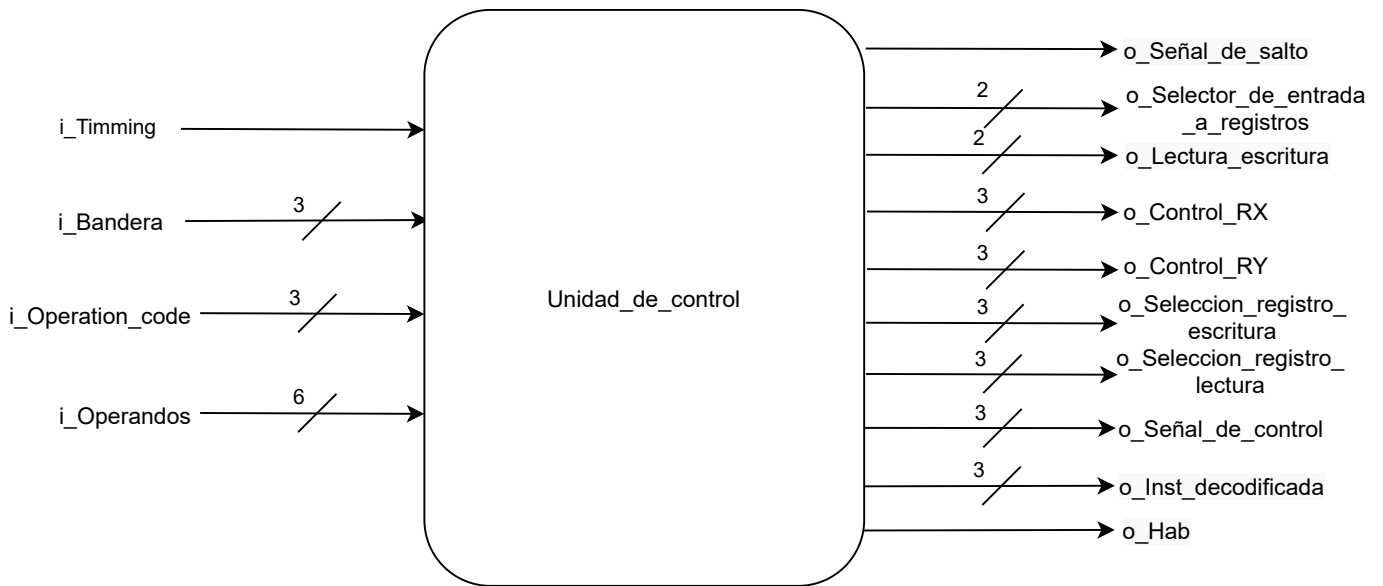


Unidad_de_control



Entradas y salidas

Señal	Núm. de Bits	Descripción
i_Timming	1 bit	Entrada de referencia en el tiempo, debe ser una señal periódica con una frecuencia se determina por el preescalador.
i_Bandera	3 bits	Entrada que contiene las banderas de estados sobre las operaciones en la unidad lógico aritmética.
i_Operation_code	3 bits	Entrada que contiene la parte de la instrucción (Sin los operandos) correspondiente al set de instrucciones.
i_Operandos	6 bits	Entrada que contiene la dirección de los datos a operar.
o_Señal_de_salto	1 bit	Señal de control dirigida al PC counter la cual le indicar cuándo realizar un salto condicional.
o_Selector_de_entrada_a_registros	2 bits	Salida que selecciona el dato que ingresaran a los registros de datos en el banco de registro en cada ciclo de operación. Esta va al MUX_1.
o_Lectura_escritura	2 bits	Señal de control que se encarga de indicar si se escribe o se lee un determinado dato en los registros de datos o existe un movimiento de registros.
o_Control_RX	3 bits	Se encarga de determinar qué registro se pasará como operando 1 ya sea a la ALU o al manager de salidas a memoria.
o_Control_RY	3 bits	Se encarga de determinar qué registro se pasará como operando 2 ya sea a la ALU o al manager de salidas a memoria.
o_Seleccion_registro_escritura	3 bits	Señal de gestión del movimiento de registros.
o_Seleccion_registro_lectura	3 bits	Señal de gestión del movimiento de registros.
o_Señal_de_control	3 bits	Señal de control dirigida al Manager_de_salidas_a_memoria y e indica qué datos saldrán del microprocesador a los buses de direcciones y datos de salidas a memoria.
o_Inst_decodificada	3 bits	Salida que le indica a la unidad lógico aritmética qué operación matemática o lógica realizará.

o_Hab	1 bit	Habilita a la ALU para realizar una operación.
-------	-------	--

Descripción funcional

Decodifica las instrucciones que llegan desde el registro de instrucciones y actúa conforme al set de instrucciones establecido y coordina las operaciones en la unidad lógica aritmética, también analiza el estado de las banderas y conforme a este establece un salto o no en el PC counter o envía una señal para que se pase algún valor almacenado en el registro al bus de datos para su posterior almacenamiento en memoria

°Pseudocódigo

Algoritmo Unidad_de_Control

//Entradas y salidas

Definir i_Timming como entero

Definir i_Rst como entero

Dimension i_Bandera[3]

Dimension i_Operation_code[3]

Dimension i_Operandos[6]

Definir o_Señal_de_salto como entero

Dimension o_Selector_de_entrada_a_registros[2]

Dimension o_Lectura_escritura[2]

Dimension o_Control_RX[3]

Dimension o_Control_RY[3]

Dimension o_Seleccion_registro_escritura[3]

Dimension o_Seleccion_registro_lectura[3]

Dimension o_Señal_de_control[3]

Dimension o_Inst_decodificada[3]

Definir o_Hab como entero

//Señales internas

Definir Señal_de_salto como entero

Dimension Selector_de_entrada_a_registros[2]

Dimension Lectura_escritura[2]

Dimension Control_RX[3]

Dimension Control_RY[3]

Dimension Seleccion_registro_escritura[3]

Dimension Seleccion_registro_lectura[3]

Dimension Señal_de_control[3]

Dimension Inst_decodificada[3]

Definir Hab como entero

//Bloque always

Mientras i_Timming=1 o i_Timming=0 Hacer

Si i_Timming=1 Entonces

Si i_Rst=1 Entonces

cont<-0

Fin Si

Si i_Rst=0 Entonces

Segun i_Operation_code Hacer

000:

//LOAD_1

Señal_de_salto<-0

Selector_de_entrada_a_registros[]<-[01]

Lectura_escritura[]<-[01]

Control_RX[]<-[000]

Control_RY[]<-[000]

Seleccion_registro_escritura[]<-i_Operandos[5:3]

Seleccion_registro_lectura[]<-[000]

Señal_de_control[]<-[000]

Inst_decodificada[]<-[000]

Hab<-0

001:

//LOAD_2

```

//LOAD_2
Señal_de_salto<-0
Selector_de_entrada_a_registros[]<-[10]
Lectura_escritura[]<-[11]
Control_RX[]<-[000]
Control_RY[]<-i_Operandos[2:0]
Seleccion_registro_escritura[]<-i_Operandos[5:3]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[100]
Inst_decodificada[]<-[000]
Hab<-0
010:
//STORE_1
Señal_de_salto<-0
Selector_de_entrada_a_registros[]<-[00]
Lectura_escritura[]<-[00]
Control_RX[]<-i_Operandos[2:0]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[000]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[111]
Inst_decodificada[]<-[000]
Hab<-0
011:
//STORE_2
Señal_de_salto<-0
Selector_de_entrada_a_registros[]<-[00]
Lectura_escritura[]<-[00]
Control_RX[]<-i_Operandos[5:3]
Control_RY[]<-i_Operandos[2:0]
Seleccion_registro_escritura[]<-[000]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[110]
Inst_decodificada[]<-[000]
Hab<-0
100:
//MOVE
Señal_de_salto<-0
Selector_de_entrada_a_registros[]<-[00]
Lectura_escritura[]<-[10]
Control_RX[]<-[000]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-i_Operandos[5:3]
Seleccion_registro_lectura[]<-i_Operandos[2:0]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
101:
//MATH
Señal_de_salto<-0
Selector_de_entrada_a_registros[]<-[00]
Lectura_escritura[]<-[11]
Control_RX[]<-i_Operandos[5:3]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[000]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-i_Operandos[2:0]
Hab<-1
110:
//Jump
Segun i_Operandos[2:0] Hacer
000:
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[00]
Lectura_escritura[]<-[00]
Control_RX[]<-[000]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[000]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
001:
//JMP
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[00]
Lectura_escritura[]<-[00]
Control_RX[]<-[000]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[000]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
010:
//JMP
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[00]
Lectura_escritura[]<-[00]
Control_RX[]<-[000]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[000]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
011:
//JMP
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[00]
Lectura_escritura[]<-[00]
Control_RX[]<-[000]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[000]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
100:
//JMP
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[00]
Lectura_escritura[]<-[00]
Control_RX[]<-[000]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[000]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
101:
//JMP
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[00]
Lectura_escritura[]<-[00]
Control_RX[]<-[000]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[000]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
110:
//JMP
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[00]
Lectura_escritura[]<-[00]
Control_RX[]<-[000]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[000]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
111:
//JMP
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[00]
Lectura_escritura[]<-[00]
Control_RX[]<-[000]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[000]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0

```

```

Lectura_escritura[]<-[00]
Control_RX[]<-i_Operandos[5:3]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[000]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
001:
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[11]
Lectura_escritura[]<-[11]
Control_RX[]<-i_Operandos[5:3]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[111]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
010:
Si bandera[0]=1 Entonces
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[11]
Lectura_escritura[]<-[11]
Control_RX[]<-i_Operandos[5:3]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[111]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
Fin Si
011:
Si bandera[0]=0 Entonces
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[11]
Lectura_escritura[]<-[11]
Control_RX[]<-i_Operandos[5:3]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[111]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
Fin Si
100:
Si bandera[1]=1 Entonces
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[11]
Lectura_escritura[]<-[11]
Control_RX[]<-i_Operandos[5:3]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[111]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
Fin Si
101:
Si bandera[1]=0 Entonces
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[11]
Lectura_escritura[]<-[11]
Control_RX[]<-i_Operandos[5:3]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[111]

```

```
Seleccion_registro_escritura[]<-[111]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
Fin Si
110:
Si bandera[2]=1 Entonces
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[11]
Lectura_escritura[]<-[11]
Control_RX[]<-i_Operandos[5:3]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[111]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
Fin Si
De Otro Modo:
Si bandera[2]=0 Entonces
Señal_de_salto<-1
Selector_de_entrada_a_registros[]<-[11]
Lectura_escritura[]<-[11]
Control_RX[]<-i_Operandos[5:3]
Control_RY[]<-[000]
Seleccion_registro_escritura[]<-[111]
Seleccion_registro_lectura[]<-[000]
Señal_de_control[]<-[000]
Inst_decodificada[]<-[000]
Hab<-0
Fin Si
Fin Segun
De Otro Modo:
Fin Segun
FinSi
FinSi
Fin Mientras
FinAlgoritmo
```