**Propuesta de ISA para el Sistema Computacional**

A continuación, se presenta una propuesta de **Arquitectura de Conjunto de Instrucciones (ISA)** diseñada específicamente para el sistema descrito. Este ISA está adaptado para un computador básico implementado con placas Arduino, manejando variables de **4 bits** y capaz de ejecutar las operaciones mínimas requeridas. El objetivo es proporcionar una base sólida para que los estudiantes puedan desarrollar su proyecto, entender el funcionamiento de una CPU y, si lo desean, proponer mejoras al ISA.

**Características Generales del ISA:**

* **Tamaño de palabra (Word Size):** 13 bits por instrucción.
* **Tamaño de los datos:** 8 bits.
* **Número de registros:** 8 registros generales de 8 bits cada uno (R0 a R7).
* **Formato de instrucción:** Instrucciones de longitud fija de 13 bits.
* **Modos de direccionamiento:** Inmediato, registro directo y memoria directa.

**Formato de Instrucción:**

Las instrucciones tendrán un tamaño fijo de **13 bits** y estarán estructuradas de la siguiente manera:

1. **Instrucciones Tipo R (Registro):**
   * **Bits 12-9 (4 bits):** Código de operación (Opcode).
   * **Bits 8-6 (3 bits):** Registro destino (Rd).
   * **Bits 5-3 (3 bits):** Registro fuente 1 (Rs1).
   * **Bits 2-0 (3 bits):** Registro fuente 2 (Rs2).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Formato de instrucción tipo R | | | | | | | | | | | | | |
| # bit | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Descripción | Opcode | | | | Rd | | | Rs1 | | | Rs2 | | |

1. **Instrucciones Tipo I y dos registros:**
   * **Bits 12-9 (4 bits):** Código de operación (Opcode).
   * **Bits 8-6 (3 bits):** Registro destino (Rd).
   * **Bits 5-3 (3 bits):** Registro fuente (Rs).
   * **Bits 2-0 (3 bits):** Valor inmediato (Inm).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Formato de instrucción tipo I | | | | | | | | | | | | | |
| # bit | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Descripción | Opcode | | | | Rd | | | Rs | | | Inmediato | | |

1. **Instrucciones Tipo I y un registro:**
   * **Bits 12-9 (4 bits):** Código de operación (Opcode).
   * **Bits 8-6 (3 bits):** Registro destino (Rd).
   * **Bits 3-0 (4 bits):** Valor inmediato (Inm).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Formato de instrucción tipo I | | | | | | | | | | | | | |
| # bit | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Descripción | Opcode | | | | Rd | | |  |  | Inmediato | | | |

1. **Instrucciones Tipo M (Memoria):**
   * **Bits 15-12 (4 bits):** Código de operación (Opcode).
   * **Bits 11-9 (3 bits):** Registro destino o fuente (dependiendo de la instrucción).
   * **Bits 8-0 (6 bits):** Dirección de memoria.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Formato de instrucción tipo M | | | | | | | | | | | | | |
| # bit | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Descripción | Opcode | | | | Rd/Rs | | | Dirección de memoria | | | | | |

**Registros Disponibles:**

* **R0 a R7:** Registros generales de propósito general de 8 bits cada uno.

1. **Instrucciones De control de flujo:**
   * **Bits 12-9 (4 bits):** Código de operación (Opcode).
   * **Bits 5-0 (6 bits):** Dirección de memoria.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Formato de instrucción tipo M | | | | | | | | | | | | | |
| # bit | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Descripción | Opcode | | | | --- | | | Dirección de memoria | | | | | |

**Códigos de Operación (Opcodes) y Descripción de Instrucciones:**

A continuación, se detallan las instrucciones disponibles en el ISA:

1. **Operaciones Aritméticas y Lógicas (Tipo R):**
   * **ADD (Suma de dos registros):**
     + **Opcode:** 0000
     + **Formato:** 0000 | Rd | Rs1 | Rs2
     + **Descripción:** R[Rd] = R[Rs1] + R[Rs2]
   * **AND (Operación lógica AND entre registros):**
     + **Opcode:** 0001
     + **Formato:** 0001 | Rd | Rs1 | Rs2
     + **Descripción:** R[Rd] = R[Rs1] AND R[Rs2]
   * **OR (Operación lógica OR entre registros):**
     + **Opcode:** 0010
     + **Formato:** 0010 | Rd | Rs1 | Rs2
     + **Descripción:** R[Rd] = R[Rs1] OR R[Rs2]
   * **NOT (Negación de un registro):**
     + **Opcode:** 0011
     + **Formato:** 0011 | Rd | Rs | ---
     + **Descripción:** R[Rd] = NOT R[Rs]
2. **Operaciones con Inmediatos (Tipo I):**
   * **ADDI (Suma de registro e inmediato):**
     + **Opcode:** 0100
     + **Formato:** 0100 | Rd | Rs | Inm (3 bits)
     + **Descripción:** R[Rd] = R[Rs] + Inmediato
   * **LDI (Cargar inmediato en registro):**
     + **Opcode:** 0101
     + **Formato:** 0101 | Rd | -- | Inm (4 bits)
     + **Descripción:** R[Rd] = Inmediato
3. **Operaciones de Movimiento y Memoria (Tipo M):**
   * **LD (Cargar desde memoria a registro):**
     + **Opcode:** 0110
     + **Formato:** 0110 | Rd | Dirección (6 bits)
     + **Descripción:** R[Rd] = MEM[Dirección]
   * **ST (Almacenar registro en memoria):**
     + **Opcode:** 0111
     + **Formato:** 0111 | Rs | Dirección (6 bits)
     + **Descripción:** MEM[Dirección] = R[Rs]
4. **Control de Flujo (Tipo M):**
   * **JMP (Salto incondicional):**
     + **Opcode:** 1000
     + **Formato:** 1000 | Dirección (6 bits)
     + **Descripción:** PC = Dirección
   * **JZ (Salto si cero):**
     + **Opcode:** 1001
     + **Formato:** 1001 | Dirección (6 bits)
     + **Descripción:** Si ZF = 1, entonces PC = Dirección
   * **JNZ (Salto si no cero):**
     + **Opcode:** 1010
     + **Formato:** 1010 | Dirección (6 bits)
     + **Descripción:** Si ZF = 0, entonces PC = Dirección

**Codificación de Registros:**

* **Registros (3 bits):**
  + 000 - R0
  + 001 - R1
  + 010 - R2
  + 011 - R3
  + 100 - R4
  + 101 - R5
  + 110 - R6
  + 111 - R7

**Flags (Banderas) del Procesador:**

* **Zero Flag (ZF):** Se establece en 1 si el resultado de una operación es cero.