Entrega de arquivos em verilog

Portas lógicas elementares

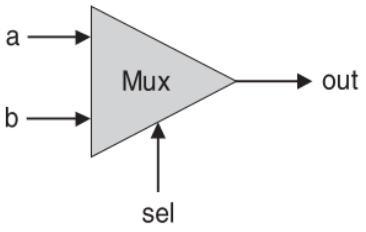
- and, or, not, xor (usar as implementadas no **iverilog**, não precisa implementar)
  Exemplos de uso:

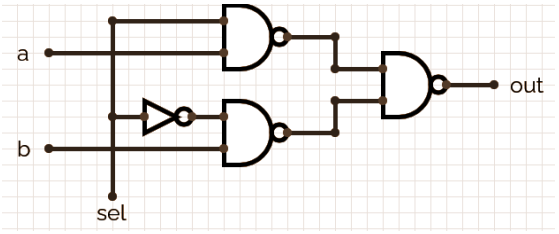
```
and and0(out, in1, in2)
or or0(out, in1, in2)
not not0(out, in)
xor xor0(out, in1, in2)
```

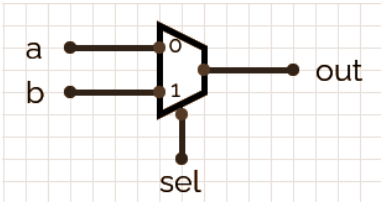
- Mux

a	b	sel	out
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

sel	out
0	a
1	b

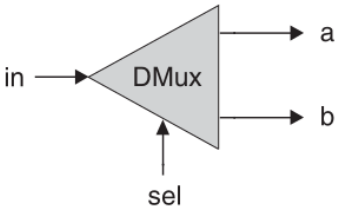


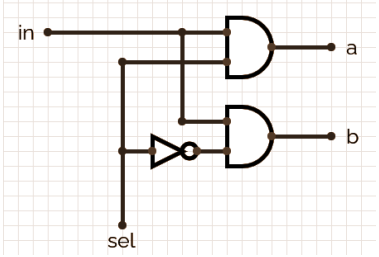


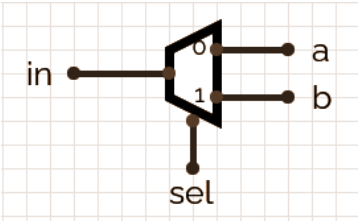


- DMux

sel	a	b
0	in	0
1	0	in

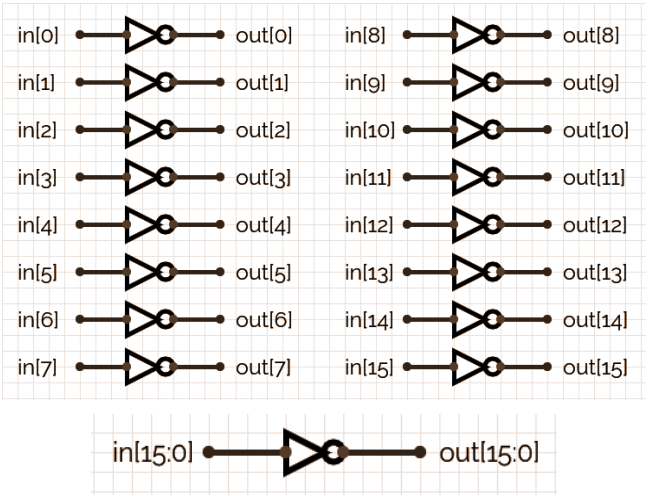






Variantes de 16-bit

- Not16



- And16
  Inspire-se na Not16

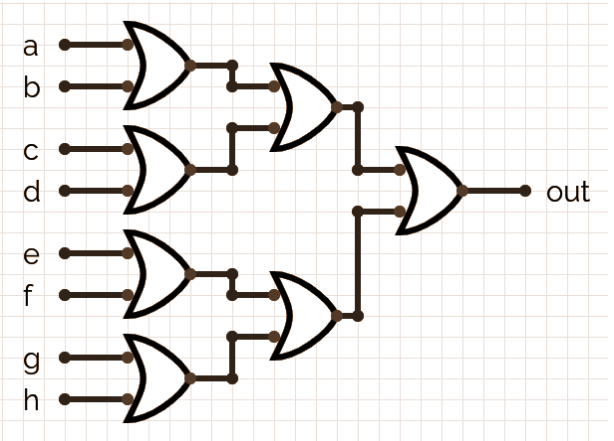
- Or16
  Inspire-se na Not16

- Mux16
  Inspire-se na Not16

- DMux16
  Inspire-se na Not16

Variantes “Multi-way”

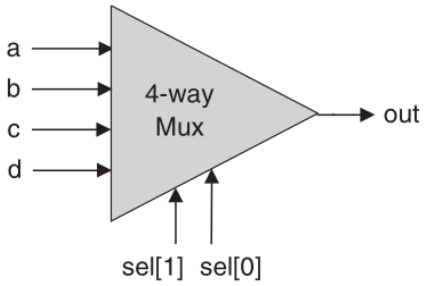
- Or8Way

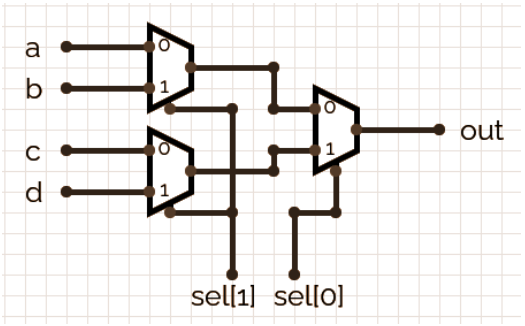


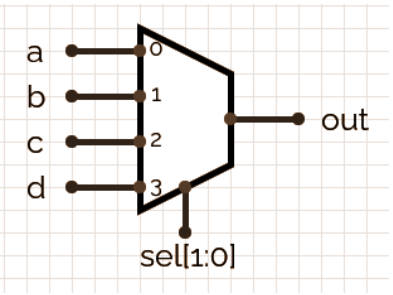
- And8Way
  Inspire-se na Or8Way

- Mux4Way

sel[1]	sel[0]	out
0	0	a
0	1	b
1	0	c
1	1	d

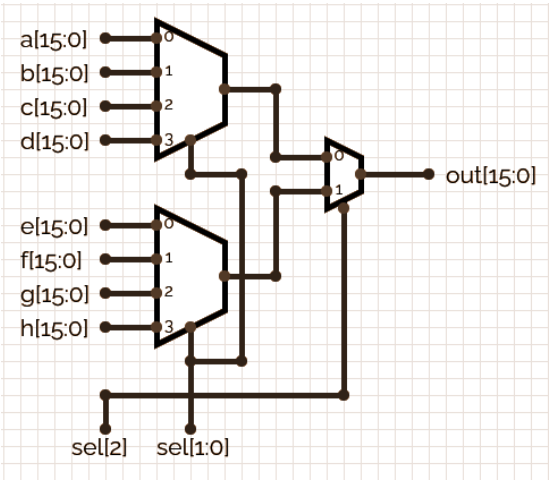






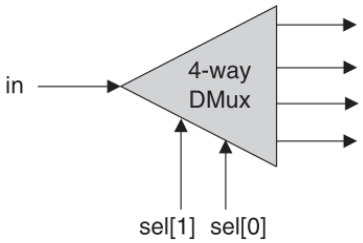
- Mux4Way16
  Dica: inspire-se na Mux4Way, mas use a Mux16

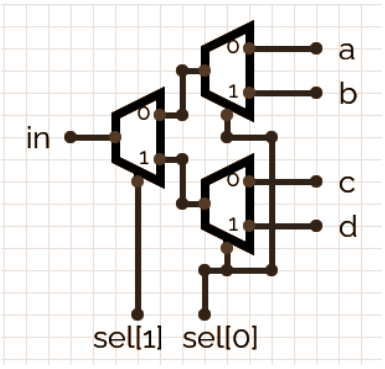
- Mux8Way16
  Dica: use ambas Mux4Way16 e Mux16

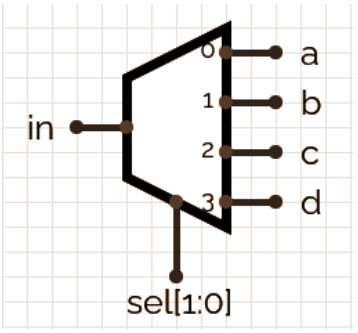


- DMux4Way

sel[1]	sel[0]	a	b	c	d
0	0	in	0	0	0
0	1	0	in	0	0
1	0	0	0	in	0
1	1	0	0	0	in





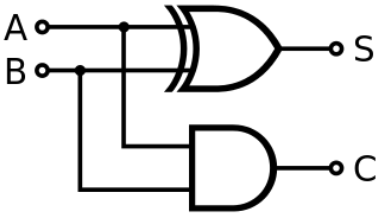


- DMux4Way16
  Dica: inspire-se na DMux4Way, mas use a DMux16

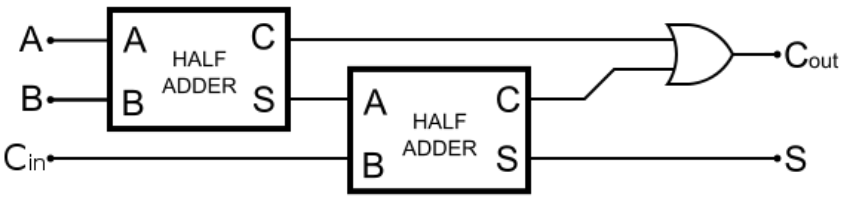
- DMux8Way16
  Dica: inspire-se nos esquemas da DMux4Way com influência da Mux8Way16, mas adapte para o uso de ambas DMux16 e DMux4Way16

Aritmética Booleana

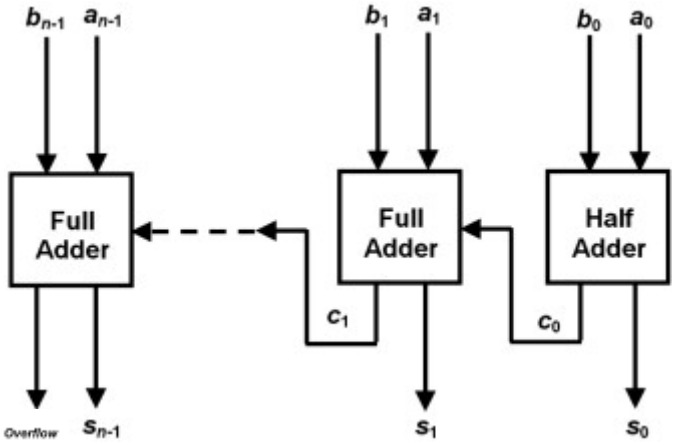
- HalfAdder



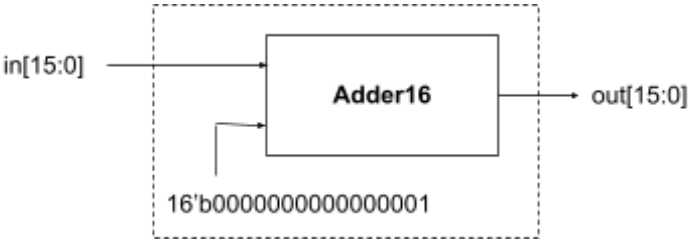
- FullAdder



- Adder16



- Increment16



Para a ordem de entradas e saídas nas declarações de seus módulos, favor consultar os respectivos arquivos de *testbench*