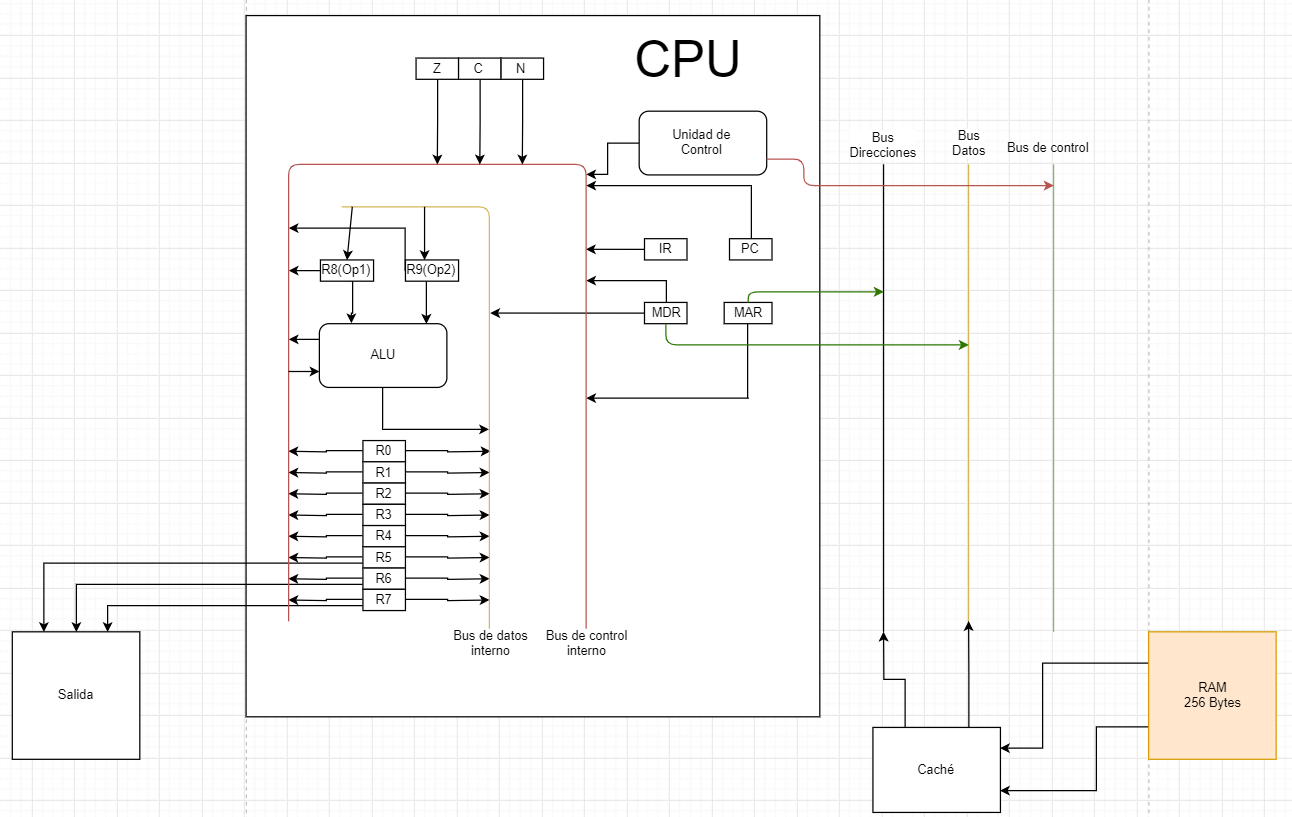
**Hoja de datos Proyecto Final**

**Diagrama de componentes**



* Salida
  + Tres registros con salida directa para poder visualizar su contenido en la pantalla LSD.
* Caché
  + Memoria caché de un solo nivel a la que se accede antes de a la memoria principal.
* RAM
  + Memoria principal del sistema con 256 Bytes de capacidad.
* CPU
  + Unidad de control
    - Realiza la organización del comportamiento de todos los componentes internos del CPU junto con el manejo de la memoria caché y principal en ausencia de una UCM.
  + Registros especializados
    - IR (Instruction Register)
      * Capacidad de 24 bits para almacenar la instrucción que se está realizando.
    - PC (Program Counter)
      * Capacidad de 8 bits para almacenar la dirección de la siguiente instrucción.
    - MAR
      * Capacidad de 8 bits para almacenar la dirección de un dato en la memoria.
    - MBR
      * Capacidad de 24 bits para almacenar datos desde la memoria.
    - R8 y R9
      * Registros en los que se almacenan los operandos para que la ALU realice una operación, ambos registros con una capacidad de 8 bits.
    - R5, R6 y R7
      * Registros que también son de propósito general, pero están conectados con la pantalla para mostrar lo que contengan, cada uno con una capacidad de 8 bits.
  + Registros generales
    - Desde R0 a R7
      * Registros de propósito general, cada uno con una capacidad de 8 bits.
  + Z, C y N
    - Bits de estado que pueden indicar:
      * Overflow: C = 1
      * Igualdad: Z = 1
      * Desigualdad: Z = 0
      * Op1 > Op2: Z = 0; C = 1; N = 0
      * Op1 >= Op2: Z = 1; C = 1; N = 0
      * Op1 < Op2: Z = 0; C = 0; N = 1
      * Op1 <= Op2: Z = 1; C = 0; N = 1
  + ALU
    - Unidad Aritmética Lógica encargada de realizar las operaciones aritméticas, lógicas y las comparaciones.

**La palabra de instrucción**



* OpCode puede tomar los valores según el tipo de instrucción (ver la sección de ISA).
* I/D indica si el primer operando es de direccionamiento inmediato o directo.
* R/M indica si el primer o segundo operando respectivamente pertenece a un registro o a un espacio de memoria.
* Operando 1 puede representar un valor en la misma instrucción, un registro, una dirección en memoria o puede estar vacío en caso de mandar una instrucción ALT.
* Operando 2 puede representar un registro, una dirección en memoria o puede estar vacío en caso de mandar una instrucción ALT o NOT.

**Ejemplos de ensamblador**

* Código ejemplo de una suma y resta mostrando los resultados en distintas posiciones de la pantalla LSD

MOV 5, R0

MOV 3, R1

SUM R0, R1

MOV R1, R7

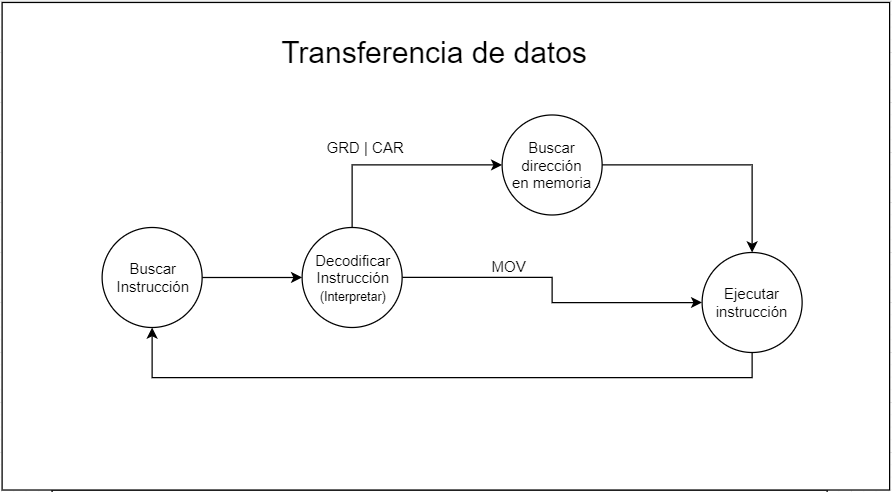
RES 7, R1

MOV R5

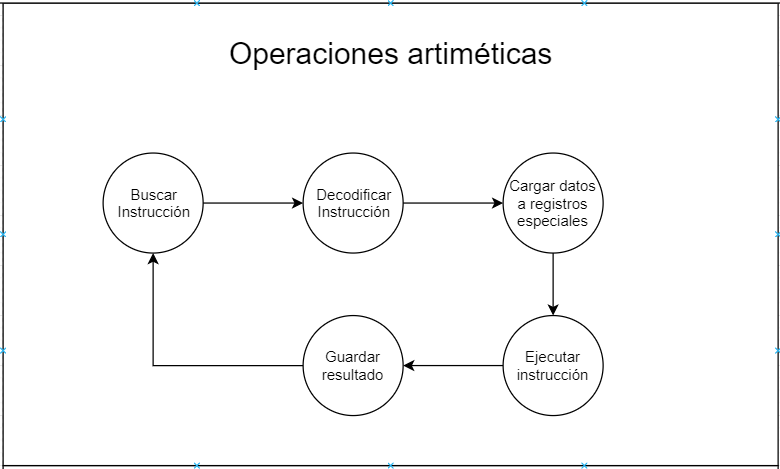
**El set de instrucciones (ISA)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Transferencia de datos** | | | | |
| **Mnemotécnico** | **Nombre** | **# operandos** | **Descripción** | **Binario** |
| 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). | 00001 |
| CAR | Cargar | CAR Op1, Op2  CAR [REG], REG  CAR [VAL], REG | Carga los datos desde un espacio de memoria (Op1) hacia un registro (Op2). | 00010 |
| 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). | 00011 |
| Instrucciones aritméticas | | | | |
| **Mnemotécnico** | **Nombre** | **# operandos** | **Descripción** | **Binario** |
| 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. | 00100 |
| 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. | 00101 |
| 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. | 00110 |
| 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. | 00111 |
| **Instrucciones lógicas** | | | | |
| **Mnemotécnico** | **Nombre** | **# operandos** | **Descripción** | **Binario** |
| 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). | 01000 |
| 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). | 01001 |
| NOT | Negación lógica | NOT Op1  NOT REG | Niega los bits de un registro | 01010 |
| 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). | 01011 |
| **Instrucciones de control de transferencia** | | | | |
| **Mnemotécnico** | **Nombre** | **# operandos** | **Descripción** | **Binario** |
| 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). | 01100 |
| 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). | 01101 |
| 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). | 01110 |
| 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). | 01111 |
| 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. | 10100 |
| **Instrucciones de comparación** | | | | |
| **Mnemotécnico** | **Nombre** | **# operandos** | **Descripción** | **Binario** |
| 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. | 10101 |

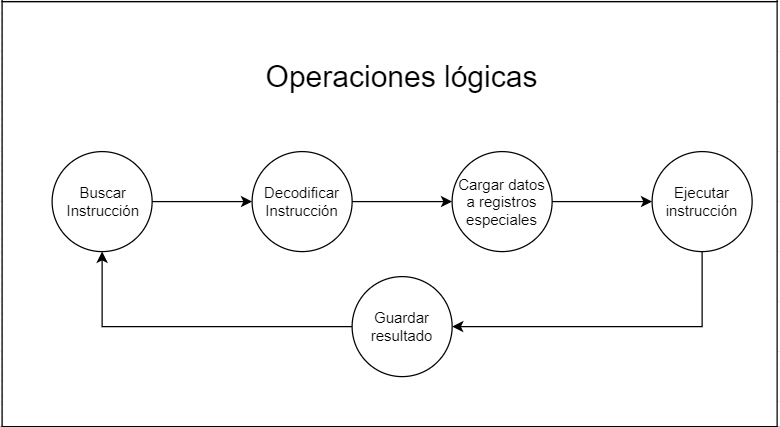
**Micro operaciones**



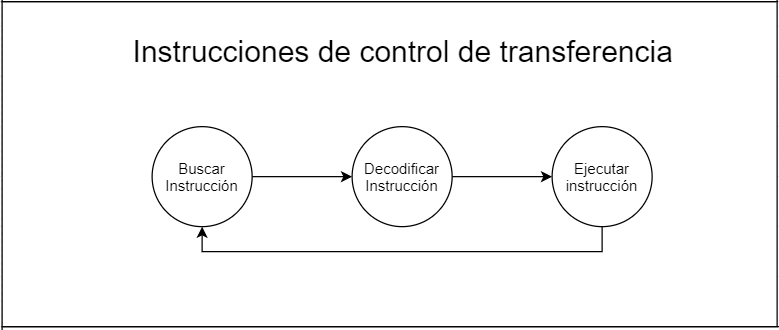
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operaciones de transferencia de datos** | | | | |
| **Instrucción** | **Buscar instrucción** | **Decodificar instrucción** | **Buscar en dirección de memoria** | **Ejecutar instrucción** |
| MOV | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <-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



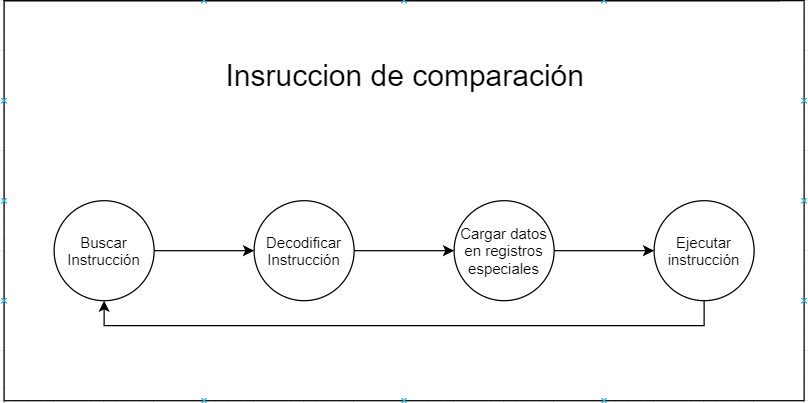
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operaciones aritméticas** | | | | | |
| **Instrucción** | **Buscar instrucción** | **Decodificar instrucción** | **Cargar datos en registros especiales** | **Ejecutar instrucción** | **Guardar resultado** |
| SUM | 1. MAR <- PC 2. PC <- S.I. 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 | Restar | 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 | Multiplicar | 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 | Dividir y truncar | 1. R1 <- Resultado |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instrucciones lógicas** | | | | | |
| **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 | Y lógico | 1. R1 <- Resultado |
| OR | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Op1 <- R0 2. Op2 <- R1 | O lógico | 1. R1 <- Resultado |
| NOT | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Op1 <- R0 | Negación lógica | 1. R0 <- Resultado |
| XOR | 1. MAR <- PC 2. PC <- S.I. 3. MBR <- [MAR] 4. IR <- MBR | Interpretar | 1. Op1 <- R0 2. Op2 <- R1 | XOR lógico | 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 |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Instrucción de comparación** | | | | |
| **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 |

**Árbol de estados**

