

Arquitectura de computadoras – 2024

Integrantes:

Roko María Guillermina DNI 45.390.594

Gamarra Bruno Elías DNI 44.772.853

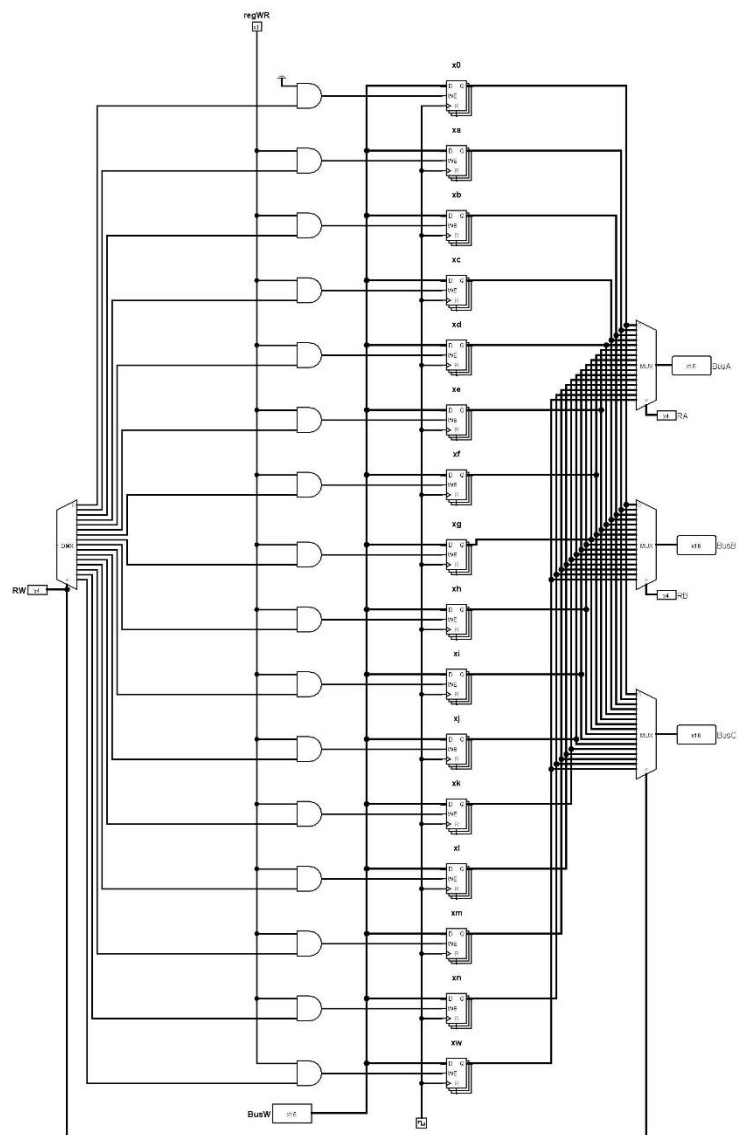
ISA procesador monociclo.

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

REGISTROS

<i>Registros de propósito general</i>	
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

Banco de registros:



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	Y rd, rf1, rf2	$R[rd] = R[rf1] \&\& R[rf2]$
0011	O 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	$R[rd] = R[rf1] + \text{SignExt}[inm]$
0111	Not rf1	$R[rf1] = !R[rf1]$
1000	Igual rf1, rf2, inm	Si $(R[rf1] == R[rf2]) \rightarrow PC = PC + \text{SignExt}[inm]$ Sino $PC = PC + 2$
1001	Mayor rf1, rf2, inm	Si $(R[rf1] > R[rf2]) \rightarrow PC = PC + \text{SignExt}[inm]$ Sino $PC = PC + 2$
1010	Mayorig rf1, rf2, inm	Si $(R[rf1] \geq R[rf2]) \rightarrow PC = PC + \text{SignExt}[inm]$ Sino Si $(R[rf1] == R[rf2]) \rightarrow PC = PC + \text{SignExt}[inm]$ Sino $PC = PC + 2$

PSEUDOINSTRUCCIONES:

Pseudoinstrucción		
Inc	Inc rf1	Sumai rf1, rf1, 1
Dec	Dec rf1	Sumai rf1, rf1, -1
nop	nop	Sumai x0, x0, 0 <- un salto

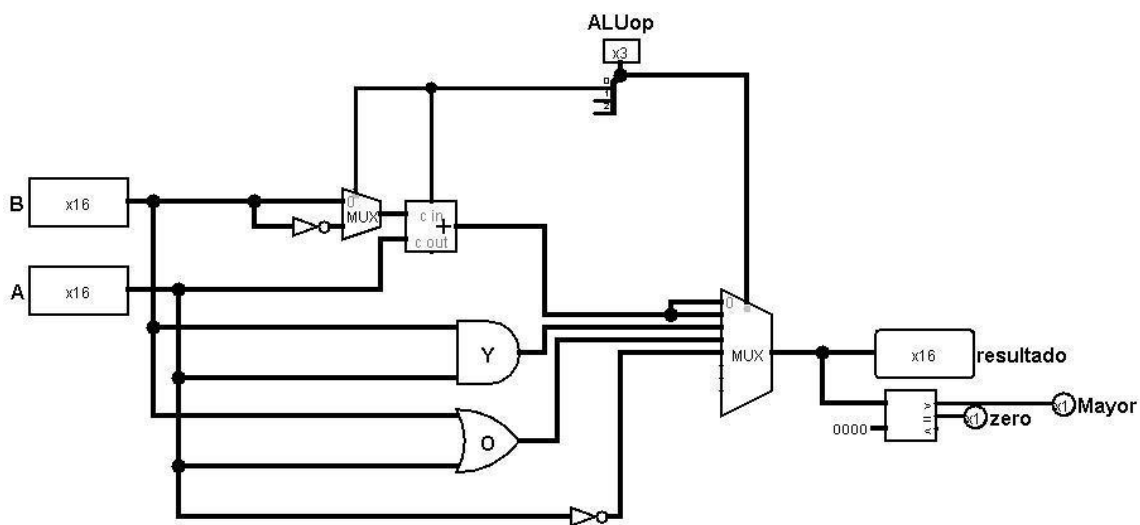
FORMATO DE LAS INSTRUCCIONES:

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
Y	0010	Rd	Rf1	Rf2	R
O	0011	Rd	Rf1	Rf2	R
Not	0111	Rf1	Rf1	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
Mayorig	1010	Rf1	Rf2	Inm	J
Inc		Rf1	Rf1	1	pseudoinstruccion
Dec		Rf1	Rf1	-1	pseudoinstruccion
Nop		Xo	Xo	0	pseudoinstruccion

ALU – Unidad Aritmético Lógica:



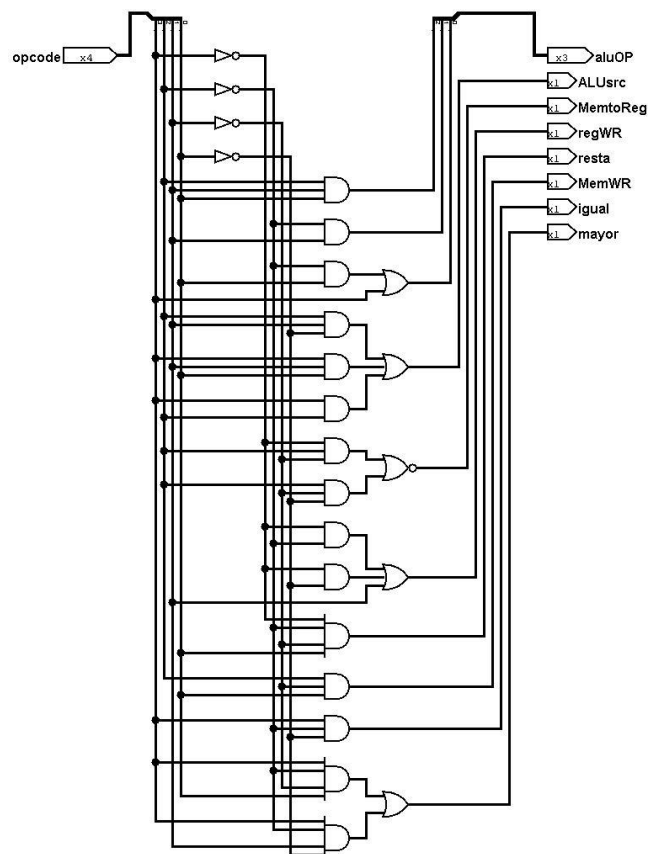
Nuestra ALU es de 16 bits, tiene como entrada un campo de 3 bits el cual permite codificar las funciones principales. (Suma, Resta, AND, OR, NOT)

Instrucción	Operación	Operación que realiza la ALU	ALU OP
Tipo-R	Suma	Suma	000
Tipo-R	Resta	Resta	001
Tipo-R	Y	Y	010
Tipo-R	O	O	011
Tipo-R	Not	Not	100
Tipo-I	Sumai	Suma con inmediato	000
Tipo-J	Igual	Resta	001
Tipo-J	Mayor	Resta	001
Tipo-J	Mayoroig	Resta	001
Tipo-M	Carga	Suma	000
Tipo-M	Almacena	Suma	000

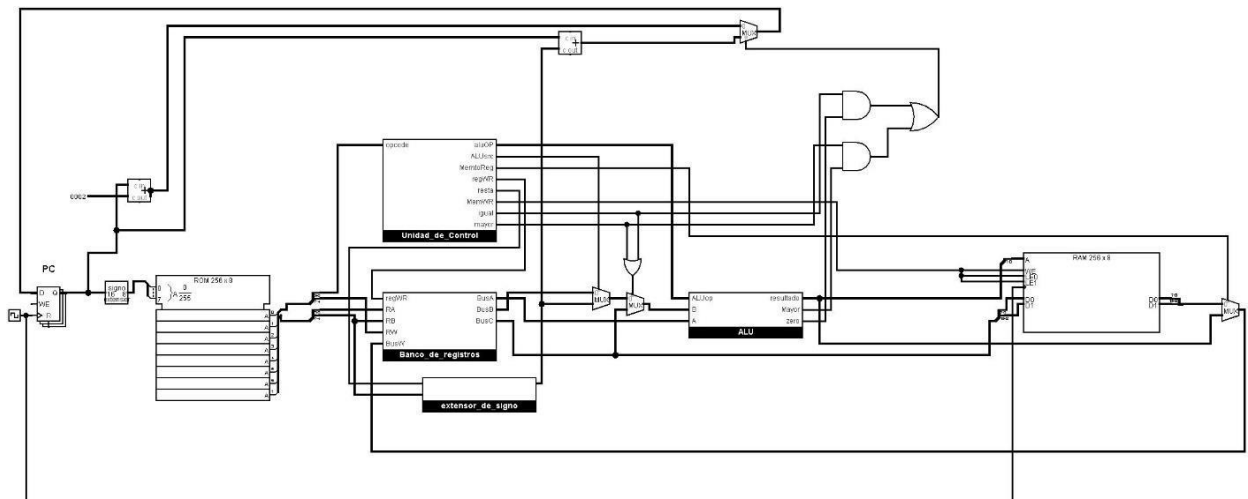
UNIDAD DE CONTROL

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

ENTRADA O SALIDA	Nombre de señal	Tipo-R					Tipo-M		Tipo-I	Tipo-J		
Entradas		Suma	Resta	Y	O	Not	Carga	Almacena	Sumai	Igual	mayor	Mayoroig
	I(bit3)	0	0	0	0	0	0	0	0	1	1	0
	I(bit2)	0	0	0	0	1	1	1	1	0	0	1
	I(bit1)	0	0	1	1	1	0	0	1	0	0	0
	I(bit0)	0	1	0	1	1	0	1	0	0	1	1
Salidas	AluSRC	0	0	0	0	0	0	0	1	1	1	1
	MemtoReg	0	0	0	0	0	1	X	0	0	0	0
	regWrite	1	1	1	1	1	1	0	1	0	0	0
	Resta	0	1	0	0	0	0	0	0	0	0	0
	memWrite	0	0	0	0	0	0	1	0	0	0	0
	Igual	0	0	0	0	0	0	0	0	1	0	1
	Mayor	0	0	0	0	0	0	0	0	0	1	1
	aluOP1	0	1	0	1	0	0	0	0	1	1	1
	aluOP2	0	0	1	1	0	0	0	0	0	0	0
	aluOP3	0	0	0	0	1	0	0	0	0	0	0



Y el camino de datos de nuestro procesador Monociclo quedaría de esta manera:



Programa requerido:

Codigo ensamblador

```
Sumai xa, xo, 8
Sumai xb, xo, 10
Carga xi, xa, xo
Carga xj, xb, xo
Igual xi, xj, 6
Almacena xb, xo, xo
Igual xi, xi, 6
```

(posición +10 del primer igual)

```
Almacena xo, xo, xa
Almacena xo, xo, xb
```

Binario correspondiente:

```
1000 0000 0001 0110
1010 0000 0010 0110
0000 0001 1001 0100
0000 0010 1010 0100
0110 1010 1001 1000
0000 0000 0010 0101
0110 1001 1001 1000
```

(salto primer condicion)

```
0001 0000 0000 0101
0010 0000 0000 0101
```