ISA procesador monociclo

REGISTROS: 16 de 16bits

2 registros especiales

r0 🡪 registro 0, su valor es constante

PC 🡪 contador de programa de 16 bits

r1-r15 registros de propósito general

DIRECCION DE MEMORIA de 8 bits

INSTRUCCIONES

Instrucciones de 16 bits.

|  |  |  |  |
| --- | --- | --- | --- |
| 4 bits | 4bits | 4bits | 4bits |
| Op-code | Rd | Rf1 | Rf2 |

|  |  |
| --- | --- |
| **Instrucción** | **OPCODE** |
| Suma | 0000 |
| Resta | 0001 |
| O | 0010 |
| Y | 0011 |
| Carga | 0100 |
| Almacena | 0101 |
| Sumai | 0110 |
| Igual | 0111 |
| mayor | 1000 |
|  | 1001 |
|  | 1010 |

**Suma**:

Descripción: suma dos registros y el resultado lo guarda en un registro destino.

Op-code: 0000

Formato de la instrucción:

Rd, rf1, rf1

Rd= Rf1 + Rf2

**RESTA:**

Descripción: resta dos registros y el resultado lo guarda en un registro destino.

Op-code: 0001

Formato de la instrucción:

Rd, rf1, rf1

Rd= Rf1 - Rf2

**O:**

Descripción: or a nivel de bits entre dos registros y el resultado lo guarda en un registro destino.

Op-code: 0010

Formato de la instrucción:

Rd, rf1, rf1

Rd= rf1 || rf2

**Y:**

Descripción: and a nivel de bits entre dos registros y el resultado lo guarda en un registro destino

Op-code: 0011

Formato de la instrucción:

Rd, rf1, rf2

Rd= rf1 && rf2

**Carga:**

Descripción: carga en un registro destino el valor almacenado en la dirección de memoria que surgen de la suma de dos registros.

Op-code: 0100

Formato de la instrucción:

Rd, rf1, rf2

Rd= rf1+rf2 🡨 suma dos registros, pero en vez de almacenar el resultado almacena la dirección de memoria

**Almacena:**

Descripción: almacena en la dirección de memoria especificada en un registro destino el valor guardado en la dirección de memoria que surge de la suma de dos registros.

Op-code: 0101

Formato de la instrucción:

Rd, rf1, rf2

Rd[dirección especificada]= [valor almacenado en dirección de] rf1 + rf2

**Sumai:**

Descripción: suma un registro mas un inmediato y guarda el resultado en un registro destino.

Op-code: 0110

Formato de la instrucción:

Rd, rf1, inm

Rd= rf1 + inm

**Nop:**

Descripción: “no operation”, no realiza ninguna operación.

Op-code: 0111

Formato de la instrucción:

Nop 🡨 solo se coloca la instrucción, realiza como un “salto”. (ciclo de reloj)

**Not:**

Descripción: no lógico ¿¿MICROARQUITECTURA??

Op-code: 1000

Formato de la instrucción: xor y not?¿

**Igual:**

Descripción: compara dos registros, si son iguales salta al valor que se coloque [PC+inm]

Op-code: 0111

Formato de la instrucción:

Rf1, rf2, inm

[rf1==rf2] 🡪 PC + inm (salto del contador de programa a la dirección del inmediato)

**Mayor:**

Descripción: compara dos registros, si el primero es mayor que el segundo salta al valor que se coloque [PC+inm]

Op-code: 1000

Formato de la instrucción:

Rf1, rf2, inm

[rf1>rf2]🡪 PC + inm (salto del contador de programa a la dirección del inmediato)

PSEUDOINSTRUCCIONES:

**Mayorig:**

Descripción: compara dos registros, si el primero es mayor o igual al segundo salta al valor que se coloque [PC+inm]

Formato de la instrucción:

Rf1, rf2, inm

[rf1>rf2 || rf1 ==rf2]🡪 PC + inm (salto del contador de programa a la dirección del inmediato)

**Inc:**

Descripción: incrementa un registro en 1

Formato de la instrucción:

Inc rf1

Rf1 = rf1 + 1

**Dec:**

Descripción: decrementa un registro en 1

Formato de la instrucción:

Dec rf1

Rf1 = rf1 - 1