1. I-type:

|  |  |
| --- | --- |
| Bits range | Used to store |
| 0-5 | Immediate number |
| 6-8 | Destination register |
| 9-11 | Operand |
| 12-15 | Opcode |

15 12 11 9 8 6 5 0

|  |  |  |  |
| --- | --- | --- | --- |
| Opcode | Operand | Dest. register | Immediate Number |

1. R-type:

|  |  |
| --- | --- |
| Bits range | Used to store |
| 0-2 | Not used |
| 3-5 | Destination register |
| 6-8 | Second Operand |
| 9-11 | First Operand |
| 12-15 | Opcode |

15 12 11 9 8 6 5 3 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Opcode | 1st operand | 2nd Operand | Dest. Register | Not used |

1. J-type:

|  |  |
| --- | --- |
| Bits range | Used to store |
| 0-11 | Immediate number |
| 12-15 | Opcode |

|  |  |
| --- | --- |
| Opcode | Immediate Number |

15 12 11 0

|  |  |
| --- | --- |
| li | li $r1, x  Opcode is 0000 (12-15 bits).  11-9 bits are empty.  8-6 bits are for $r1.  5-0 bits are for immediate number x. |
| add | Add $r1, $r2, $r3  Opcode is 0001 (12-15 bits).  11-9 bits are for $r2.  8-6 bits are for $r3.  5-3 bits are for $r1.  2-0 bits are empty. |
| and | And $r1, $r2, $r3  Opcode is 0010 (12-15 bits).  11-9 bits are for $r2.  8-6 bits are for $r3.  5-3 bits are for $r1  2-0 bits are empty |
| or | Or $r1, $r2, $r3  Opcode is 0011 (12-15 bits).  11-9 bits are for $r2.  8-6 bits are for $r3.  5-3 bits are for $r1  2-0 bits are empty |
| load | Load $r1, $r2  Opcode is 0100 (12-15 bits).  11-9 bits are for $r2  8-6 bits are for $r1  5-0 bits are empty |
| store | store $r1, $r2  Opcode is 0101 (12-15 bits).  11-9 bits are for $r2  8-6 bits are for $r1  5-0 bits are empty |
| move | Move $r1, $r2  Opcode is 0110 (12-15 bits).  11-9 bits are for $r2  8-6 bits are for $r1  5-0 bits are empty |
| addi | addi $r1, $r2, x  Opcode is 0111 (12-15 bits).  11-9 bits are for $r2.  8-6 bits are for $r1.  5-0 bits are for immediate number x. |
| andi | andi $r1, $r2, x  Opcode is 1000 (12-15 bits).  11-9 bits are for $r2.  8-6 bits are for $r1.  5-0 bits are for immediate number x. |
| ori | ori $r1, $r2, x  Opcode is 1001 (12-15 bits).  11-9 bits are for $r2.  8-6 bits are for $r1.  5-0 bits are for immediate number x. |
| ble | ble $r1, $r2, x  Opcode is 1010 (12-15 bits).  11-9 bits are for $r1.  8-6 bits are for $r2.  5-0 bits are for immediate number x. |
| bne | bne $r1, $r2, x  Opcode is 1011 (12-15 bits).  11-9 bits are for $r1.  8-6 bits are for $r2.  5-0 bits are for immediate number x. |
| jump | Jump x  Opcode is 1100 (12-15 bits).  11-0 bits are for immediate number x. |
| call | call x  Opcode is 1101 (12-15 bits).  11-0 bits are for immediate number x. |
| rtn | rtn  Opcode is 1110. (12-15 bits)  All other bits are empty. |
| halt | halt  Opcode is 1111. (12-15 bits)  All other bits are empty |

**Test program 1:**

|  |  |  |
| --- | --- | --- |
| instruction | machine code (binary) | machine code (hex) |
| li $r1, 1 | 0000 000 000 000001 | 0001 |
| li $r2, 2 | 0000 000 001 000010 | 0042 |
| li $r3, 10 | 0000 000 010 001010 | 008A |
| add $r2, $r1, $r2 | 0001 000 001 001 000 | 1048 |
| ble $r2, $r3, -1 | 1010 001 010 111111 | A2BF |
| halt | 1111 000000000000 | F000 |

**Test program 2:**

|  |  |  |
| --- | --- | --- |
| instruction | machine code (binary) | machine code (hex) |
| li $r1, 6 | 0000 000 000 000110 | 0006 |
| li $r2, 5 | 0000 000 001 000101 | 0045 |
| andi $r3, $r1, 3 | 1000 000 010 000011 | 8083 |
| ori $r4, $r3, 8 | 1001 010 011 001000 | 94C8 |
| halt | 1111 000000000000 | F000 |

**Test program 3:**

|  |  |  |
| --- | --- | --- |
| instruction | machine code (binary) | machine code (hex) |
| li $r1, 6 | 0000 000 000 000110 | 0006 |
| li $r2, 5 | 0000 000 001 000101 | 0045 |
| and $r3, $r1, $r2 | 0010 001 000 010 000 | 2210 |
| li $r8, 0 | 0000 000 111 000000 | 01C0 |
| store $r3, $r8 | 0101 111 010 000000 | 5E80 |
| or $r4, $r1, $r2 | 0011 000 001 011 000 | 3058 |
| li $r8, 1 | 0000 000 111 000001 | 01C1 |
| store $r4, $r8 | 0101 111 011 000000 | 5EC0 |
| li $r8, 1 | 0000 000 111 000001 | 01C1 |
| load $r7, $r8 | 0100 111 110 000000 | 4F80 |
| halt | 1111 000000000000 | F000 |

**Test program 4:**

|  |  |  |
| --- | --- | --- |
| instruction | machine code (binary) | machine code (hex) |
| li $r1, 6 | 0000 000 000 000110 | 0006 |
| li $r2, 4 | 0000 000 001 000100 | 0044 |
| call 7 | 1101 000000000111 | D007 |
| move $r4, $r3 | 0110 010 011 000000 | 64C0 |
| li $r1, 7 | 0000 000 000 000111 | 0007 |
| li $r2, 8 | 0000 000 001 001000 | 0048 |
| call 3 | 1101 000000000011 | D003 |
| move $r5, $r3 | 0110 010 100 000000 | 6500 |
| jump 3 | 1100 000000000011 | C003 |
| add $r3, $r1, $r2 | 0001 000 001 010 000 | 1050 |
| rtn | 1110 000000000000 | E000 |
| halt | 1111 000000000000 | F000 |