

RP CPU Fundamentals

As with all digital CPU, this processor processes digital signals, namely 0s and 1s. Using series of predefined streams of 0s and 1s, RP Enterprise have come up with the most brilliant set of instructions to process advance computational work. For this processor, there are **4 general purpose registers** and **3 system registers** to store values and allow instructions to perform operations with the registers.

The Top Secret Architecture:

The 4 general purpose registers are labeled as **EAX, EBX, ECX, and EDX** that maps to binary code **0000, 0001, 0010, and 0011** respectively.

The 3 system registers **EIP, EBP, and ESP** that maps to binary code **0100, 0101, and 0110** respectively

RP Assembly

The RP assembly is the lowest level of programming language for this processor. Each **operation** (or Op for short) maps to a single binary Op code. Each **Op, register, or immediate** (some constant) is represent by **4 bits**. Each instructions begins with an Op, followed 2 operands, which can be register or immediate. One instruction has a total of **12-bits**. If the instruction uses an immediate, the operand that is the immediate is padded with 0s, and the next 12 bits after the instruction is the immediate.

OP	OP code	Machine Code	Description
MOV	0000 0001	[OP REG] + [IMM] [OP REG REG]	e.g MOV 5, EAX – move number 5 into EAX register MOV EAX, EBX – move EAX into EBX
PUSH	0010	[OP REG]	Decrements the stack pointer onto the stack at the specified location
POP	0011	[OP REG]	Loads a value to the location specified and decrements the stack pointer
ADD	0100 0101	[OP REG] + [IMM] [OP REG REG]	ADD 5, EAX – add 5 to EAX register ADD EAX, EBX – add the value of EAX register to EBX register
CALL	0110	[OP] + [IMM]	Calls a function at address specified. Function parammters are first pushed on to the stack. e.g. MOV 10, EAX PUSH EAX CALL 1234
RET	0111	[OP]	Pops saved EIP (return address) and jumps to that location

Where OP = Op code, REG = Reg code, IMM = immediate (constant value)

An instruction is made up of: **[OP-4-BIT|REG/IMM 4-BIT|REG-4-BIT] + [12-BIT-IMMEDIATE]**

For example an instruction such as:

MOV EAX, EBX is equivalent to 000000000001

MOV 128, ECX is equivalent to 000100100000000010000000

RET is equivalent to 011100000000

With this amazing instruction set architecture. RP enterprise will soon rule the world.