Tipo de datos

* 8 bits

**El set de instrucciones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mnemotécnico** | **Nombre** | **# operandos** | **Descripción** | **Binario** |
| **Transferencia de datos** | | | | |
| MOV | Mover | MOV Op1, Op2  MOV REG, REG  MOV VAL, REG | Mueve los datos desde un registro o un valor (Op1) hacia otro registro (Op2). | 00000 |
| CAR | Cargar | CAR Op1, Op2  CAR [REG], REG  CAR [VAL], REG | Carga los datos desde un espacio de memoria (Op1) hacia un registro (Op2). | 00001 |
| GRD | Guardar | GRD Op1, Op2  GRD REG, [REG]  GRD VAL, [REG] | Guarda los datos desde un registro o un valor (Op1) hacia un espacio en la memoria (Op2). | 00010 |
| Instrucciones aritméticas | | | | |
| SUM | Sumar | SUM Op1, Op2  SUM REG, REG  SUM VAL, REG | Suma entre un valor o un registro (Op1) y otro registro (Op2), para luego guardar el resultado en el Op2. | 00101 |
| RES | Restar | RES Op1, Op2  RES REG, REG  RES VAL, REG | Resta entre un valor o un registro (Op1) y otro registro (Op2), para luego guardar el resultado en el Op2. | 00110 |
| MUL | Multiplicar | MUL Op1, Op2  MUL REG, REG  MUL VAL, REG | Multiplica un valor o registro (Op1) por otro registro (Op2) y guarda el resultado en Op2. | 00111 |
| DIV | Dividir | DIV Op1, Op2  DIV REG, REG  DIV VAL, REG | Divide un valor o un registro (Op1) entre un registro (Op2) y guarda el resultado en Op2. | 01000 |
| **Instrucciones lógicas** | | | | |
| AND | Operación de Y lógico | AND Op1, Op2  AND REG, REG | Realiza la operación de Y lógico entre dos registros y guarda el resultado en el segundo registro (Op2). | 01001 |
| OR | Operación de O lógico | OR Op1, Op2  OR REG, REG | Realiza la operación de O lógico entre dos registros y guarda el resultado en el segundo registro (Op2). | 01010 |
| NOT | Negación lógica | NOT Op1  NOT REG | Niega los bits de un registro | 01011 |
| XOR | Operación de XOR lógico | XOR Op1, Op2  XOR REG, REG | Realiza la operación de XOR lógico entre dos registros y guarda el resultado en el segundo registro (Op2). | 01100 |
| **Instrucciones de control de transferencia** | | | | |
| SAL | Saltar | SAL Op1  SAL REG  SAL VAL | Salta hacia la dirección en la memoria que puede estar representado por un valor o un registro (Op1). | 01101 |
| SMQ | Saltar si menor que | SMQ Op1  SMQ REG  SMQ VAL | Salta hacia la dirección en la memoria que puede estar representado por un valor o un registro (Op1). | 01110 |
| SMI | Saltar si menor o igual que | SMI Op1  SMI REG  SMI VAL | Salta hacia la dirección en la memoria que puede estar representado por un valor o un registro (Op1). | 01111 |
| SGQ | Saltar si mayor que | SGQ Op1  SGQ REG  SGQ VAL | Salta hacia la dirección en la memoria que puede estar representado por un valor o un registro (Op1). | 10000 |
| SGI | Saltar si mayor o igual que | SGI Op1  SGI REG  SGI VAL | Salta hacia la dirección en la memoria que puede estar representado por un valor o un registro (Op1). | 10001 |
| SIQ | Saltar si igual que | SIQ Op1  SIQ REG  SIQ VAL | Salta hacia la dirección en la memoria que puede estar representado por un valor o un registro (Op1). | 10010 |
| SDQ | Saltar si diferente que | SDQ Op1  SDQ REG  SDQ VAL | Salta hacia la dirección en la memoria que puede estar representado por un valor o un registro (Op1). | 10011 |
| ALT | Alto | ALT | Termina la ejecución. | 10111 |
| **Instrucciones de comparación** | | | | |
| CMP | Comparar | CMP Op1, Op2  CMP REG, REG  CMP VAL, REG | Compara un valor o un registro (Op1) con otro registro (Op2) y modificará las banderas (flags) según el resultado. | 11000 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operaciones de transferencia de datos** | | | | |
| **Instrucción** | **Buscar instrucción** | **Decodificar instrucción** | **Buscar en dirección de memoria** | **Ejecutar instrucción** |
| MOV | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar |  | 1. R3 <- R2 R3 <- VAL |
| CAR | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. MAR <- R1 MAR <- VAL 2. MBR <- [MAR] | 1. R2 <- MBR |
| GRD | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. MAR <- R1 MAR <- VAL 2. MBR <- R2 | 1. [MAR] <- MBR |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operaciones artimeticas** | | | | | |
| **Instrucción** | **Buscar instrucción** | **Decodificar instrucción** | **Cargar datos en registros especiales** | **Ejecutar instrucción** | **Guardar resultado** |
| SUM | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Op1 <- R0 Op1 <- VAL 2. Op2 <- R1 | Sumar | 1. R1 <- Resultado |
| RES | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Op1 <- R3 Op1 <- VAL 2. Op2 <- R4 | Sumar complemento a 2 | 1. R4 <- Resultado |
| MUL | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Op1 <- R5 Op1 <- VAL 2. Op2 <- R6 | Sumas sucesivas | 1. R6 <- Resultado |
| DIV | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Op1 <- R0 Op1 <- VAL 2. Op2 <- R1 |  | 1. R1 <- Resultado |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instrucciones logicas** | | | | | |
| **Instrucción** | **Buscar instrucción** | **Decodificar instrucción** | **Cargar datos en registros especiales** | **Ejecutar instrucción** | **Guardar resultado** |
| AND | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Op1 <- R0 2. Op2 <- R1 | Sumar | 1. R1 <- Resultado |
| OR | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Op1 <- R0 2. Op2 <- R1 | Sumar complemento a 2 | 1. R1 <- Resultado |
| NOT | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Op1 <- R0 | Sumas sucesivas | 1. R0 <- Resultado |
| XOR | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Op1 <- R0 2. Op2 <- R1 |  | 1. R1 <- Resultado |

|  |  |  |  |
| --- | --- | --- | --- |
| **Instrucciones de control de transferencia** | | | |
| **Instrucción** | **Buscar instrucción** | **Decodificar instrucción** | **Ejecutar instrucción** |
| SAL | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | Salta a la dirección de memoria especificada |
| SMQ | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Verificar flags 2. Saltar a la dirección de memoria especificada |
| SMI | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Verificar flags 2. Saltar a la dirección de memoria especificada |
| SGQ | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Verificar flags 2. Saltar a la dirección de memoria especificada |
| SGI | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Verificar flags 2. Saltar a la dirección de memoria especificada |
| SIQ | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Verificar flags 2. Saltar a la dirección de memoria especificada |
| SDQ | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Verificar flags 2. Saltar a la dirección de memoria especificada |
| ALT | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Verificar flags 2. Saltar a la dirección de memoria especificada |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Instrucciones de control de transferencia** | | | | |
| **Instrucción** | **Buscar instrucción** | **Decodificar instrucción** | **Cargar datos a registros especiales** | **Ejecutar instrucción** |
| CMP | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Op1 <- R0 Op1 <- VAL 2. Op2 <- R1 | 1. Comparar valores 2. Modificar flags |