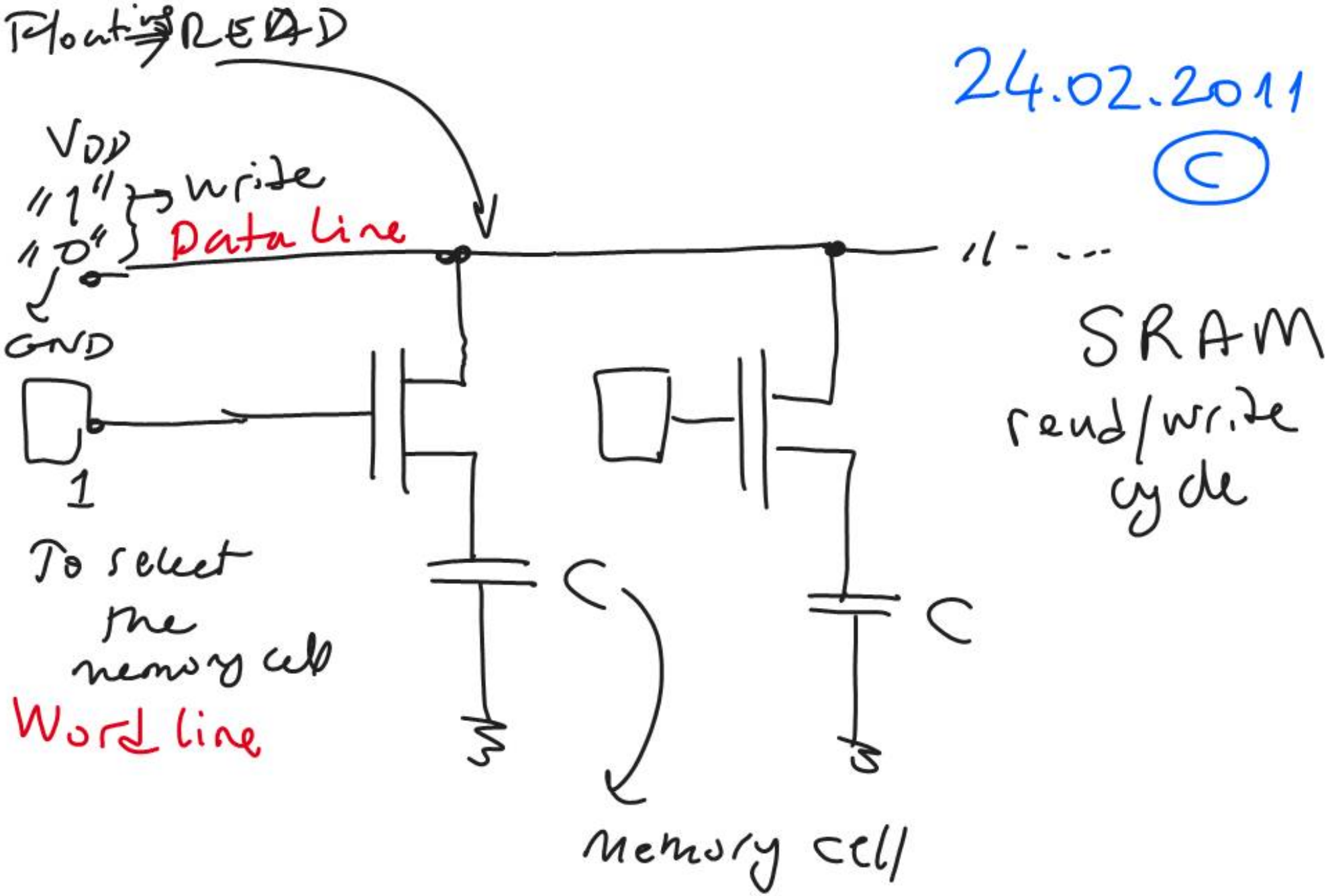
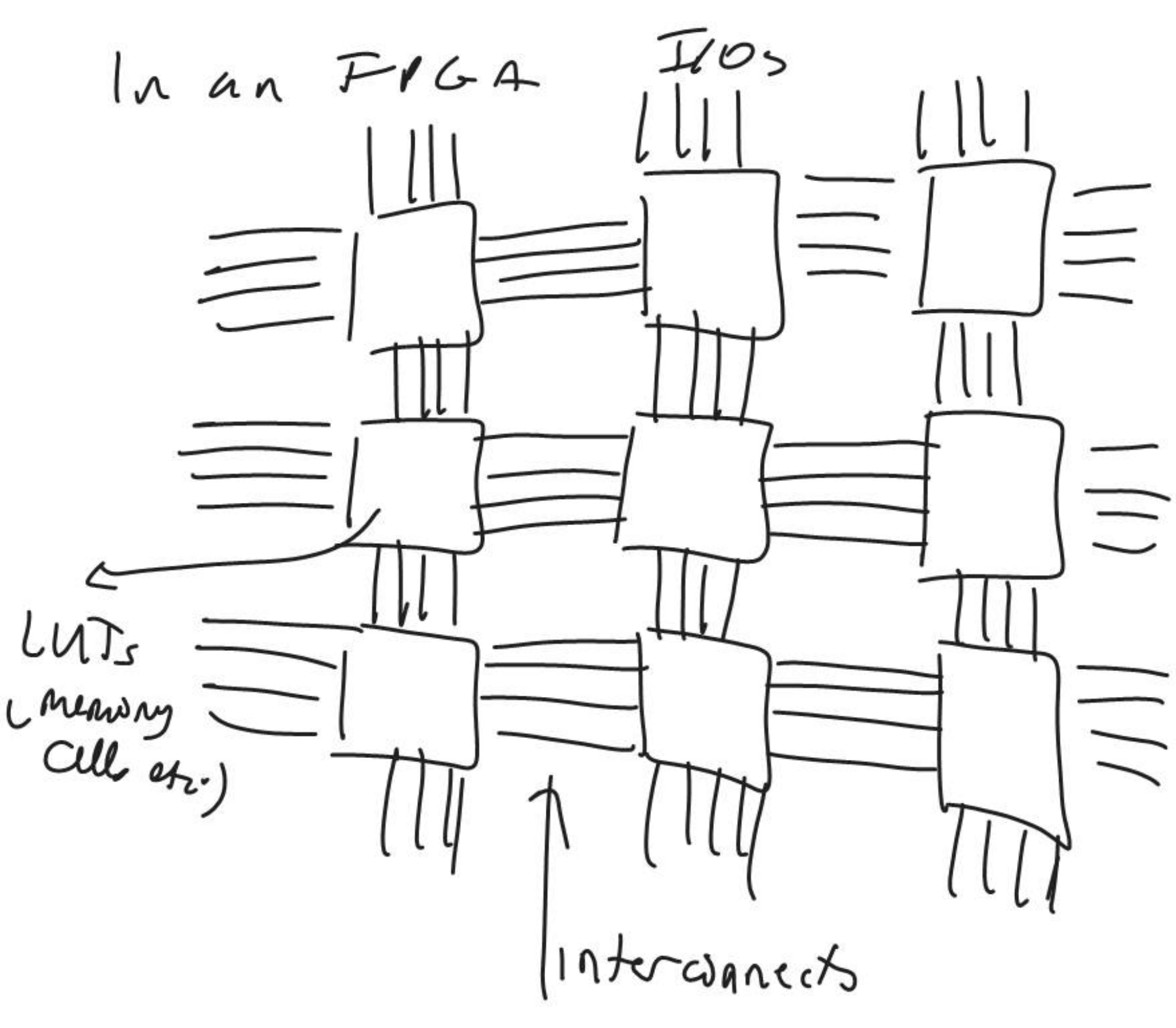
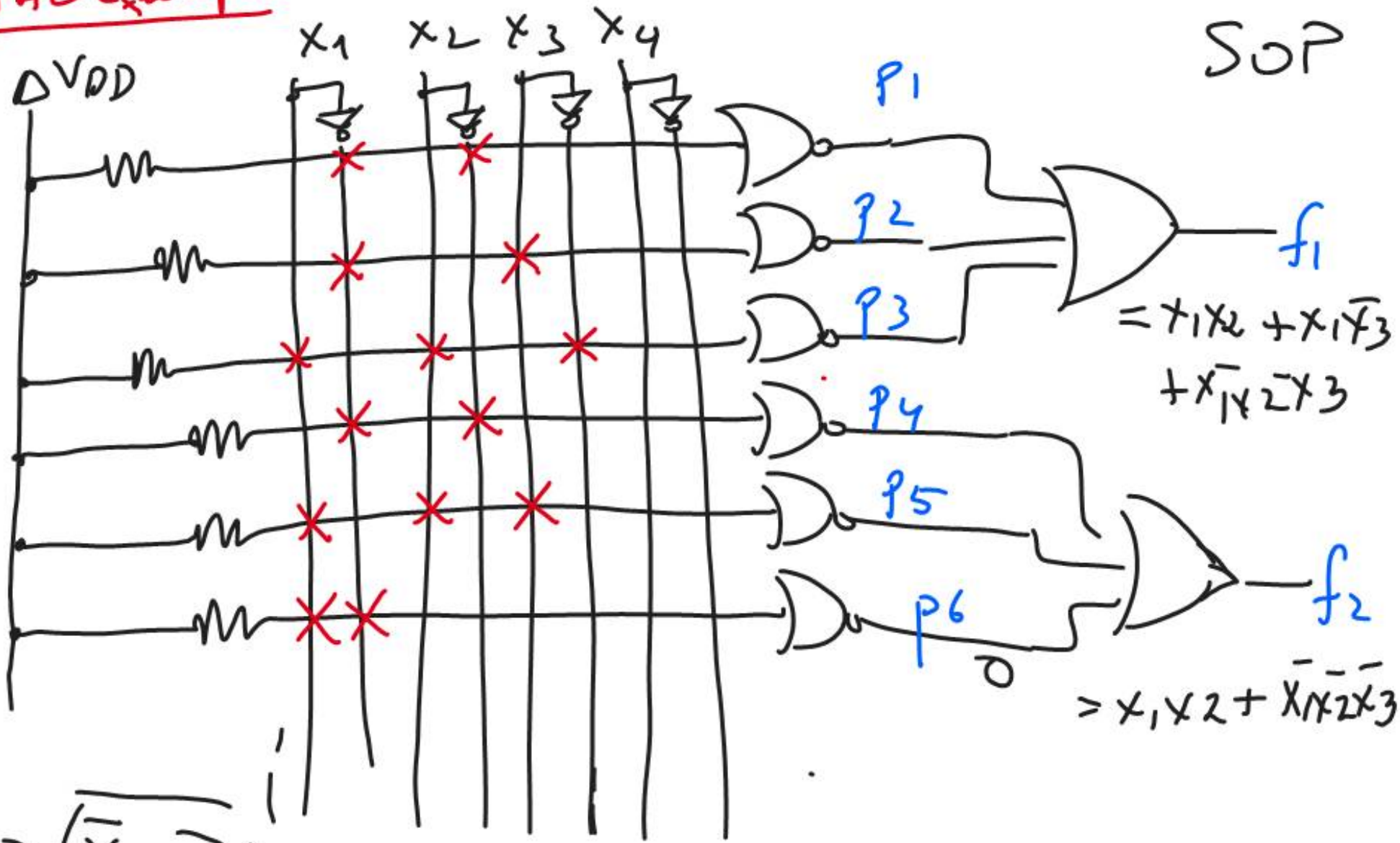

Floating READ
24.02.2011
C
VDD
"1"
"0"
write
Data line
GND
SRAM
read/write
cycle
1
To select
the
memory cell
Word line
C
C
Memory cell


In an FPGA
I/Os
LUTs
(Memory
Cells etc.)
Interconnects

APAL example



$P_1 = \overline{(\bar{x}_1 + \bar{x}_2)} = x_1x_2$

$P_2 = x_1\bar{x}_3$

$P_3 = \bar{x}_1\bar{x}_2x_3$

$P_4 = x_1x_2$

$P_5 = \bar{x}_1\bar{x}_2\bar{x}_3$

$P_6 = \bar{x}_1x = 0$

FPGA with LUTs of 2 bit

x_1	x_6	A
0	0	0
0	1	0
1	0	1
1	1	0

$A = x_1 \bar{x}_6$

x_2	x_7	B
0	0	0
0	1	0
1	0	0
1	1	1

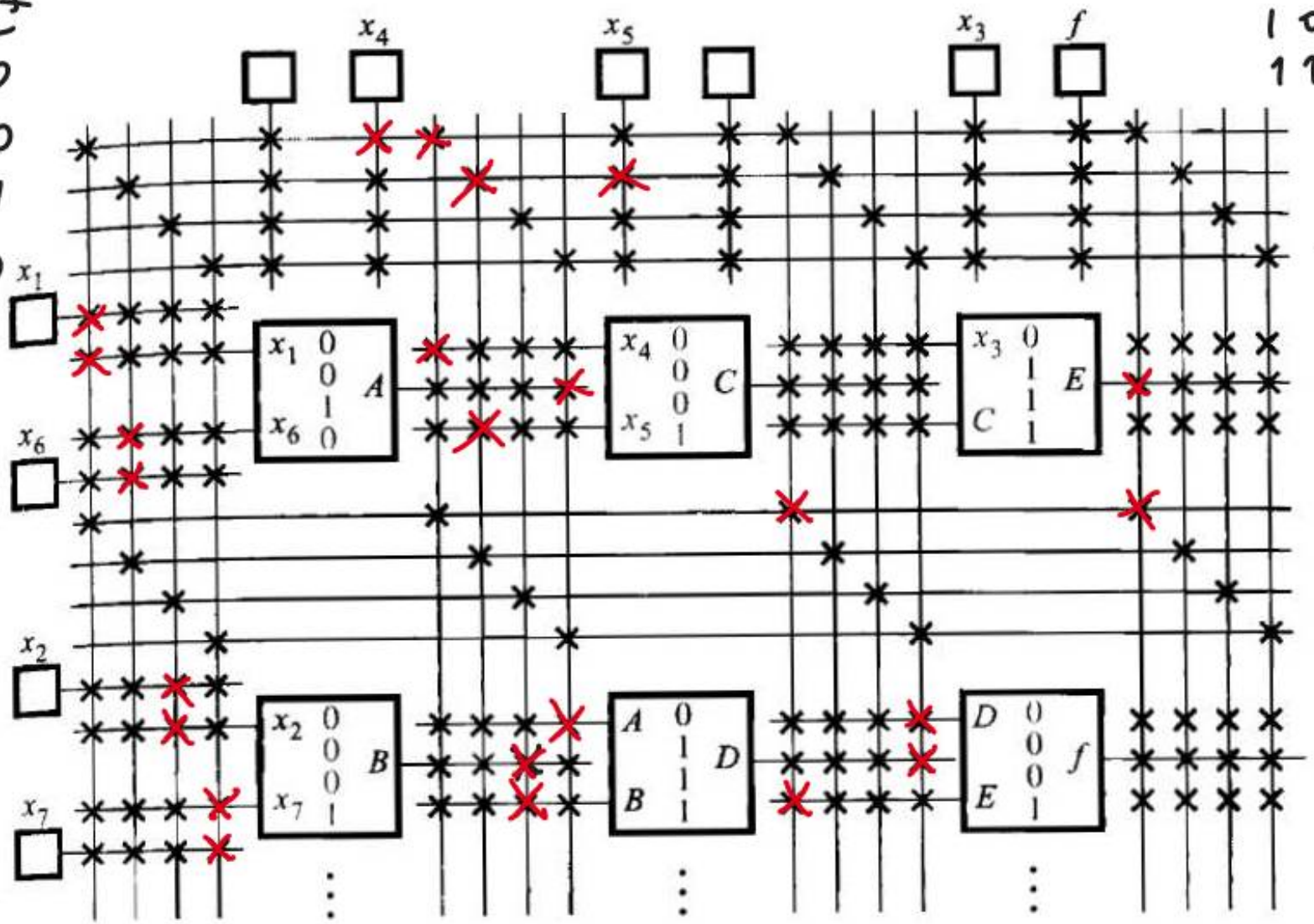
$B = x_2 x_7$ AND

x	y	E
0	0	0
0	1	1
1	0	1
1	1	0

A	B	D
0	0	0
0	1	1
1	0	1
1	1	1

$D = A + B$

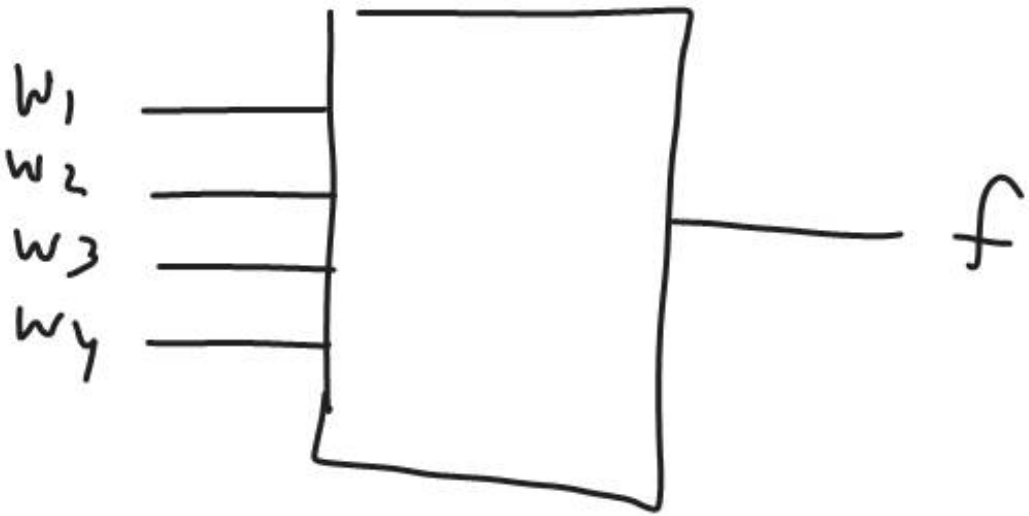
$D = x_1 \bar{x}_6 + x_2 x_7$



ex:

$$f = \bar{w}_2 w_3 + \bar{w}_1 w_2 \bar{w}_3 + w_2 \bar{w}_3 w_4 + w_1 \bar{w}_2 \bar{w}_4$$

$$f = f(w_1, w_2, w_3, w_4)$$

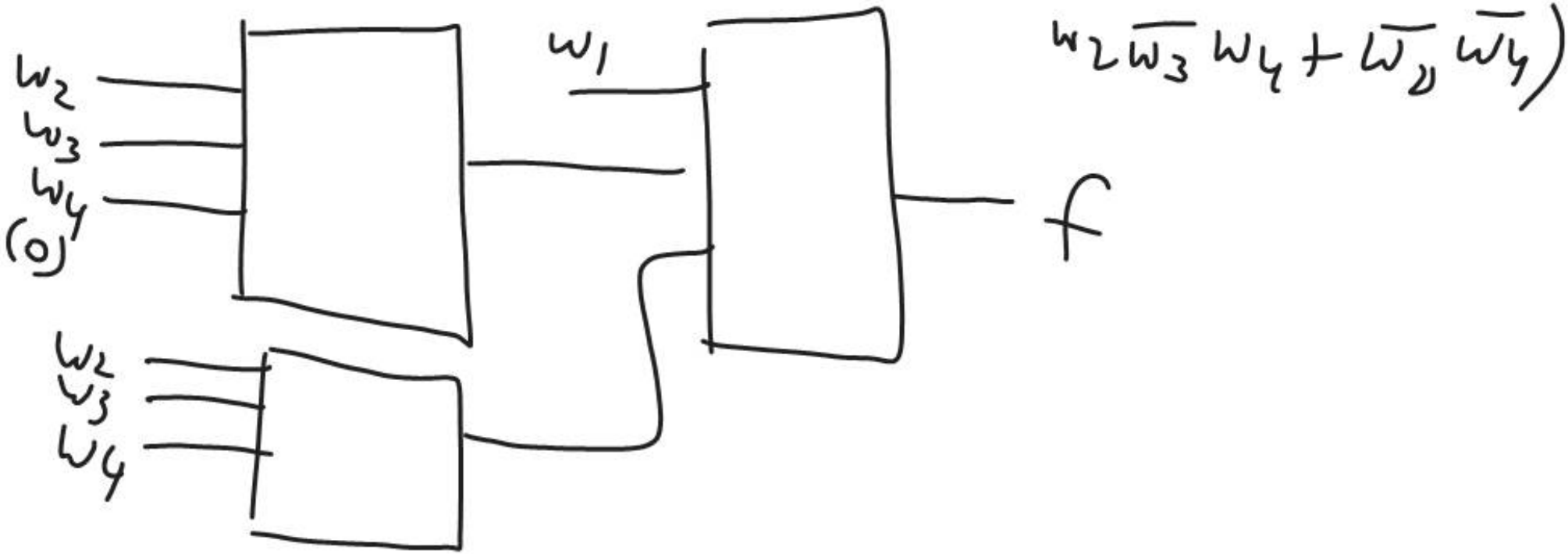


1 FPGA block of 4 input is enough to implement this function

If we have 3 input FPGA blocks how it is implemented?

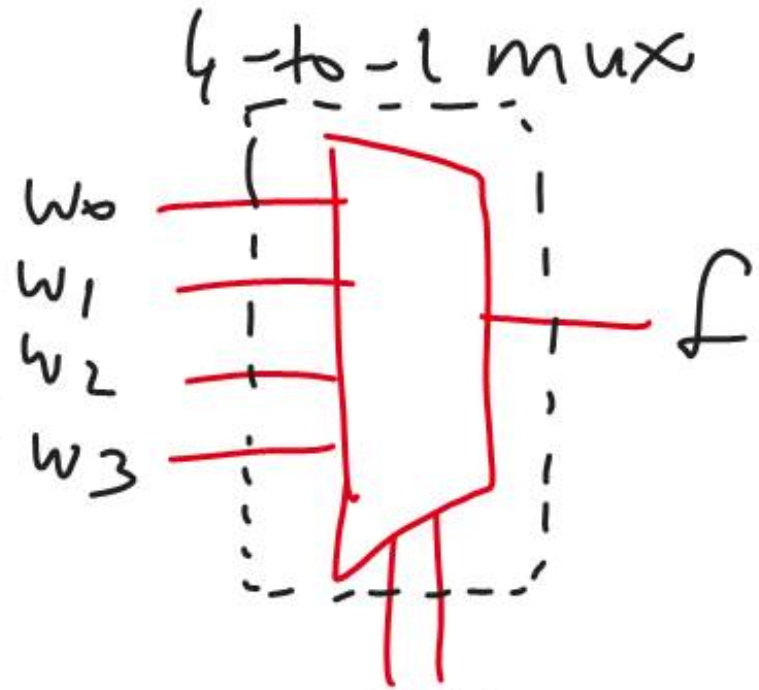
If we reduce this function after some algebra

$$f = \bar{w}_1(\bar{w}_2 w_3 + w_2 \bar{w}_3) + w_1(\bar{w}_2 w_3 + w_2 \bar{w}_3 w_4 + \bar{w}_2 \bar{w}_4)$$



USING SUBCIRCUITS IN THE VHDL

- COMPONENT
- PACKAGE




```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY mux4to1 IS
PORT ( W: IN STD_LOGIC_VECTOR (0 TO 3);
       S: IN STD_LOGIC_VECTOR (1 DOWNTO 0);
       f: OUT STD_LOGIC );

END mux4to1;
```

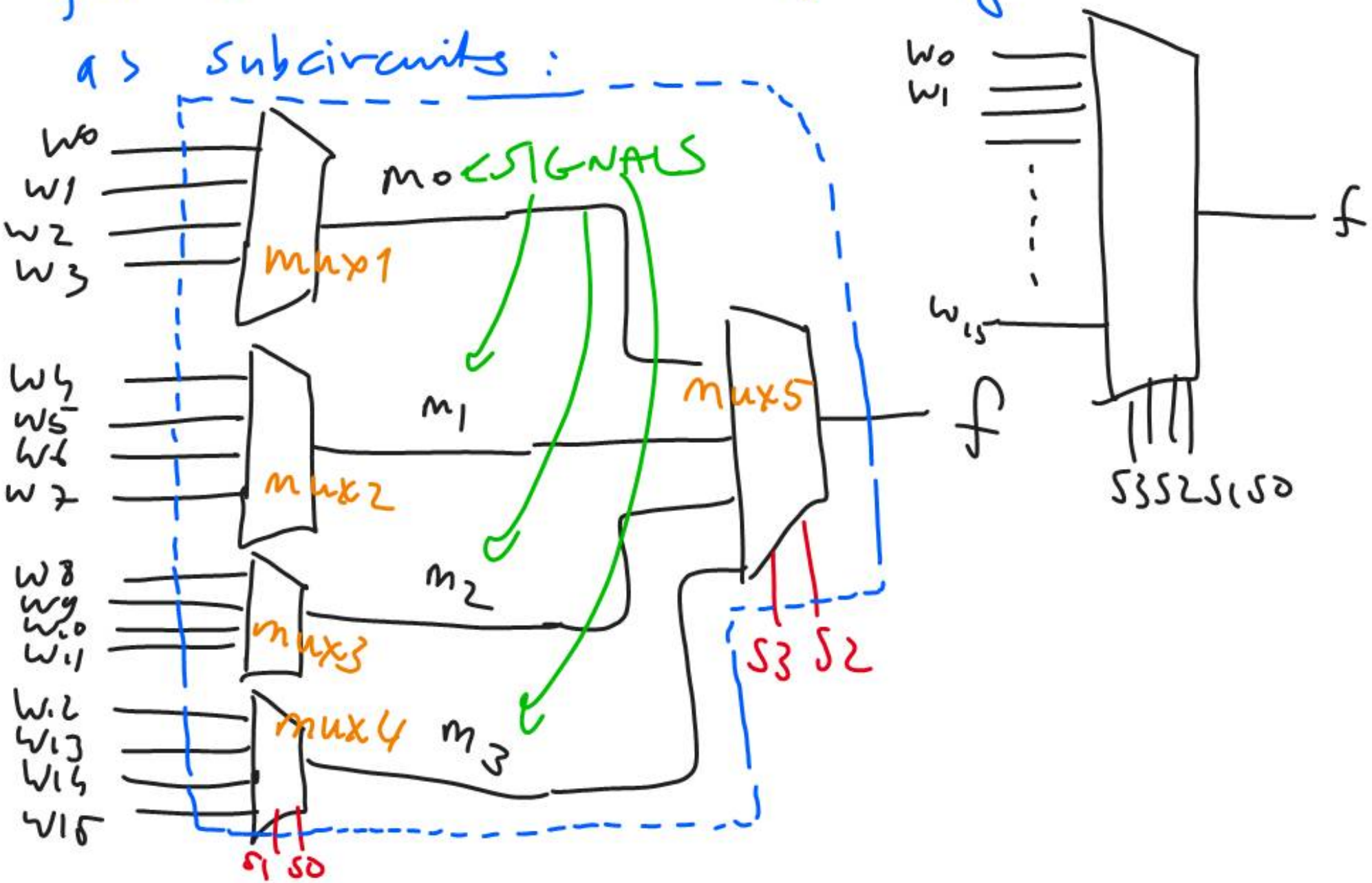
```
ARCHITECTURE Behavior OF MUX4TO1 IS
BEGIN

  WITH S SELECT
    f <= W0 WHEN "00"
         W1 WHEN "01"
         W2 WHEN "10"
         W3 WHEN OTHERS;
  END BEHAVIOR;
```

If I want this to be used in the future it is better to make it a PACKAGE

To make mux4to1 a PACKAGE for later use:

```
LIBRARY ieee;

USE ieee.std_logic_1164.all;

PACKAGE mux4to1_package IS
COMPONENT mux4to1
PORT( W : IN STD_LOGIC_VECTOR(0 TO 3);
      S : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
      f : OUT STD_LOGIC);

END COMPONENT;

END mux4to1_package;
```

Assume we need to generate a VHDL code for a 16-to-1 mux by using 4-to-1 muxes as subcircuits:



Developing 16-to-1 mux VHDL code by using the mux4-to-1 code as a subcircuit (PACKAGE)

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY work;
USE work.mux4to1_package.all;

ENTITY mux16to1 IS
PORT ( W : IN STD_LOGIC_VECTOR (0 TO 15);
       S : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
       f : OUT STD_LOGIC);

END mux16to1;
```

```
ARCHITECTURE STRUCTURE OF MUX16TO1 IS
  SIGNAL m : STD_LOGIC_VECTOR (0 TO 3);
BEGIN
  MUX1 : MUX4TO1 PORT MAP
    (w(0), w(1), w(2), w(3), S(1 DOWNTO 0),
     m(0));
  MUX2 : MUX4TO1 PORT MAP
    (w(4), w(5), w(6), w(7), S(1 DOWNTO 0), m(1));
```

* name

name

calls the subcircuit (package)