

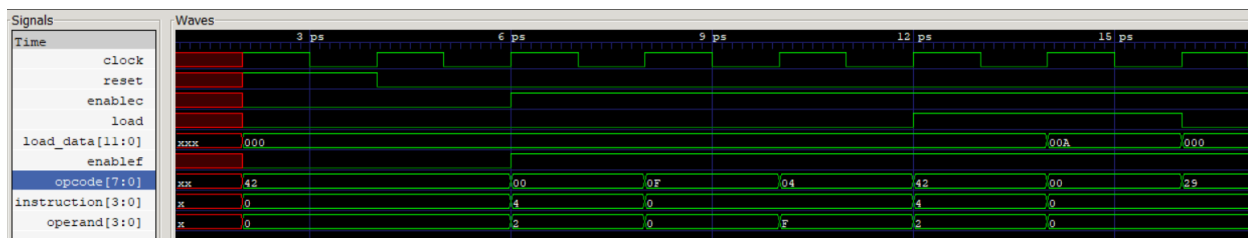
## Laboratorio 10

## Ejercicio 1:

En este ejercicio se realizó la parte del microprocesador en la que el program counter comienza a contar y se le puede colocar un valor inicial con el load\_data. El valor del program counter es la variable que sirve como dirección en la memoria ROM, la cual contiene las instrucciones u opcode para el microprocesador. Finalmente, el Fetch separa el opcode entre instrucción y operando. Tanto el program counter como el Fetch tienen bits de enable.

Programa								
clock	reset	enablec	enablef	load	load_data	opcode	instruction	operand
1	1	0	0	0	000000000000	01000010	0000	0000
0	1	0	0	0	000000000000	01000010	0000	0000
1	0	0	0	0	000000000000	01000010	0000	0000
0	0	0	0	0	000000000000	01000010	0000	0000
1	0	1	1	0	000000000000	00000000	0100	0010
0	0	1	1	0	000000000000	00000000	0100	0010
1	0	1	1	0	000000000000	00001111	0000	0000
0	0	1	1	0	000000000000	00001111	0000	0000
1	0	1	1	0	000000000000	00000100	0000	1111
0	0	1	1	0	000000000000	00000100	0000	1111
1	0	1	1	1	000000000000	01000010	0100	0010
0	0	1	1	1	000000000000	01000010	0100	0010
1	0	1	1	1	000000001010	00000000	0000	0000
0	0	1	1	1	000000001010	00000000	0000	0000
1	0	1	1	0	000000000000	00101001	0000	0000
0	0	1	1	0	000000000000	00101001	0000	0000
1	0	1	1	0	000000000000	11010101	0010	1001

gtkwave Ejercicio1\_tb.vcd Ejercicio1\_tb.gtkw



## Ejercicio 2:

En este ejercicio se implementó una ALU y un acumulador con un par de buffers tri-estado. El primer buffer permite el ingreso de los datos al bus de data derecha a la ALU; este bus de datos vendría del operando que sale del Fetch. El segundo buffer permite la salida del resultado dado por la ALU. La ALU opera el valor del registro Accu o W con el bus de datos y el resultado sale al buffer y se guarda nuevamente en el registro W. La ALU tiene dos banderas para los resultados, las banderas Zero y Carry; la primera indica si alguna de las operaciones da 0 y la segunda indica si existe algún overflow.

Programa							in	flags	out
clock	reset	enablea	enablebuff1	enablebuff2	selector				
1	1	0	0	0	000		0000	01	zzzz
0	1	0	0	0	000		0000	01	zzzz
1	0	0	0	0	000		0000	01	zzzz
0	0	0	0	0	000		0000	01	zzzz
1	0	1	1	1	000		1010	01	0000
0	0	1	1	1	000		1010	01	0000
1	0	1	1	1	001		0100	10	1100
0	0	1	1	1	001		0100	10	1100
1	0	1	1	1	010		0001	00	0001
0	0	1	1	1	010		0001	00	0001
1	0	1	1	1	011		0110	00	0111
0	0	1	1	1	011		0110	00	0111
1	0	1	1	1	100		0111	10	1000
0	0	1	1	1	100		0111	10	1000
1	0	1	1	1	100		0111	10	1111

gtkwave code\_tb.vcd code\_tb.gtkw

