

# **Informe de prácticas**

## **CPU monociclo**

Carlos Gregorio Martín Pérez  
Georg Sperl  
Sara Báez  
Leonor Priego García

### Codificación elegida para cada una de las instrucciones.

Teniendo en cuenta el número de operandos que se recibe en cada instrucción y el tamaño del bus de datos que hemos empleado (16b) hemos intentado hacer una codificación sencilla ya que solo usaremos 4 tipos de instrucciones diferentes. En prácticas posteriores debemos modificar los opcodes para conseguir un uso más eficiente, que permita el aprovechamiento de los bit opcode.

Los cuatro tipos diferentes de instrucciones que usaremos son los siguientes:

- ALU

Registro destino (4b)	Registro OperandoA (4b)	Registro OperandoB (4b)	Opcode (4b)
-----------------------	-------------------------	-------------------------	-------------

Para las operaciones aritmético-lógicas usaremos un prefijo común → 0xxx. Para cada una de las operaciones específicas emplearemos los 3 bits menos significativos (los que están como 'x')

Asignación =op. A:	0000
NOT:	0001
Suma:	0010
Resta:	0011
AND:	0100
OR:	0101
Cambio de signo op. A:	0110
Cambio de signo op. B:	0111

- CARGA

Registro (4b)	Valor constante (8b)	Opcode (4b)
---------------	----------------------	-------------

La instrucción cargará en la dirección de *registro* introducida el valor constante. El opcode para la instrucción de carga → 1000

- SALTO ABSOLUTO

Dirección de salto (10b)	Opcode (6b)
--------------------------	-------------

La instrucción moverá el *program counter* (PC) a la *Dirección de salto*. El opcode que se ha elegido es → 101001

- SALTO CONDICIONAL

Dirección de salto (10b)	Opcode (6b)
--------------------------	-------------

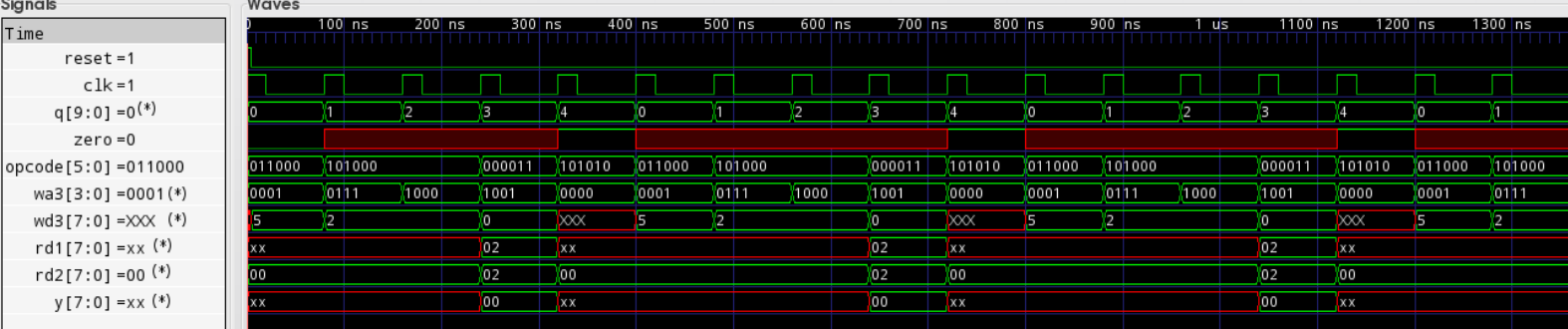
La instrucción moverá el *program counter* (PC) a la *Dirección de salto* siempre y cuando se cumpla una condición, es decir, que el *flag* de control Z esté activado(1). El otro caso, la ejecución del programa se desarrollará de forma secuencial, es decir, PC+1. El opcode que se ha elegido es → 101010

### Instrucciones cuyo funcionamiento ha sido verificado.

En las imágenes del *GTKWAVE* las señales marcadas con (\*) equivalen a las siguientes:

q = PC  
wa3 = registro  
wd3 = valor  
rd1 = operandoA  
rd2 = operandoB  
y = resultadoALU

0001_0000_0101_1000	#Cargamos 5 en Reg.1
0111_0000_0010_1000	#Cargamos 2 en Reg.7
1000_0000_0010_1000	#Cargamos 2 en Reg.8
1001_0111_1000_0011	#Restamos Reg 7 y 8 en Reg.9
0000_0000_0010_1010	#Saltamos (cond.) a la pos 0 de memoria
0001_0000_0010_1000	#Cargamos 2 en Reg.1
0010_0000_0011_1000	#Cargamos 3 en Reg.2



**Tipo de unidad de control implementada.**

Unidad de control combinacional(cableada)

**Dificultades encontradas para la realización.**

- Tiempos de sincronización
- Codificación de las instrucciones sobre todo las instrucciones de *ALU* para que no hubiera solapamiento con el uso de otras.
- Conflicto en la propagación de bits *Don't Care*
- Uso de *GTKWAVE* para depurar el CPU monociclo