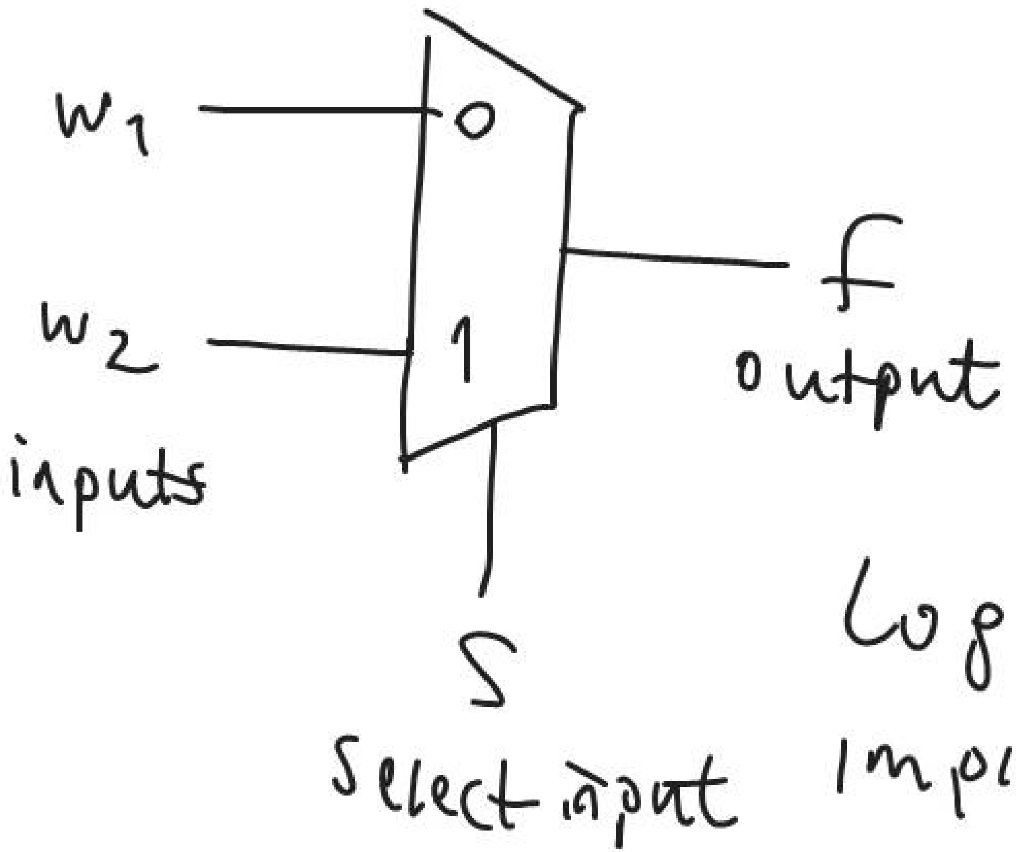
A MULTIPLEXER

A 2-to-1 mux



s	f
0	w_1
1	w_2

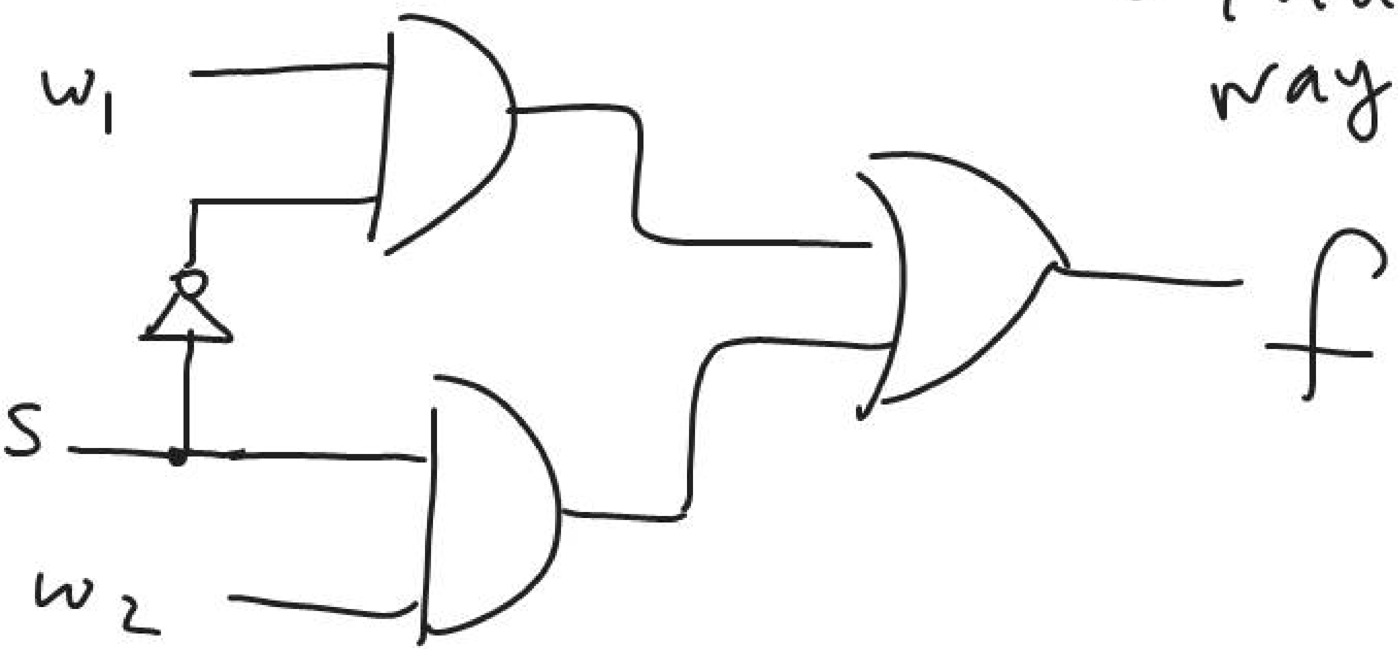
$$f(w_1, w_2) = \bar{S}\, w_1 + S\, w_2$$

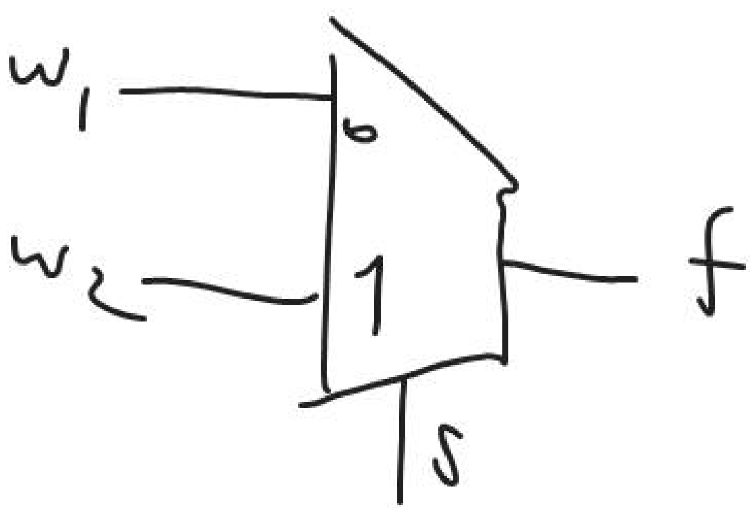
Logic functions can be implemented by multiplexers.

Shannon's Theorem.

$$f(S, w_1, w_2) = \bar{S} w_1 + S w_2$$

STRUCTURAL way of representing mux

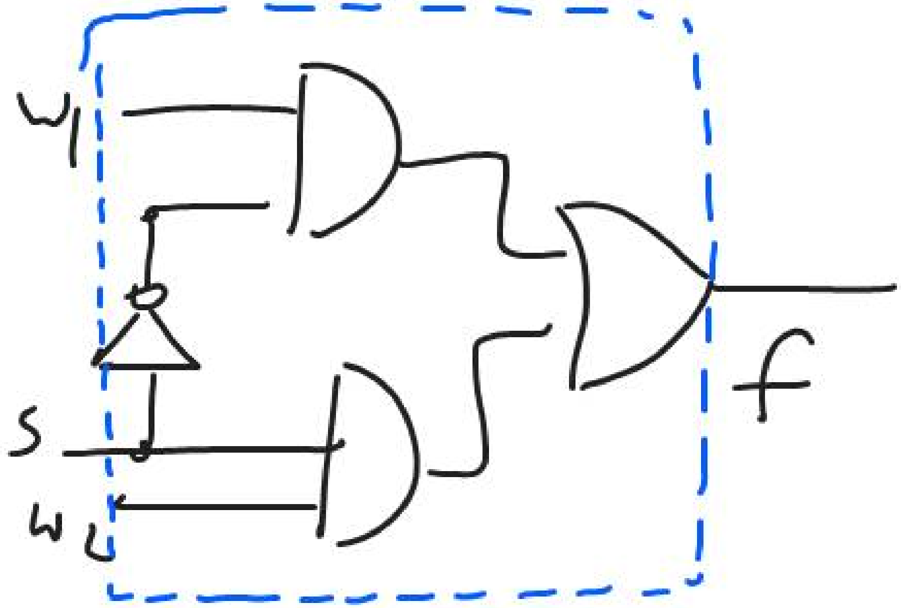




a BEHAVIORAL way of representing the mux

VHDL code for the mux:

STRUCTURAL

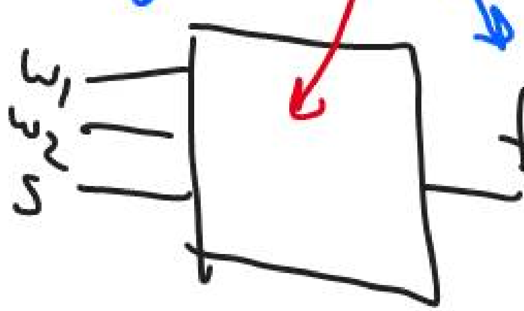


VHDL

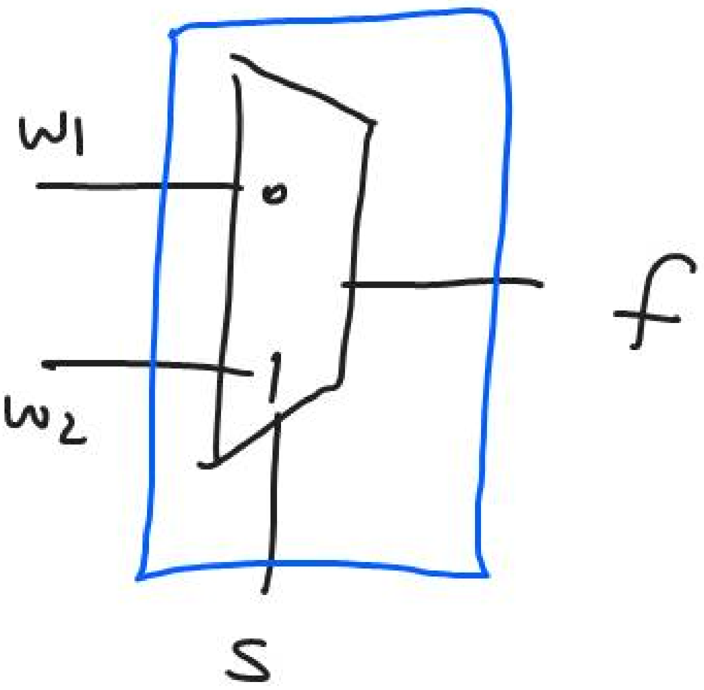
```
ENTITY MUX2TO1 IS
PORT( w1, w2, S : IN BIT;
      f         : OUT BIT);
END MUX2TO1;

ARCHITECTURE Structure of
MUX2TO1 IS
BEGIN
f <= (W1 AND NOT S) OR (
      W2 AND S);
END Structure;
```

INPUTS	OUTPUT(S)
w_1	f
w_2	
s	



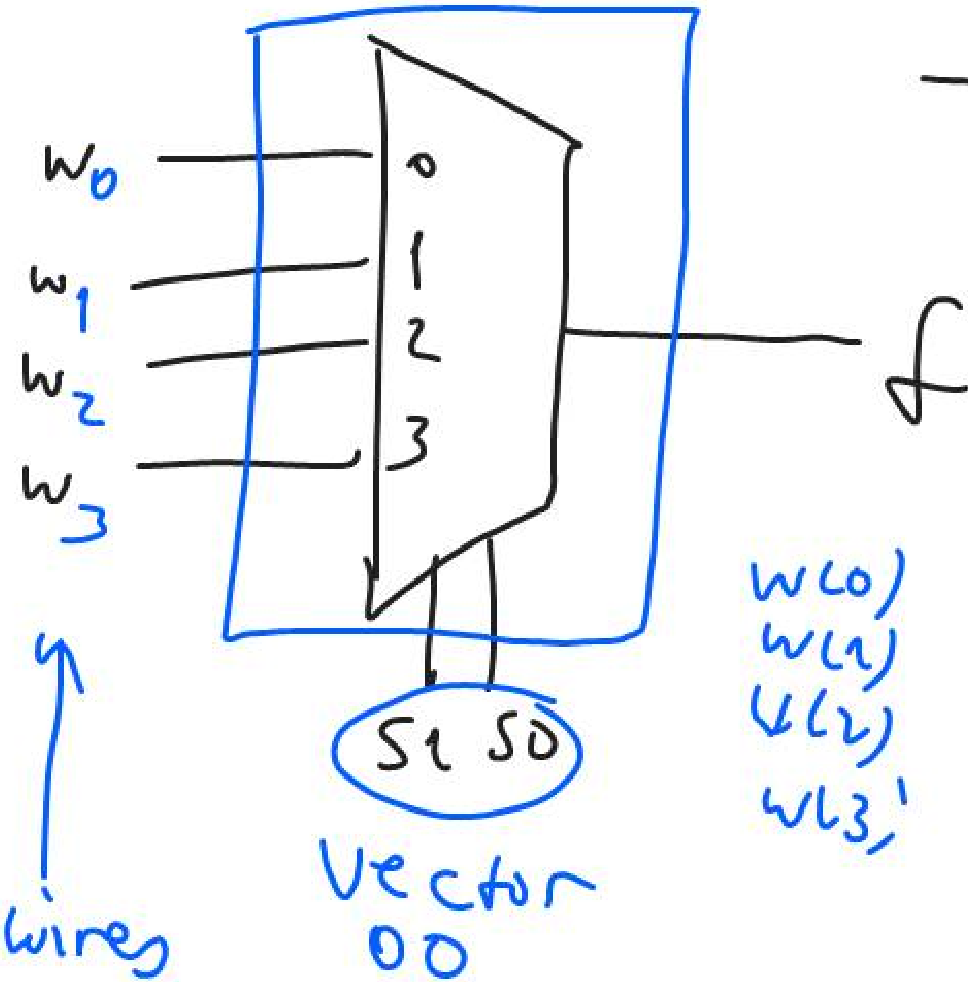
BEHAVIORAL VHDL CODE OF mux




```
ENTITY mux2to1 IS
PORT ( w1, w2, s : IN  BIT;
       f         : OUT BIT );
END mux2to1 ;

ARCHITECTURE Behavior of mux2to
IS
BEGIN
with s SELECT
 f <= w1 WHEN '0',
      w2 WHEN OTHERS ;
END Behavior ;
```

If we have a 4-to-1 mux:



w(0)
w(1)
w(2)
w(3)

Vector
00
01
10
11
:

BEHAVIOR:

```
ENTITY Mux4to1 IS
PORT( W = IN STD LOGIC
      VECTOR (0 TO 3);
S : IN STD_LOGIC_VECTOR
      (1 DOWN TO 0);
f : OUT STD_LOGIC );
END Mux4TO1 ;
ARCHITECTURE Behavior of Mux4to1
IS
BEGIN
```

'0', '1'

"00"

"01001"

WITH S SELECT

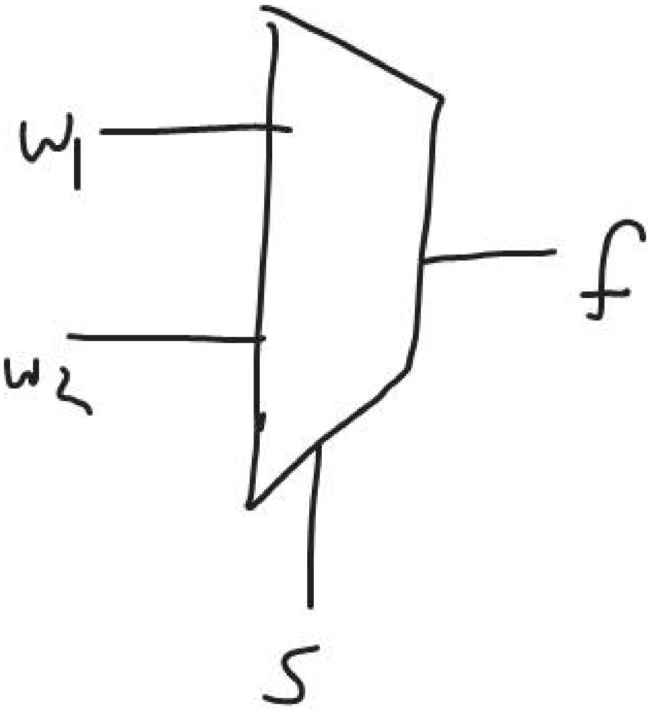
f <= w(0) WHEN "00",

w(1) WHEN "01",

w(2) WHEN "10",

w(3) WHEN OTHERS;

END Behavior;

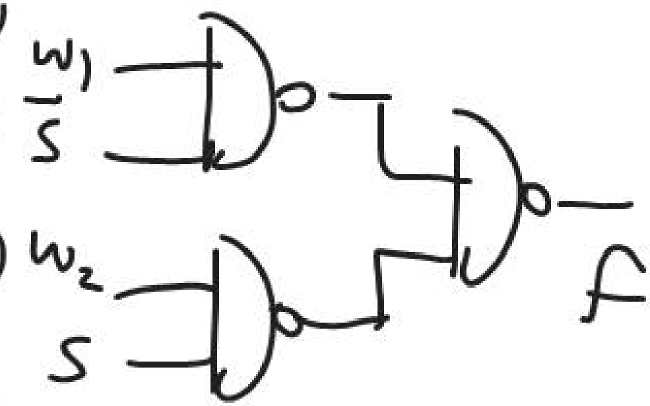


$f(s, w_1, w_2) = m(2) + m(3) + m(5) + m(7)$

minterms

Sum of Products (SOP)

s	w_1	w_2	f	
0	0	0	0	$m(0)$
0	0	1	0	$m(1)$
0	1	0	1	$m(2)$
0	1	1	1	$m(3)$
1	0	0	0	$m(4)$
1	0	1	1	$m(5)$
1	1	0	0	$m(6)$
1	1	1	1	$m(7)$



Truth Table:

s \ w_1w_2	00	01	11	10
0			1	1
1		1	1	

$f = \bar{s}w_1 + s w_2$

ex:

Implementing with muxs (Shannon's Theorem)

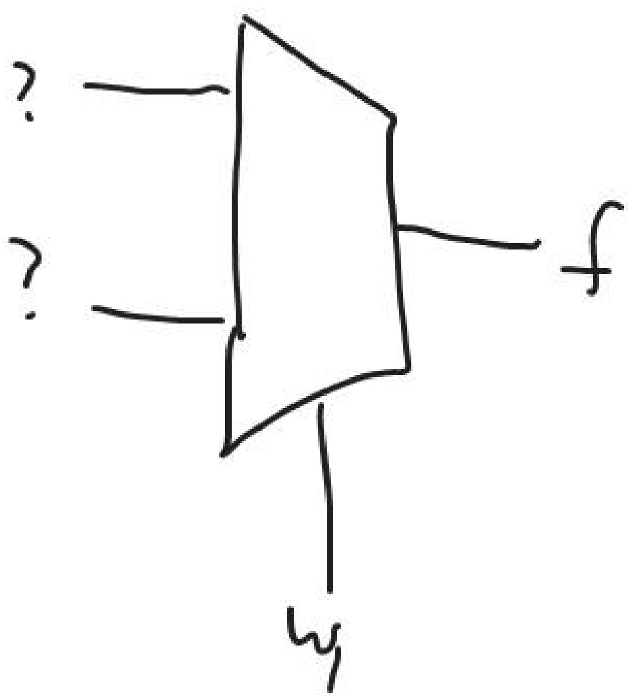
$$f(w_1, w_2, w_3) = w_1 w_2 + w_1 w_3 + w_2 w_3$$

Majority function

$$f(w_1, w_2, w_3) = \bar{w}_1 f(\bar{w}_1) + w_1 f(w_1)$$

cofactors of $\bar{w}_1$

cofactors of w_1



$$f = w_1 w_2 + w_1 w_3 + w_2 w_3$$

$$= w_1 (w_2 + w_3) + w_2 w_3 \overbrace{(\bar{w}_1 + w_1)}^{1}$$

$$= w_1 (w_2 + w_3) + \bar{w}_1 w_2 w_3 + w_2 w_3 w_1$$

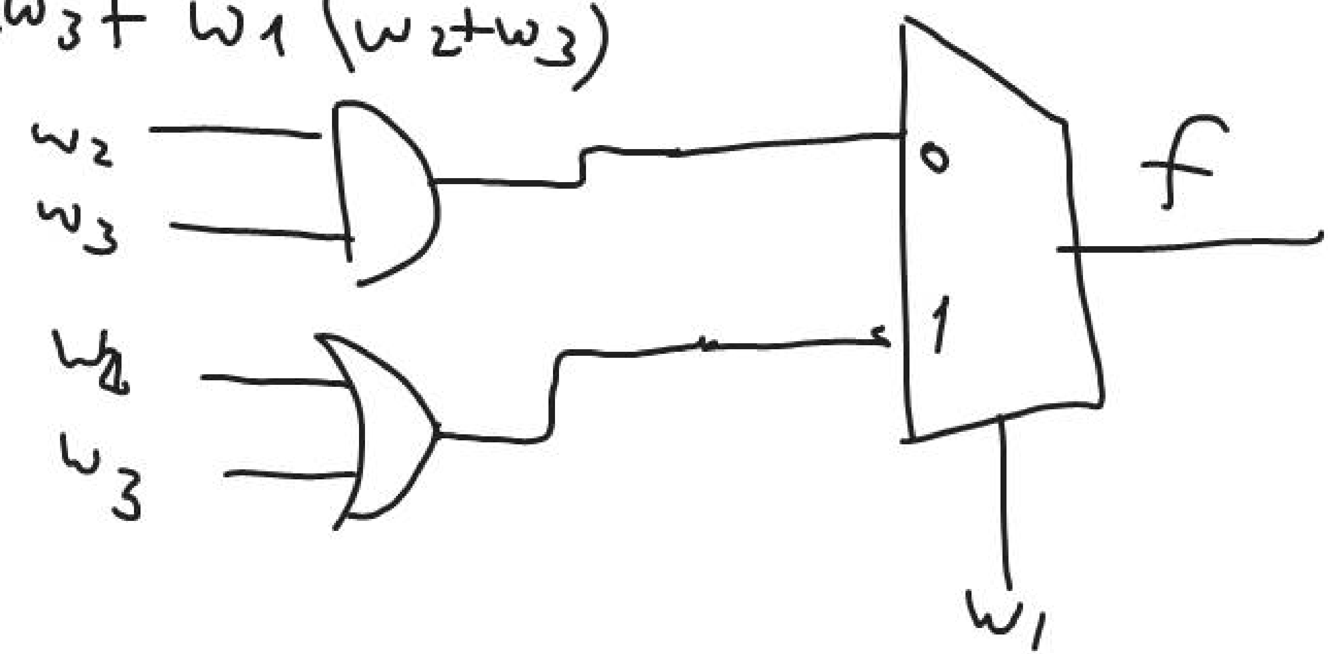
$$= w_1 (w_2 + w_3) + w_1 w_2 w_3 + \bar{w}_1 w_2 w_3$$

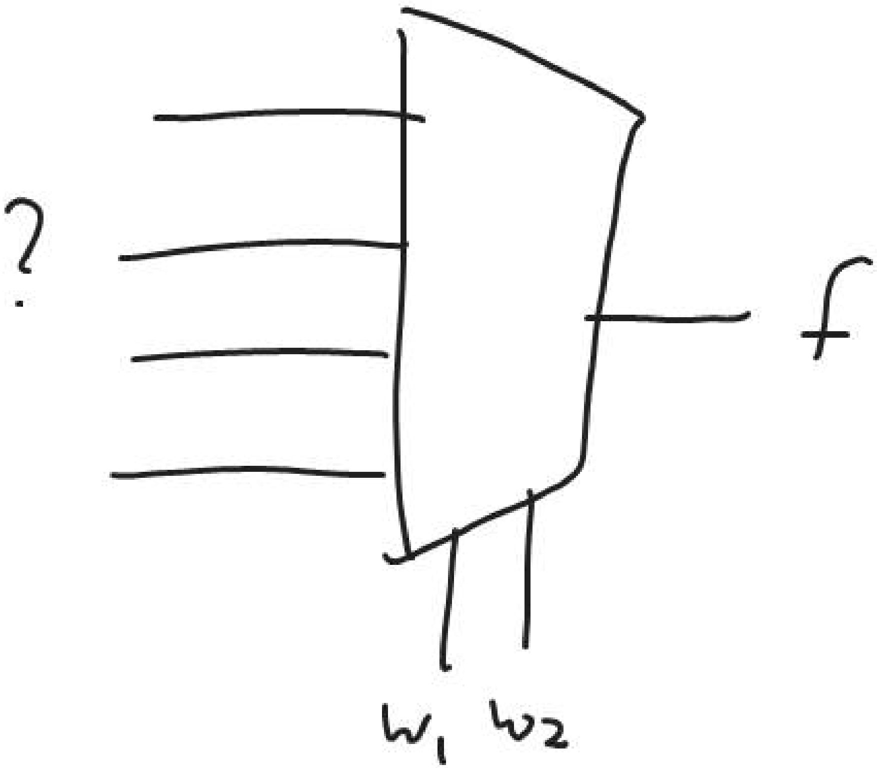
$$= \bar{w}_1 w_2 w_3 + w_1 (w_2 + w_3) + w_1 w_2 w_3$$

$$= \bar{w}_1 w_2 w_3 + w_1 [\underbrace{w_2 + w_3 + w_2 w_3}_{w_2 + w_3(\underbrace{1 + w_2}_{1})}]$$

$$\overbrace{w_2 + w_3 \cdot 1} = w_2 + w_3$$

$$f = \overbrace{\bar{w}_1 w_2 w_3}^{f(\bar{w}_1)} + w_1 \overbrace{(w_2 + w_3)}^{f(w_1)}$$





$$f = \bar{w}_1\bar{w}_2(\cancel{8}) + \bar{w}_1 w_2(\quad) + w_1\bar{w}_2(\quad) + w_1 w_2(\quad)$$

DESIGN OF DIGITAL ICs

Design