ISA procesador monociclo.

* 16 registros de propósito general de 16 bits
* Instrucciones de 16 bits
* PC de 16 bits
* 8 bits por dirección de memoria

REGISTROS

* Registro especial x0 -> mantiene siempre su valor en 0
* PC -> contador de programa

|  |  |
| --- | --- |
| *Registros de propósito general* | |
| 0 | x0 |
| 1 | xa |
| 2 | xb |
| 3 | xc |
| 4 | xd |
| 5 | xe |
| 6 | xf |
| 7 | xg |
| 8 | xh |
| 9 | xi |
| 10 | xj |
| 11 | xk |
| 12 | xl |
| 13 | xm |
| 14 | xn |
| 15 | xw |

INSTRUCCIONES

|  |  |  |
| --- | --- | --- |
| **OPCODE** | **Instrucción** | **RTL BASICO** |
| 0000 | Suma rd, rf1, rf2 | R[rd]= R[rf1]+R[rf2] |
| 0001 | Resta rd, rf1, rf2 | R[rd]= R[rf1]-R[rf2] |
| 0010 | O rd, rf1, rf2 | R[rd]= R[rf1] || R[rf2] |
| 0011 | Y rd, rf1, rf2 | R[rd]= R[rf1] && R[rf2] |
| 0100 | Carga rd, rf1, rf2 | R[rd]= M{R[rf1]+R[rf2]} |
| 0101 | Almacena rf1, rf2, rd | M{R[rf1]+R[rf2]}= R[rd] |
| 0110 | Sumai rd, rf1, inm | R[rd]= R[rf1]+SignExt[inm] |
| 0111 | Not rf1 | R[rf1]= !R[rf1] |
| 1000 | Igual rf1, rf2, inm | Si(R[rf1]==R[rf2])-> PC=PC+ SignExt[inm]  Sino PC= PC + 16 |
| 1001 | Mayor rf1, rf2, inm | Si(R[rf1]>R[rf2])-> PC=PC+ SignExt[inm]  Sino PC= PC + 16 |

PSEUDOINSTRUCCIONES

|  |  |  |  |
| --- | --- | --- | --- |
| Opcode | Pseudoinstrucción |  |  |
| 1010 | Inc | Inc rf1 | Sumai rf1, 1 |
| 1011 | Dec | Dec rf1 | Sumai rf1, -1 |
| 1100 | nop | nop | Sumai x0, 0 <- un salto |
| 1101 | Mayorig | Mayorig rf1, rf2, inm | Si (igual rf1, rf2, inm)  Sino (Mayor rf1, rf2, inm)  Sino PC=PC+16 |

*FORMATO DE LAS INSTRUCCIONES:*

*4bits 4bits 4bits 4bits*

|  |  |  |  |
| --- | --- | --- | --- |
| TIPO-R | | | |
| Registro-fuente-2 | Registro-fuente-1 | Registro-destino | Op-code |
| TIPO-M | | | |
| Registro-fuente-2 | Registro-fuente-1 | Registro-destino | Op-code |
| TIPO-I | | | |
| Inmediato | Registro-fuente-1 | Registro-destino | Op-code |
| TIPO-J | | | |
| Inmediato | Registro-fuente-2 | Registro-fuente-1 | Op-code |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Instrucción | 4bits | 4bits | 4bits | 4bits | Tipo |
|  | Opcode |  |  |  |  |
| Suma | 0000 | Rd | Rf1 | Rf2 | R |
| Resta | 0001 | Rd | Rf1 | Rf2 | R |
| O | 0010 | Rd | Rf1 | Rf2 | R |
| Y | 0011 | Rd | Rf1 | Rf2 | R |
| Not | 0111 | Rf1 | X | X | R |
|  |  |  |  |  |  |
| Sumai | 0110 | Rd | Rf1 | Inm | I |
|  |  |  |  |  |  |
| Carga | 0100 | Rd | Rf1 | Rf2 | M |
| Almacena | 0101 | Rf1 | Rf2 | Rd | M |
|  |  |  |  |  |  |
| Igual | 1000 | Rf1 | Rf2 | Inm | J |
| mayor | 1001 | Rf1 | Rf2 | Inm | J |
|  |  |  |  |  |  |
| Inc | 1010 | Rf1 | Rf1 | 1 | pseudoinstruccion |
| Dec | 1011 | Rf1 | Rf1 | -1 | pseudoinstruccion |
| Nop | 1100 | X0 | X0 | 0 | pseudoinstruccion |
| Mayorig | 1101 | Rf1 | Rf2 | Inm | pseudoinstruccion |

*ALU – Unidad Aritmético Lógica:*

Nuestra ALU es de 16 bits, tiene como entrada un campo de 2 bits el cual permite codificar las funciones principales.

|  |  |  |  |
| --- | --- | --- | --- |
| Instrucción | Operación | Operación  que realiza  la ALU | ALU OP |
| Tipo-R | Suma | Suma | 0000 |
| Tipo-R | Resta | Resta | 0001 |
| Tipo-R | O | O | 0010 |
| Tipo-R | Y | Y | 0011 |
| Tipo-R | Not | Not | 0100 |
| Tipo-I | Sumai | Suma con inmediato | 0101 |
| Tipo-J | Igual | Resta | 0001 |
| Tipo-J | Mayor | Resta | 0001 |
| Tipo-M | Carga | Suma | 0000 |
| Tipo-M | Almacena | Suma | 0000 |

UNIDAD DE CONTROL

Para la construcción de la unidad de control se realizo una tabla de verdad según las instrucciones y lo que hace cada una.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instrucción | AluSRC | Memtoreg | Regwrite | Memread | Memwrite | Branch | aluOP1 | AluOP2 |
| Tipo R | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| Tipo I |  |  |  |  |  |  |  |  |
| Tipo J |  |  |  |  |  |  |  |  |
| Carga |  |  |  |  |  |  |  |  |
| Almacena |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |