**ASSIGNING OPCODE TO DIFFERENT TYPES OF INSTRUCTIONS**

Assuming we are working on a MP with size of register as 16-bit.

Also assuming we have address of 16-bit as well.

|  |  |
| --- | --- |
| **Instruction** | **Opcode** |
| Move | 0x0X |
| Load | 0x1X |
| Store | 0x2X |
| Arithmetic Operation | 0x3X |
| Comparison Operation | 0x4X |
| Branching | 0x5X |

X = (0 - 9)

**Naming of Register**

There are 7 registers in our Microprocessor:

Syntax: **R** Number of register (1 - 7)

R1, R2, R3, R4, R5, R6, R7

**INSTRUCTIONS SET**

M : 16-bit memory address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **INSTRUCTION** | **OPCODE** | **OPERAND** | **OPERATION** | **EXAMPLE** |
| MOV | 0x00 | R1, R2 | Copy value of Register into another | MOV R1, R2 |
| LD | 0x10 | R, M | Copy value at memory(M) into Register | MOV R1, $3000 |
| LD | 0x11 | R, 16-bit value | Set value of Register as value | LD R1, 0xFFFF |
| STL | 0x20 | R1, R2 | Store value of Register R1 into the address stored at R2 | STL R1, R2 |
| STL | 0x21 | R, M | Store value of Register into Memory location (M) | STL R1, $3000 |
| ADD | 0x30 | R1, R2 | Perform addition of content of both registers and store in R1 | ADD R1, R2 |
| ADD | 0x31 | R, 16-bit value | Perform Addition of content of Register with the value and store the result back in Register | ADD R1, 1 |
| SUB | 0x32 | R1, R2 | Perform subtraction of content of both registers and store in R1 | SUB R1, R2 |
| SUB | 0x33 | R, 16-bit value | Perform Subtraction of content of Register with the value and store the result back in Register | SUB R1, 1 |
| CMP | 0x40 | R, M | Compares value of register and content at the address | CMP R1, 0x3000 |
| CMP | 0x41 | R, 16-bit value | Compare value of Register with the value and set the flags as per the result | CMP R1, 10 |
| BLT | 0x50 | Label | If Less than flag is 1 then go to label | BLT label |
| BGT | 0x51 | Label | If Greater than flag is 1 then go to label | BGT label |
| BEQ | 0x52 | Label | If Equal flag is 1 then go to label | BEQ label |
| UCB | 0x53 | Label | Unconditional jump to label | UCB label |