Conjunto de instruções (ISA)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Opcode | Tipo | Mnemonic | Nome | Operação |
| 0000 | R | brzr | Branch on Zero Register | if (R[ra] == 0) PC = R[rb] |
| 0001 | I | ji | Jump Immediate | PC = PC + Imm. |
| 0010 | R | ld | Load | R[ra] = M[ R[rb] ] |
| 0011 | R | st | Store | M[ R[rb] ] = R[ra] |
| 0100 | I | addi | Add Immediate | R[0] = R[0] + Imm |
| 0101 | I | addui | Add Unsigned Immediate | R[0] = R[0] + Imm(ext zero) |
| 0110 | I | addupi | Add Upper Immediate | R[0] = R[0] + (Imm<<4) |
| 0111 |  |  |  |  |
| 1000 | R | not | Not | R[ra] = not R[rb] |
| 1001 | R | and | And | R[ra] = R[ra] and R[rb] |
| 1010 | R | or | Or | R[ra] = R[ra] or R[rb] |
| 1011 | R | xor | Xor | R[ra] = R[ra] xor R[rb] |
| 1100 | R | add | Add | R[ra] = R[ra] + R[rb] |
| 1101 | R | sub | Sub | R[ra] = R[ra] - R[rb] |
| 1110 | R | slr | Shift Left Register | R[ra] = R[ra] << R[rb] |
| 1111 | R | srr | Shift Right Register | R[ra] = R[ra] >> R[rb] |

|  |  |  |
| --- | --- | --- |
| Opcode | Hexa | Sinal de Controle |
| 0000 | 0 | 10000100001 |
| 0001 | 1 | 01010010000 |
| 0010 | 2 | 00001000000 |
| 0011 | 3 | 00100000000 |
| 0100 | 4 | 01011001000 |
| 0101 | 5 | 10011001000 |
| 0110 | 6 | 11011001000 |
| 0111 | 7 | -------------------- |
| 1000 | 8 | 00001001111 |
| 1001 | 9 | 00001001100 |
| 1010 | a | 00001001101 |
| 1011 | B | 00001001110 |
| 1100 | c | 00001001000 |
| 1101 | d | 00001001001 |
| 1110 | e | 00001001010 |
| 1111 | f | 00001001011 |

Diagrama

O conteúdo gerado por IA pode estar incorreto.