

Ejercicio No.1:

Ejercicio No.2:

```
vvp Ejercicio_2_tb.out
-----Memoria Rom-----
Localidad |      Lectura
VCD info: dumpfile Ejercicio_1_tb.vcd opened for output.
000000000000 | 00001111
000000000001 | 11110000
000000000010 | xxxxxxxx
000000000011 | xxxxxxxx
000000000100 | xxxxxxxx
000000000101 | 10101010
000000000110 | 01010101
000000000111 | 11001100
000000001000 | 00110011
000000001001 | 10011001
000000001010 | 01100110
000000001011 | 11111111
000000001100 | 00000000
000000001101 | xxxxxxxx
```

0000	1111
1111	0000
@5	
1010	1010
0101	0101
1100	1100
0011	0011
1001	1001
0110	0110
1111	1111
0000	0000

En este ejercicio se implementó una memoria rom de 4096 filas y 8 bits. Se escribió un archivo de 10 filas que contienen 8 bits de información. En el módulo de la memoria rom, se leyó el archivo creado y se grabó la información en la matriz previamente definida.

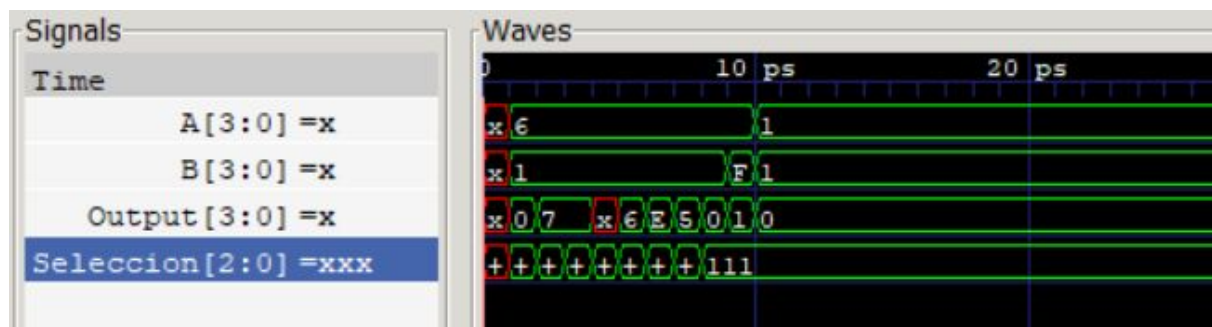
Ejercicio No.3:

```

ALU de 4 bits
Selección | A | B | Salida
-----|----|-----|-----
VCD info: dumpfile Ejercicio_3_tb.vcd opened for output.
xxx      xxxx  xxxx  xxxx
000      0110  0001  0000
001      0110  0001  0111
010      0110  0001  0111
011      0110  0001  xxxx
100      0110  0001  0110
101      0110  0001  1110
110      0110  0001  0101
111      0110  0001  0000
111      0110  1111  0001
111      0001  0001  0000

gtkwave Ejercicio_3_tb.vcd Ejercicio_3_tb.gtkw
WM Destroy

```



Las entradas A y B son de cuatro bits; en el módulo de la ALU, se implementó un case, en el cual se analiza la entrada de selección. Dentro de las opciones que se

tienen en la ALU están: (A and B), (A or B), (A + B), (A and B'), (A or B'), (A - B) y de comparación.