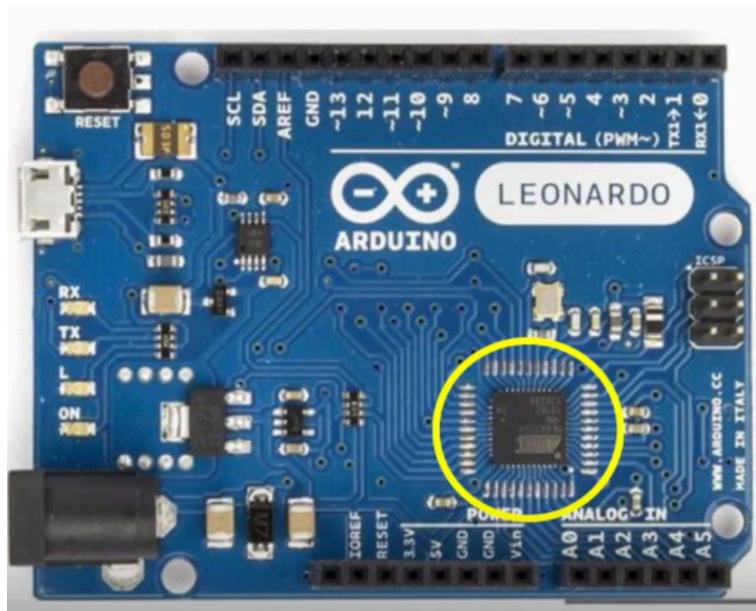


# DAY 1 NASSDOM \_VSD-IAT

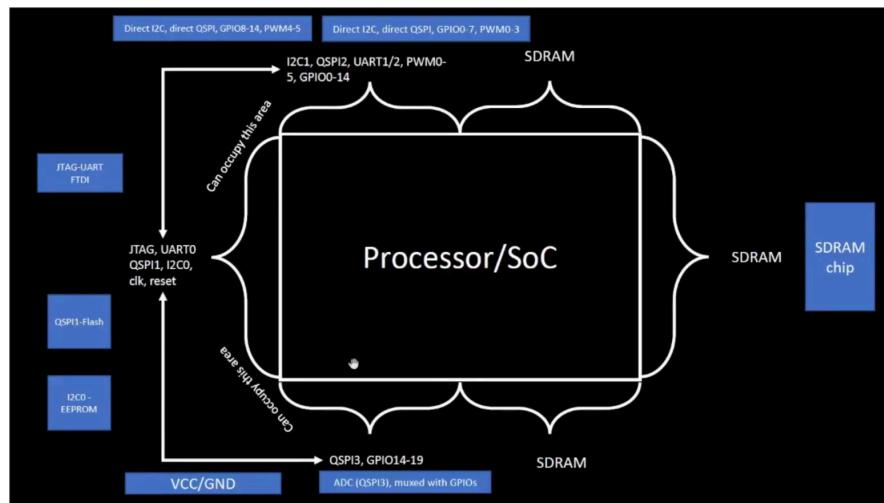
## SKY130\_D1\_SK1: How to talk to Computers

SKY\_L1 - Introduction to QFN-48 Package, chip, pads, core, die and IPs



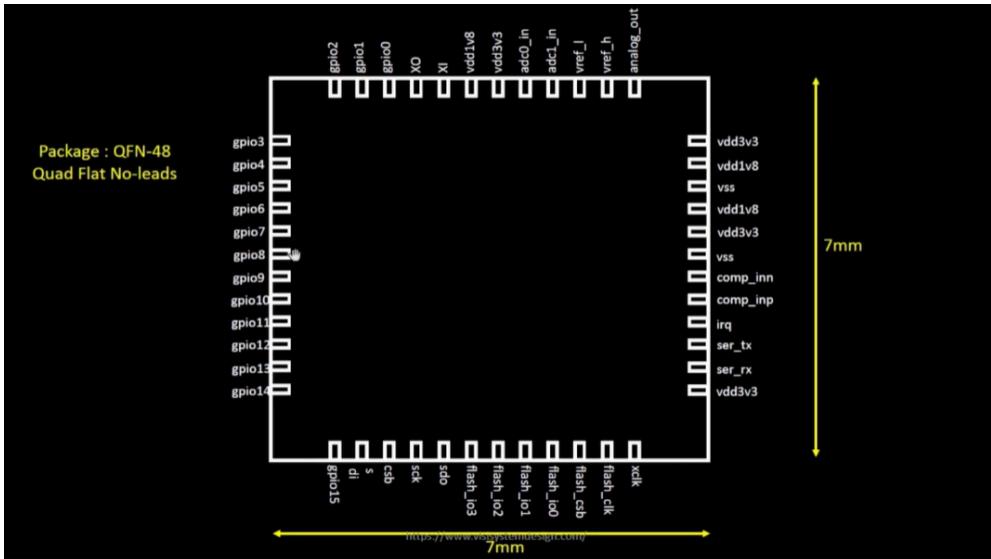
[ARDUINO UNO BOARD]

The highlighted chip is what we are going to study for now. The diagram below is the block diagram of the board above, with the centre chip or processor highlighted.



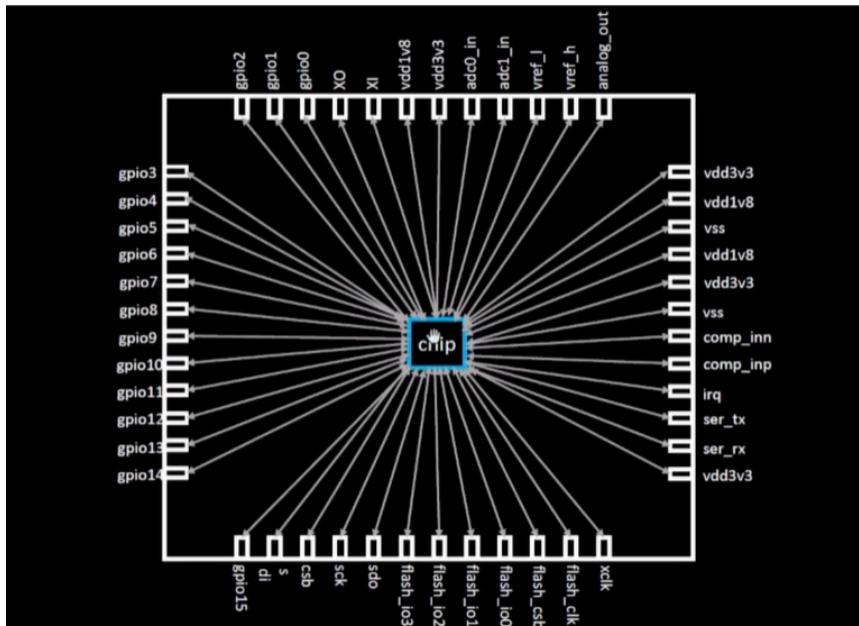
[BLOCK DIAGRAM OF BOARD]

Now if we try to open and look inside that IC, the below figure is what it will look like. From now on we will call this a package. There are different types of packages and all look different. The pin location of this package is driven by the arduino board it is embedded on.



[PACKAGE]

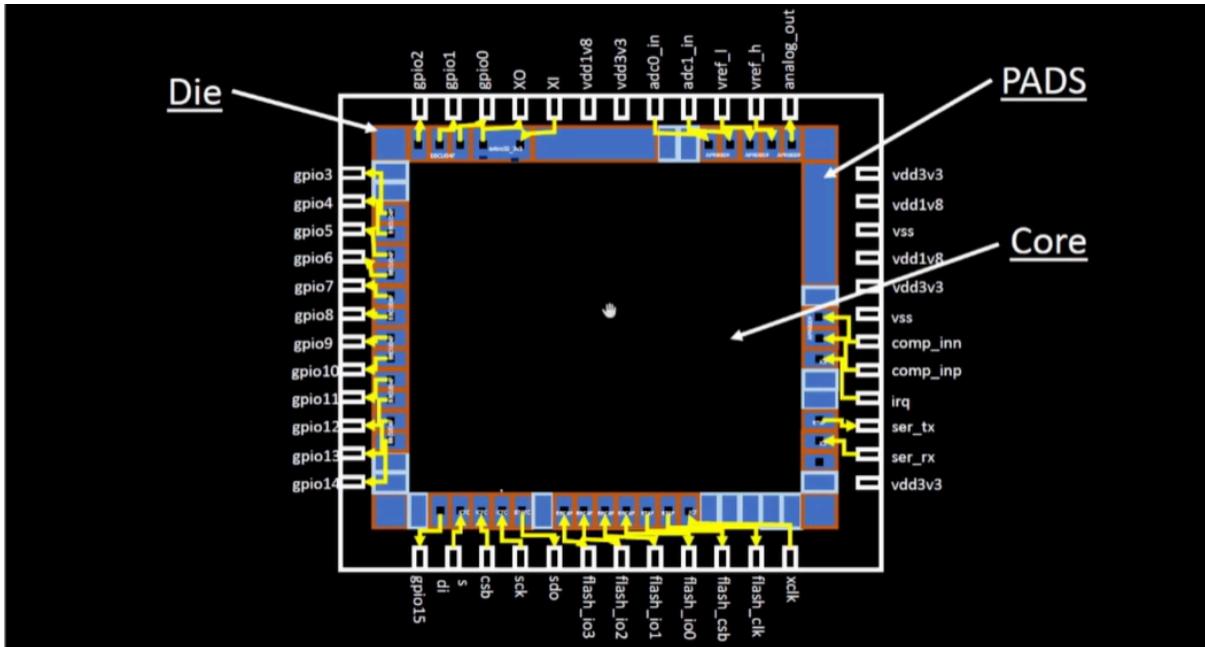
The chip is sitting at the centre of the package and it is connected to the pins using bond wires. The signals from the outside world are sent to the chip using these wires.



[CONNECTION BETWEEN CHIP AND PINS]

Now if you look into the chip and study its important components one of which is pads, which are something through which we send the signal inside the chip and vice versa.

Then comes the core of the chip where our digital logic sits like our gates like OR, AND NOR. And die is the size of the entire chip, which is getting manufactured on silicon paper.

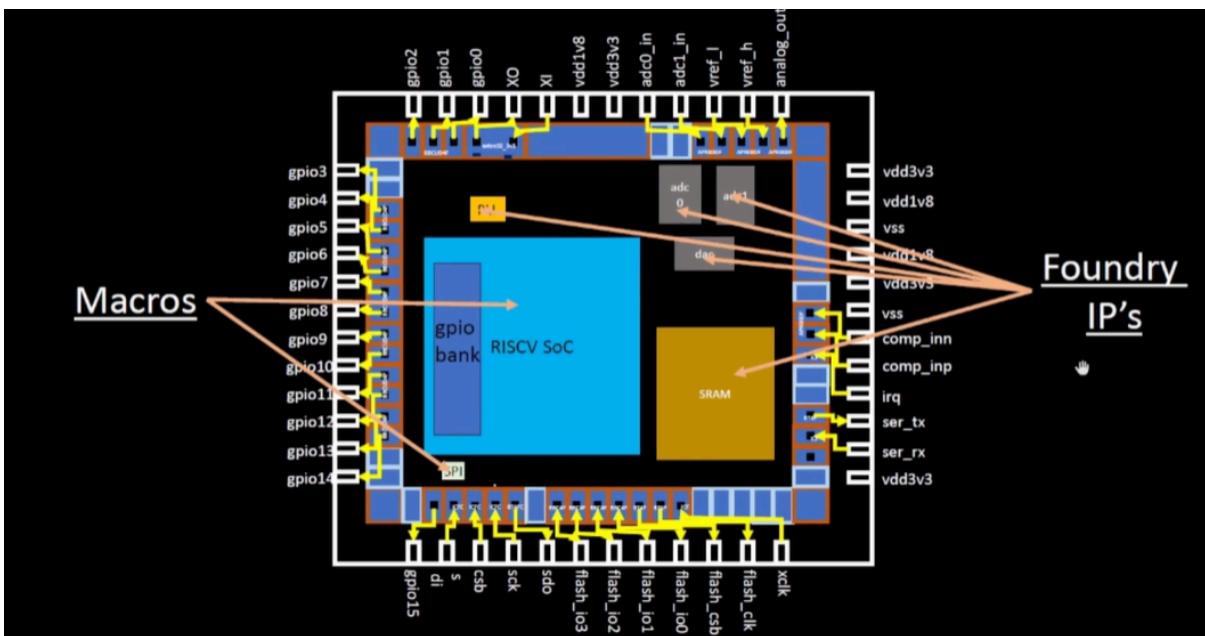


[CHIP]

**FOUNDRY:** All performance, all devices are dependent on it. It is a big factory where chips get manufactured,

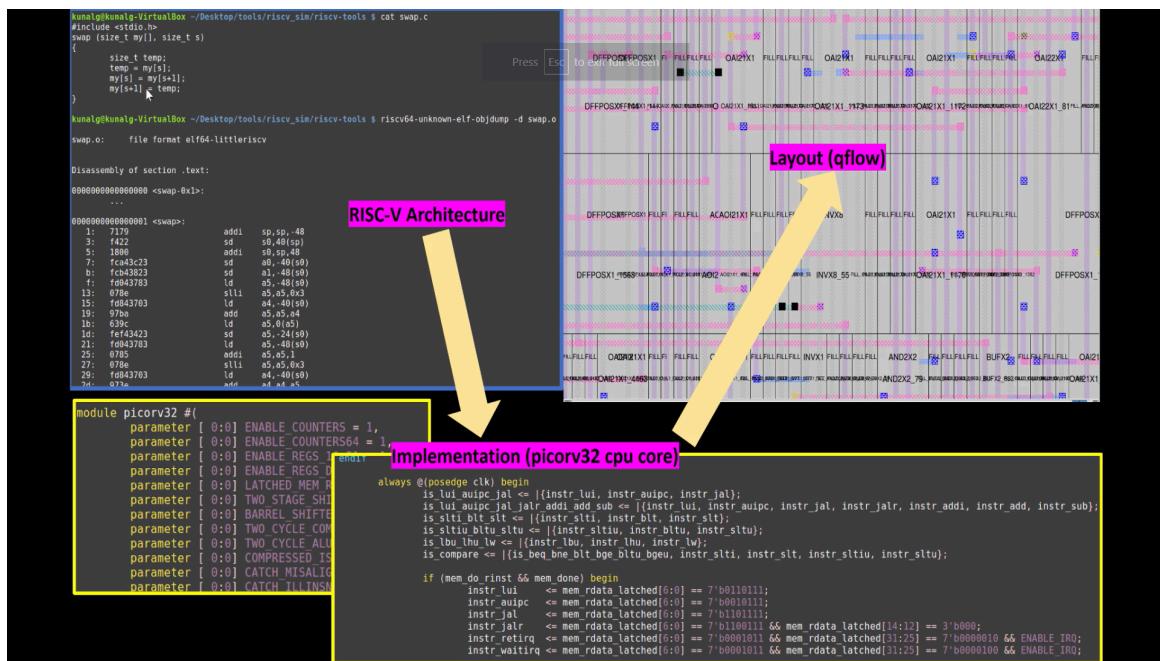
**IP:** Intellectual Property and it needs intelligence to build the blocks.

**MACROS:** Pure digital logic



[ FOUNDRY IPS AND MACROS INSIDE A CHIP]

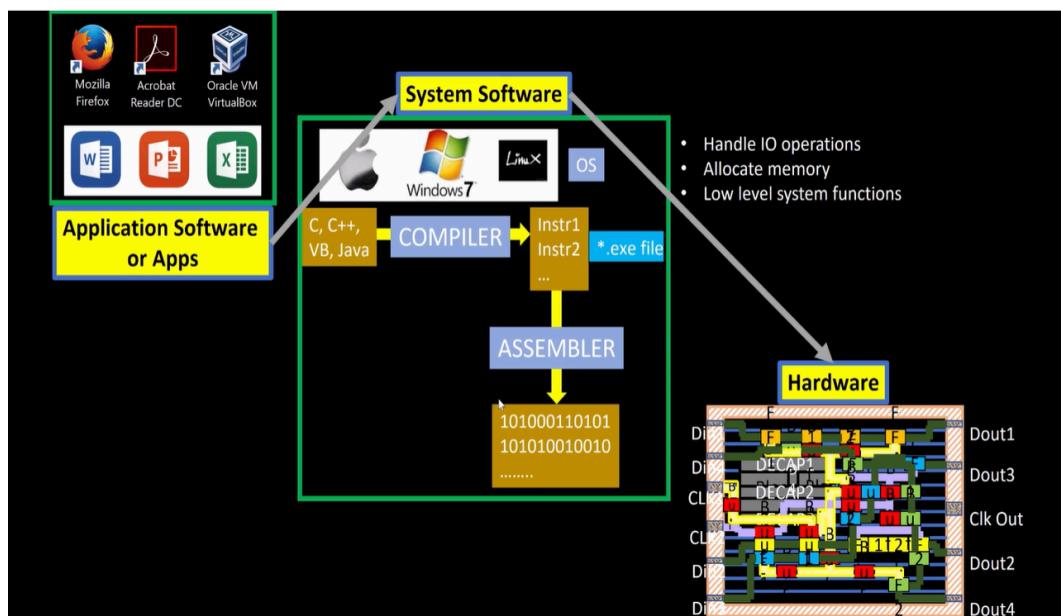
## SKY\_L2 - Introduction to RISC-V



RISC-V is the language of the computer. If we want the C program to run on the particular layout, we need to pass the information to the hardware in particular terms. The C program is compiled and converted into machine language.

One more interface is needed to be present between RISC-V architecture and layout and that is the hardware description language. RISC-V specifications need to be implemented using some RTL.

## SKY\_L3 - From Software Applications to Hardware



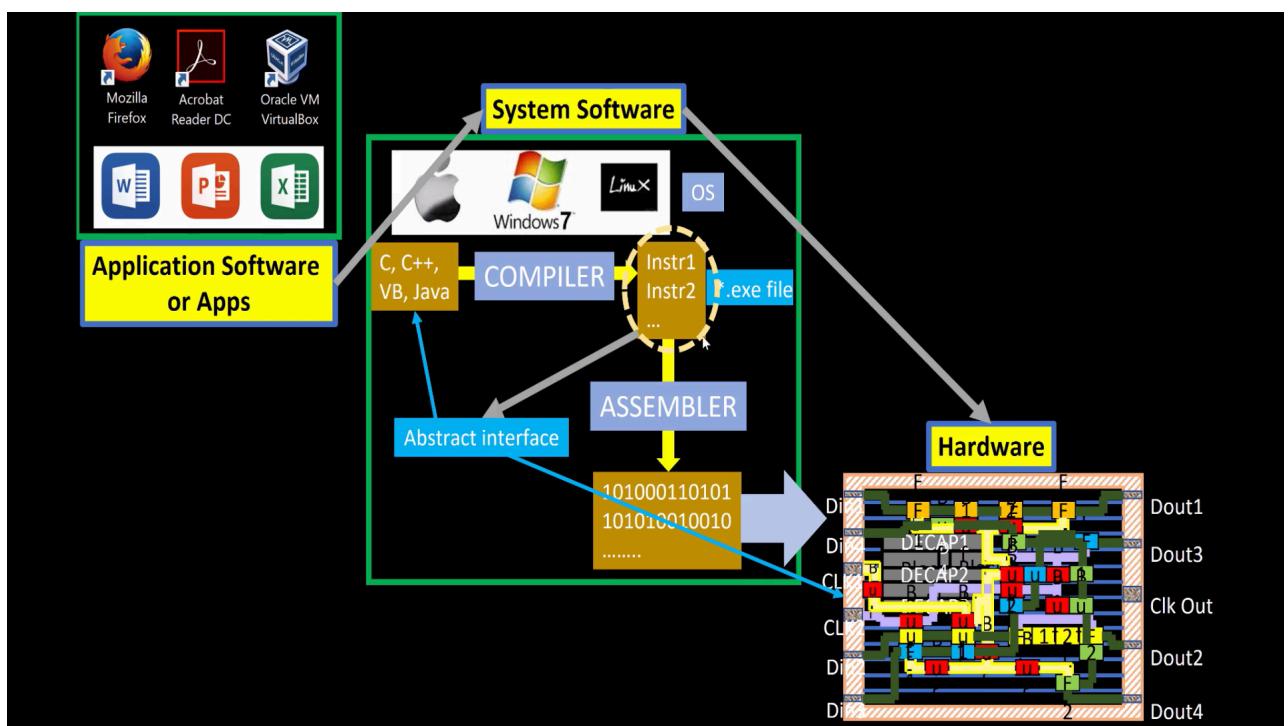
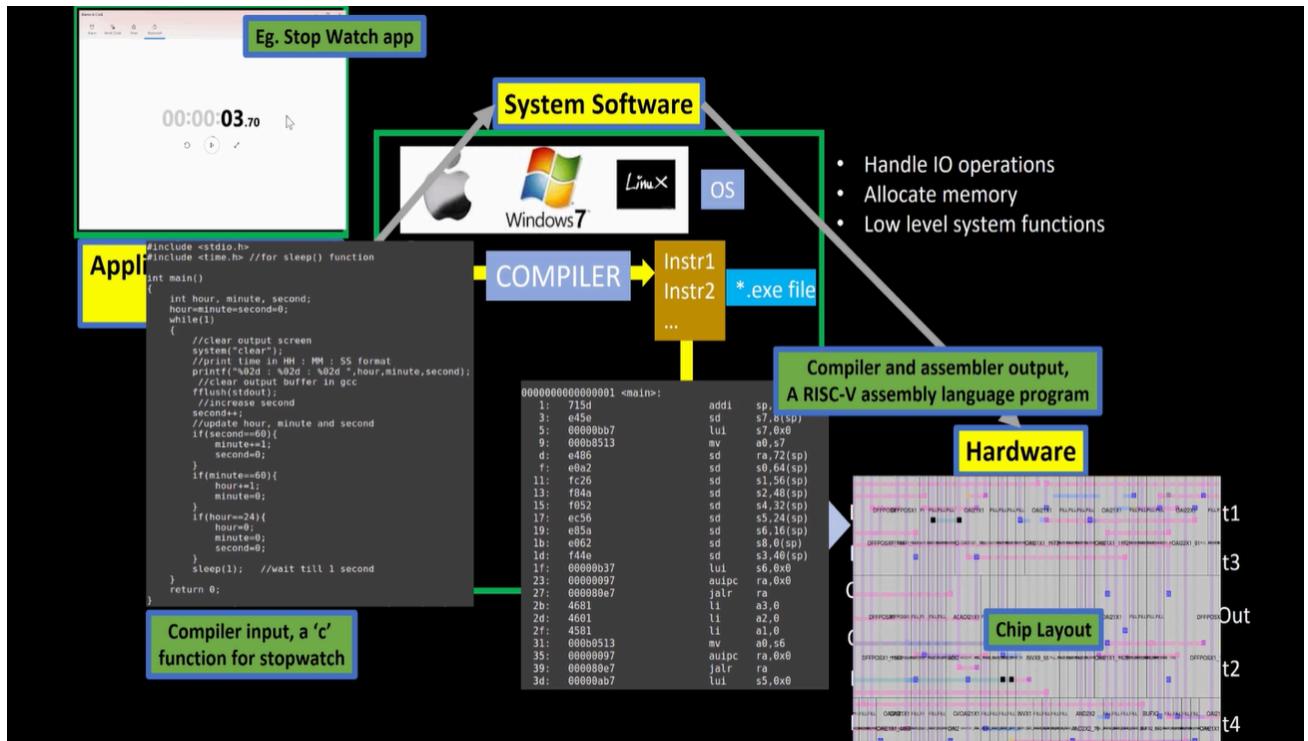
Application Software enters into system software which converts the program into binary language.

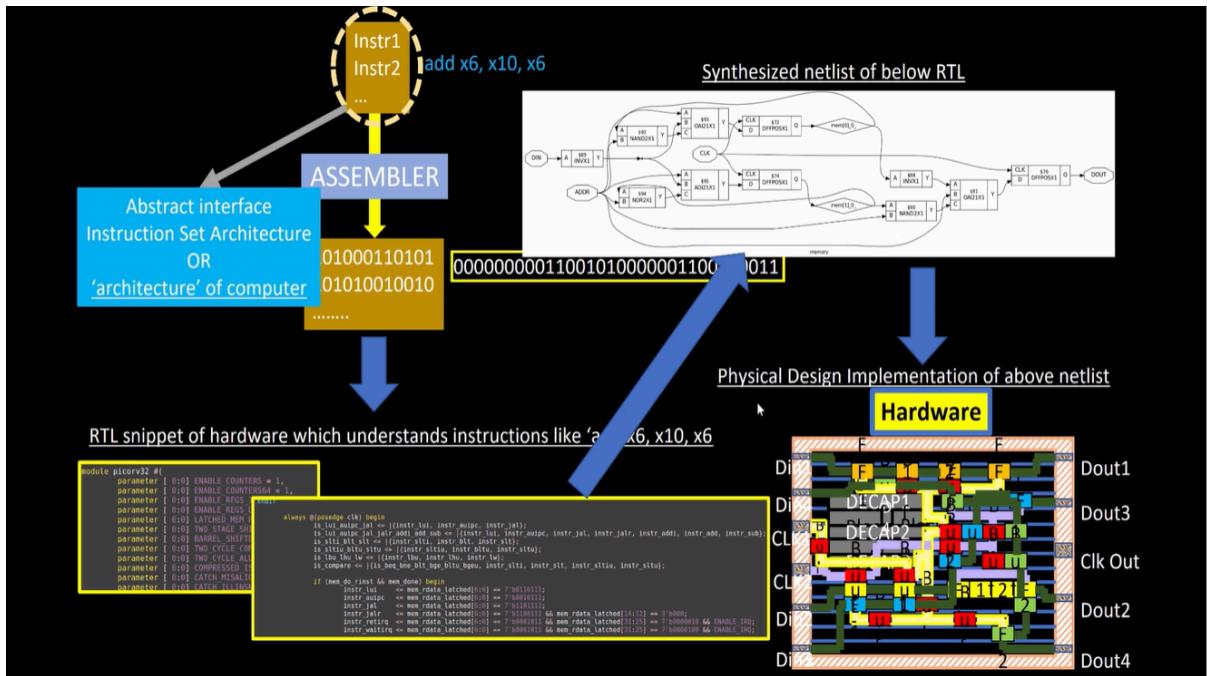
Major components of system software are OS, COMPILER and ASSEMBLER. Job of an OS is to take the app and convert it into a respective assembly language program so that it is understood by the hardware.

The output of the OS is small functions in C,C++ JAVA language which using their respective compiler are converted into instructions. The syntax of the instructions depends on the type of hardware.

The work of the assembler is to convert instructions into binary language. Then this binary code is fed to the hardware which performs the specific function.

ex:





Start from the instruction set, get specification of the instruction set, write a hardware description language of the instruction, synthesise it into the gate level, then the gate level is converted into its respective layout using RTL to GDS.