

PRÁCTICA 2

PROYECTO LIBRE CON CPU MONOCICLO

Eduardo Borges y Antonio Sanjuan Prieto.

10/06/17

Informe de la Entrega Parcial 2 de la Práctica 2 - CPU Monociclo con Entrada-Salida.

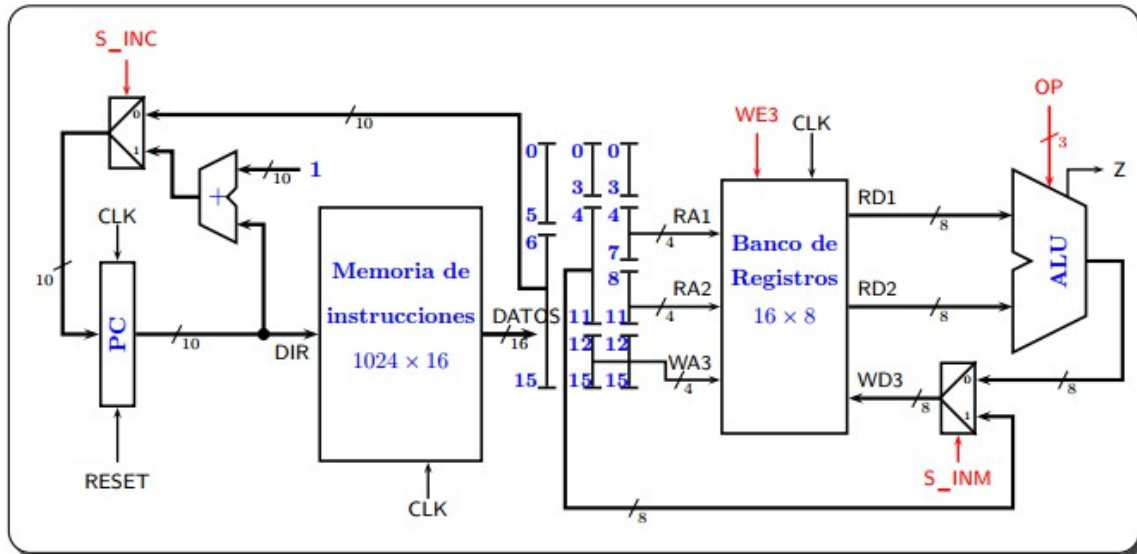
Índice de Contenidos

1. Introducción.....	3
2. Objetivos.....	3
3. Codificación elegida para cada una de las instrucciones.....	4
3.1. Aritmético - Lógicas.....	4
3.2. Carga de un inmediato.....	4
3.3. Salto incondicional Absoluto.....	5
3.4. Salto Condicional.....	5
3.5. Entrada de un puerto a registro.....	5
3.6. Salida de un registro a un puerto.....	5
3.7. Salida de un inmediato a un puerto.....	6
3.8. Salto incondicional Relativo.....	6
3.9. Salto a subrutinas.....	6
3.9. Vuelta de subrutinas.....	6
3.10. Método de codificación.....	7
4. Modificaciones en el camino de datos.....	9
4.1. CPU Básica.....	9
4.2. CPU con entrada/salida.....	9
4.3. CPU con Salto relativo.....	10
4.4. CPU con salto a subrutinas.....	10
4.5. Camino de datos final.....	11
5. Instrucciones cuyo funcionamiento ha sido verificado.....	12
5.1. Comprobación sin Entrada/Salida.....	12
5.2. Capturas GTKWAVE.....	13

INFORME DE LA PRÁCTICA 2 – ENTREGA PARCIAL CPU. PROYECTO LIBRE: CPU MONOCICLO CON E/S.

1. Introducción.

En esta segunda práctica debemos modificar la unidad de control y el camino de datos que diseñamos. Representado en la siguiente figura:



En esta entrega realizaremos los cambios que hemos estudiado en la práctica anterior. Modificaremos camino de datos y unidad de control para poder realizar las nuevas instrucciones y cumplir con los objetivos de esta práctica. Se comprobará el resultado mediante un programa.

2. Objetivos.

Los objetivos de esta entrega parcial podemos resumirlos en los siguientes:

- Nuevo Conjunto de Instrucciones con una correcta asignación de codificación.
- Camino de datos y unidad de control definidos sobre papel para soportar Entrada / Salida, saltos a subrutinas y saltos relativos.
- Reprogramación de unidad de control y camino de datos para soportar dichas instrucciones.
- Definir un nuevo programa para comprobar el funcionamiento de las nuevas instrucciones.

**INFORME DE LA PRÁCTICA 2 – ENTREGA PARCIAL CPU.
PROYECTO LIBRE: CPU MONOCICLO CON E/S.**

3. Codificación elegida para cada una de las instrucciones.

Hemos usado un bus de datos de 16 bits, es decir, las instrucciones ocuparán 16 bits. Los opcodes de las instrucciones varían de una a otra. Diseñamos un repertorio de instrucciones con cuatro tipos de instrucción:

3.1. Aritmético - Lógicas.

El formato de instrucción de estas operaciones es:

Registro destino	Registro 2	Registro 1	Opcode
------------------	------------	------------	--------

Donde cada registro ocupa 4 bits (dirección) y el opcode otros 4 bits. En las instrucciones aritmético-lógicas podemos encontrar 8 instrucciones diferentes:

- Asignación con el opcode 0000
- Negación lógica (NOT) con el opcode 0001
- Suma con el opcode 0010
- Resta con el opcode 0011
- Operación lógica (AND) con el opcode 0100
- Operación lógica (OR) con el opcode 0101
- Cambio de signo del operando 2 (Registro 2) con el opcode 0110
- Cambio de signo del operando 1 (Registro 1) con el opcode 0111

3.2. Carga de un inmediato.

El formato de instrucción de estas operaciones es:

Registro	Inmediato	Opcode
----------	-----------	--------

Donde el registro ocupa 4 bits (dirección), el valor inmediato 8 bits y el opcode otros 4 bits.

Sólo es una instrucción por tanto usa solo un opcode: 1000

**INFORME DE LA PRÁCTICA 2 – ENTREGA PARCIAL CPU.
PROYECTO LIBRE: CPU MONOCICLO CON E/S.**

3.3. Salto incondicional Absoluto.

El formato de instrucción de estas operaciones es:

Dirección absoluta	Opcode
--------------------	--------

Donde la dirección ocupa 10 bits y el opcode otros 6 bits.

Sólo es una instrucción por tanto usa solo un opcode: 001001

3.4. Salto Condicional.

El formato de instrucción de estas operaciones es:

Dirección para el salto	Opcode
-------------------------	--------

Donde la dirección ocupa 10 bits y el opcode otros 6 bits.

En este tipo hay dos instrucciones:

- Condicional si z=0 con opcode 011001

- Condicional si z=1 con opcode 101001

3.5. Entrada de un puerto a registro.

El formato de instrucción de estas operaciones es:

Libre	Registro	Puerto	Opcode
-------	----------	--------	--------

Donde el registro ocupa 4 bits (dir), el puerto 2 bits y el opcode 6 bits. Dejando 2 bits libres.

Sólo es una instrucción por tanto usa solo un opcode: 111010

3.6. Salida de un registro a un puerto.

El formato de instrucción de estas operaciones es:

Libre	Registro	Puerto	Opcode
-------	----------	--------	--------

Donde el registro ocupa 4 bits (dir), el puerto 2 bits y el opcode 6 bits. Dejando 2 bits libres.

Sólo es una instrucción por tanto usa solo un opcode: 011010

**INFORME DE LA PRÁCTICA 2 – ENTREGA PARCIAL CPU.
PROYECTO LIBRE: CPU MONOCICLO CON E/S.**

3.7. Salida de un inmediato a un puerto.

El formato de instrucción de estas operaciones es:

Inmediato	Puerto	Opcode
-----------	--------	--------

Donde el inmediato ocupa 8 bits, el puerto 2 bits y el opcode 6 bits.

Sólo es una instrucción por tanto usa solo un opcode: **101010**

3.8. Salto a subrutinas.

El formato de instrucción de estas operaciones es:

Dirección	Opcode
-----------	--------

Donde la dirección ocupa 10 bits y el opcode otros 6 bits.

Sólo es una instrucción por tanto usa solo un opcode: **111001**

3.9. Vuelta de subrutinas.

El formato de instrucción de estas operaciones es:

Libree	Opcode
--------	--------

Donde el opcode ocupa 6 bits.

Sólo es una instrucción por tanto usa solo un opcode: **001010**

**INFORME DE LA PRÁCTICA 2 – ENTREGA PARCIAL CPU.
PROYECTO LIBRE: CPU MONOCICLO CON E/S.**

3.10. Método de codificación.

Hemos ordenado los opcodes según los 4 últimos bits de éste. En la siguiente tabla recogeremos todas las combinaciones que forman los opcodes.

Leyenda:

Instrucciones Aritmético-Lógicas.	Instrucciones de Carga.
Instrucciones de Entrada/Salida.	NO OPERACIÓN.
Instrucciones de Salto incondicional.	Instrucciones de Salto condicional.
Instrucciones de Salto a subrutinas	

Tabla:

0	0	0	0	0	0	ALU: Asignación.
0	0	0	0	0	1	ALU: NOT.
0	0	0	0	1	0	ALU: Suma.
0	0	0	0	1	1	ALU: Resta.
0	0	0	1	0	0	ALU: AND.
0	0	0	1	0	1	ALU: OR.
0	0	0	1	1	0	ALU: Cambio Signo 1.
0	0	0	1	1	1	ALU: Cambio Signo 2.
0	0	1	0	0	0	Carga de un valor inmediato.
0	0	1	0	0	1	Salto Incondicional
0	0	1	0	1	0	Instrucciones de Salto a subrutinas
0	0	1	0	1	1	
0	0	1	1	0	0	
0	0	1	1	0	1	
0	0	1	1	1	0	
0	0	1	1	1	1	
0	1	0	0	0	0	ALU: Asignación.
0	1	0	0	0	1	ALU: NOT.
0	1	0	0	1	0	ALU: Suma.
0	1	0	0	1	1	ALU: Resta.
0	1	0	1	0	0	ALU: AND.
0	1	0	1	0	1	ALU: OR.
0	1	0	1	1	0	ALU: Cambio Signo 1.
0	1	0	1	1	1	ALU: Cambio Signo 2.
0	1	1	0	0	0	Carga de un valor inmediato.
0	1	1	0	0	1	Salto condicional (Z=0)
0	1	1	0	1	0	E/S de registro (SALIDA)
0	1	1	0	1	1	
0	1	1	1	0	0	
0	1	1	1	0	1	
0	1	1	1	1	0	
0	1	1	1	1	1	

**INFORME DE LA PRÁCTICA 2 – ENTREGA PARCIAL CPU.
PROYECTO LIBRE: CPU MONOCICLO CON E/S.**

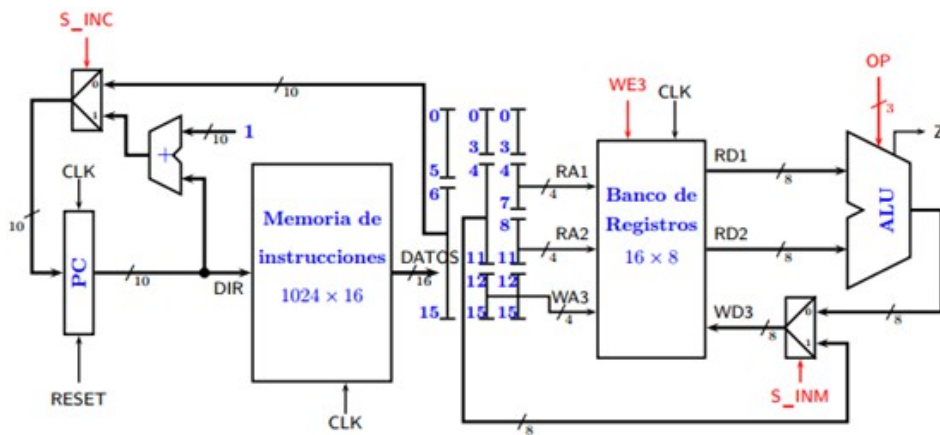
1	0	0	0	0	0	ALU: Asignación.
1	0	0	0	0	1	ALU: NOT.
1	0	0	0	1	0	ALU: Suma.
1	0	0	0	1	1	ALU: Resta.
1	0	0	1	0	0	ALU: AND.
1	0	0	1	0	1	ALU: OR.
1	0	0	1	1	0	ALU: Cambio Signo 1.
1	0	0	1	1	1	ALU: Cambio Signo 2.
1	0	1	0	0	0	Carga de un valor inmediato.
1	0	1	0	0	1	Salto condicional (Z=1)
1	0	1	0	1	0	E/S de inmediato (SALIDA)
1	0	1	0	1	1	
1	0	1	1	0	0	
1	0	1	1	0	1	
1	0	1	1	1	0	
1	0	1	1	1	1	
1	1	0	0	0	0	ALU: Asignación.
1	1	0	0	0	1	ALU: NOT.
1	1	0	0	1	0	ALU: Suma.
1	1	0	0	1	1	ALU: Resta.
1	1	0	1	0	0	ALU: AND.
1	1	0	1	0	1	ALU: OR.
1	1	0	1	1	0	ALU: Cambio Signo 1.
1	1	0	1	1	1	ALU: Cambio Signo 2.
1	1	1	0	0	0	Carga de un valor inmediato.
1	1	1	0	0	1	Instrucciones de Salto a subrutinas
1	1	1	0	1	0	E/S (Entrada)
1	1	1	0	1	1	
1	1	1	1	0	0	
1	1	1	1	0	1	
1	1	1	1	1	0	
1	1	1	1	1	1	NO OPERACIÓN.

INFORME DE LA PRÁCTICA 2 – ENTREGA PARCIAL CPU.
PROYECTO LIBRE: CPU MONOCICLO CON E/S.

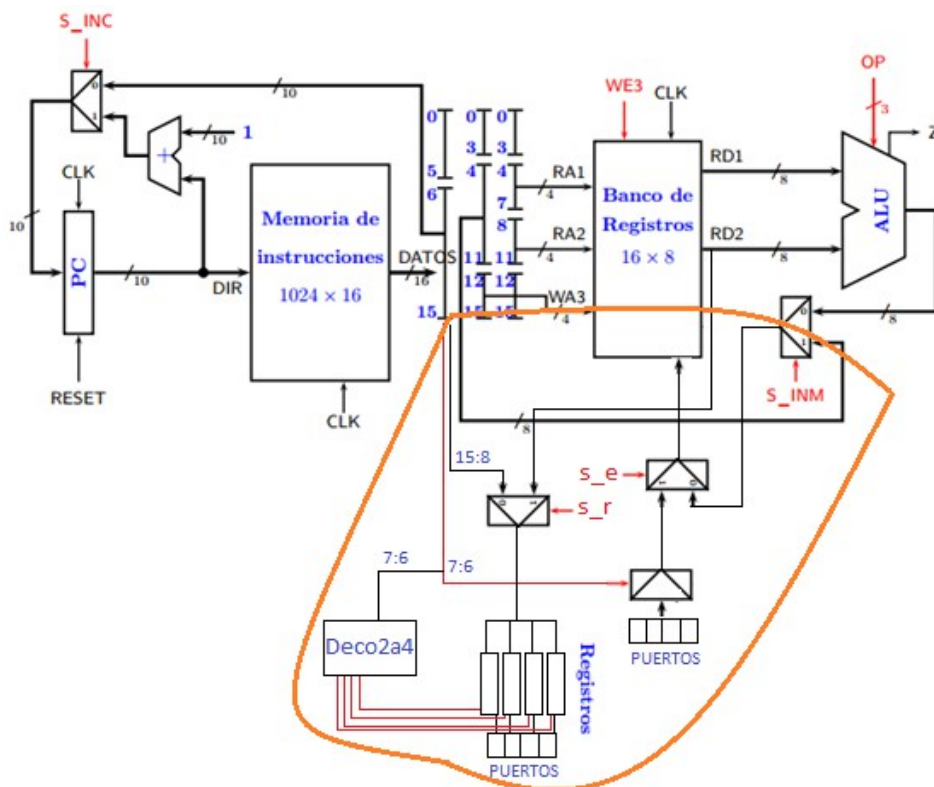
4. Modificaciones en el camino de datos.

Para añadir las nuevas instrucciones a nuestro repertorio, hemos tenido que modificar el camino de datos partiendo del de nuestra cpu monociclo básica.

4.1. CPU Básica.

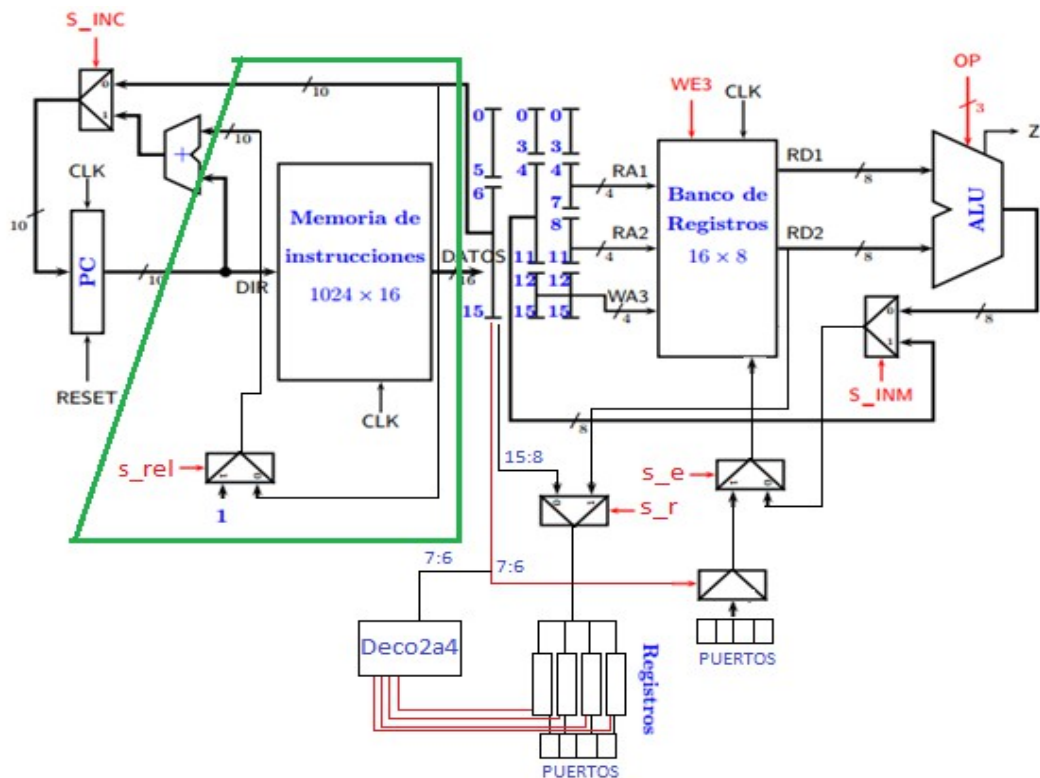


4.2. CPU con entrada/salida.

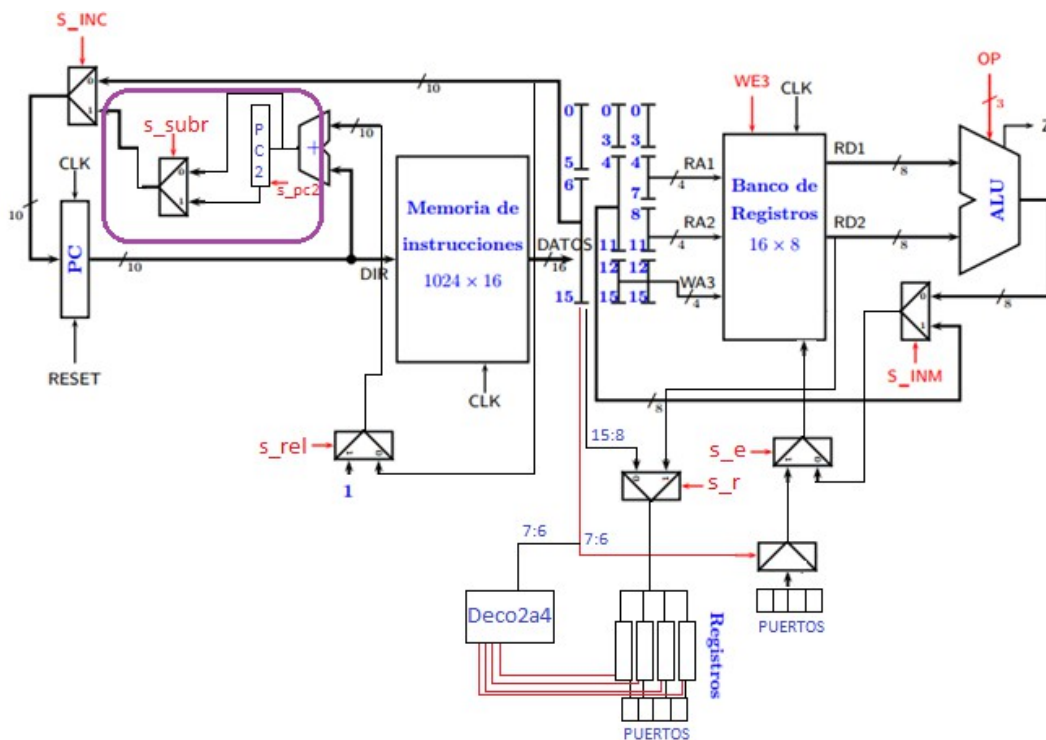


PROYECTO LIBRE: CPU MONOCICLO CON E/S.

4.3. CPU con Salto relativo.

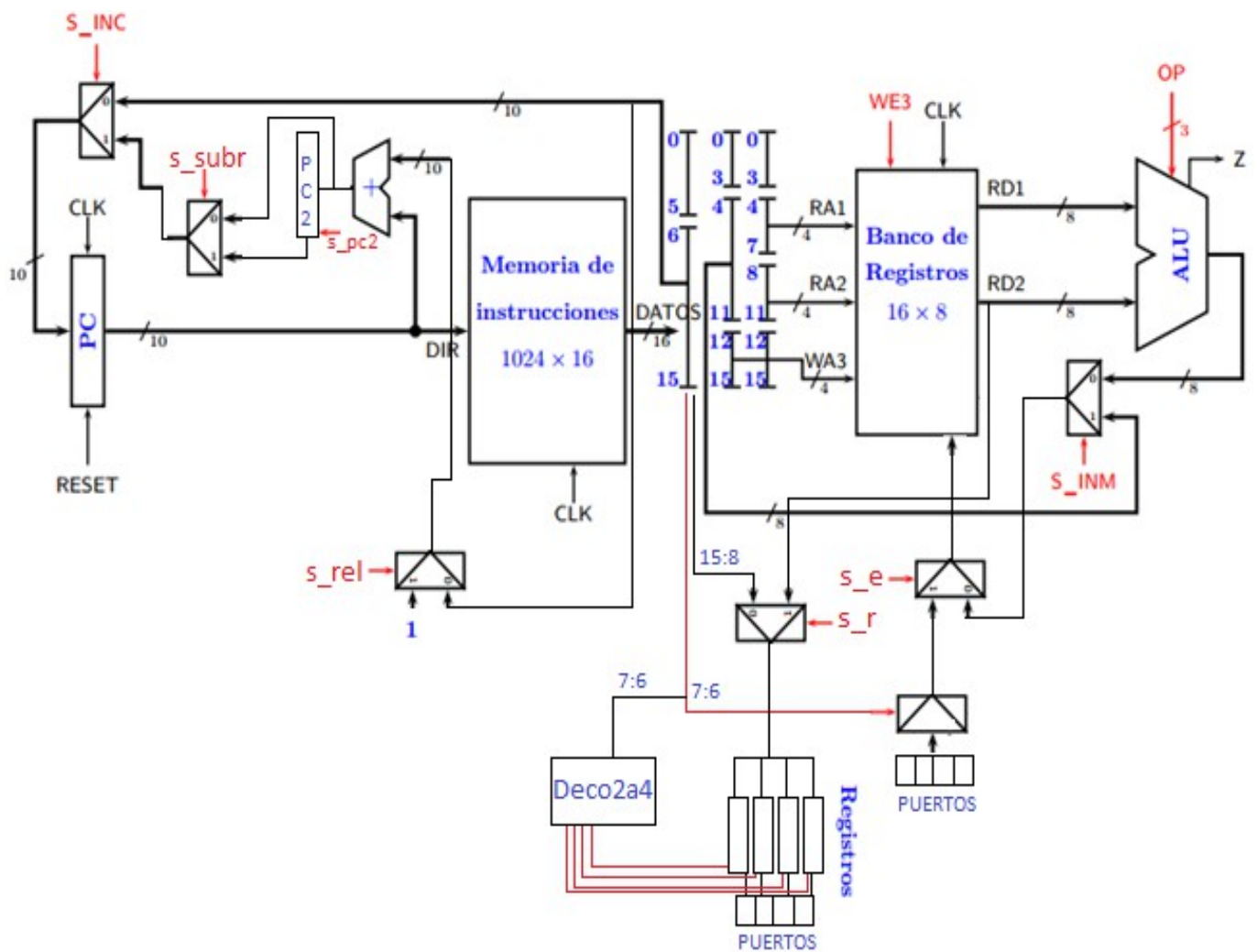


4.4. CPU con salto a subrutinas.



INFORME DE LA PRÁCTICA 2 – ENTREGA PARCIAL CPU.
PROYECTO LIBRE: CPU MONOCICLO CON E/S.

4.5. Camino de datos final.



5. Instrucciones cuyo funcionamiento ha sido verificado.

5.1. Comprobación sin Entrada/Salida.

El programa que hemos utilizado para verificar las instrucciones se encuentra en el fichero “progfile.dat”, y contiene el siguiente programa:

1. 0001_0000_0001_1000 E0: LI \$1, 1

2. 0010_0000_0000_1000 LI \$2, 0

3. 0011_0010_0001_0011 SUB \$3, \$1,\$2

4. 0000_0001_0110_1001 BZ E1

5. 0011_0000_0010_0010 ADD \$3, \$0, \$2

6. 0000_0001_1110_1001 E1: BZ E2

7. 1111_1111_1111_1111 NOP

8. 0000_0010_0101_1001 E2: BNZ E3

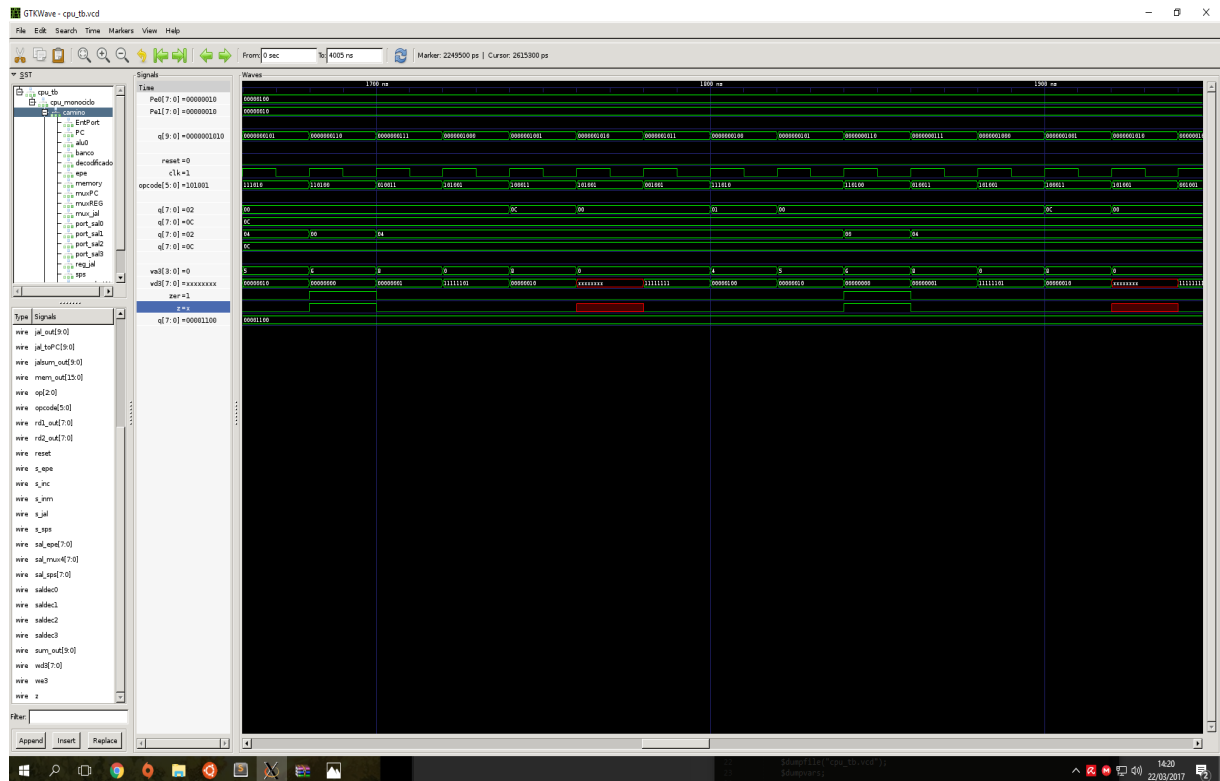
9. 0011_0001_0010_0010 ADD \$3, \$1, \$2

10. 0000_0010_1101_1001 E3: BNZ E4

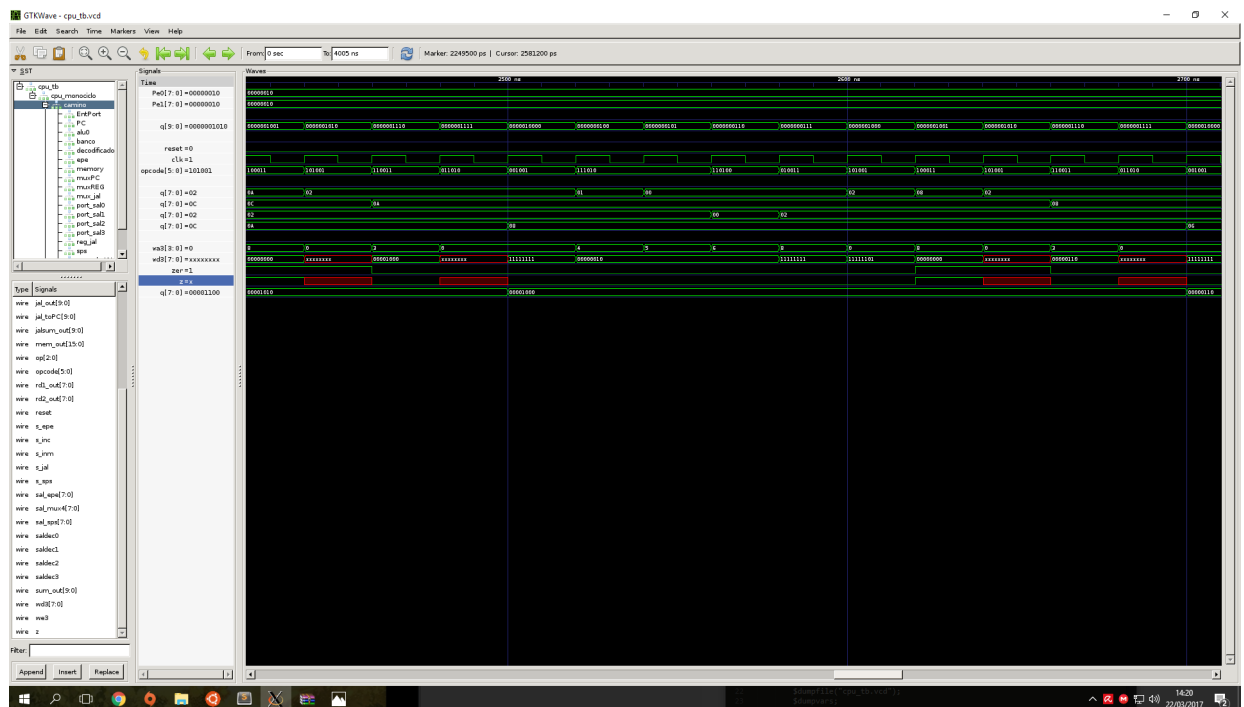
11. 1111_1111_1111_1111 NOP

12. 0000_0000_0000_1001 JE0

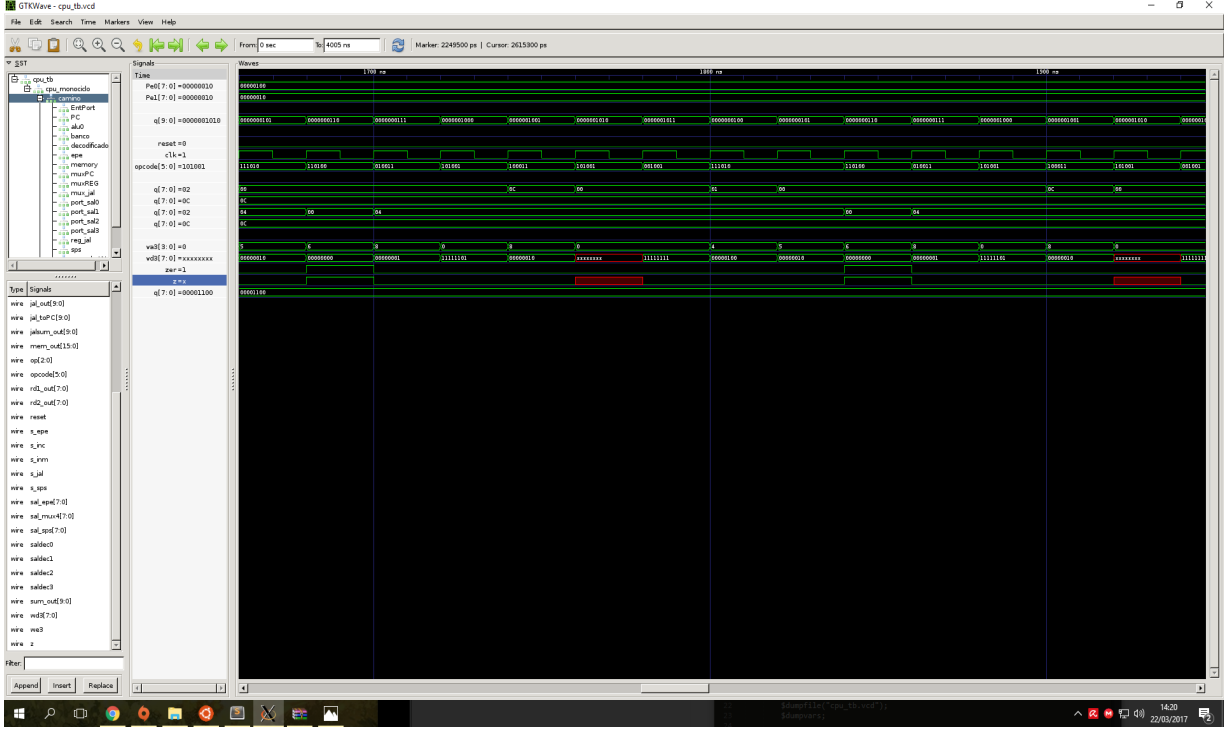
5.2. Capturas GTKWAVE.



Suma

**Resta**

**INFORME DE LA PRÁCTICA 2 – ENTREGA PARCIAL CPU.
PROYECTO LIBRE: CPU MONOCICLO CON E/S.**



Nada (A la espera de operacion)