ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

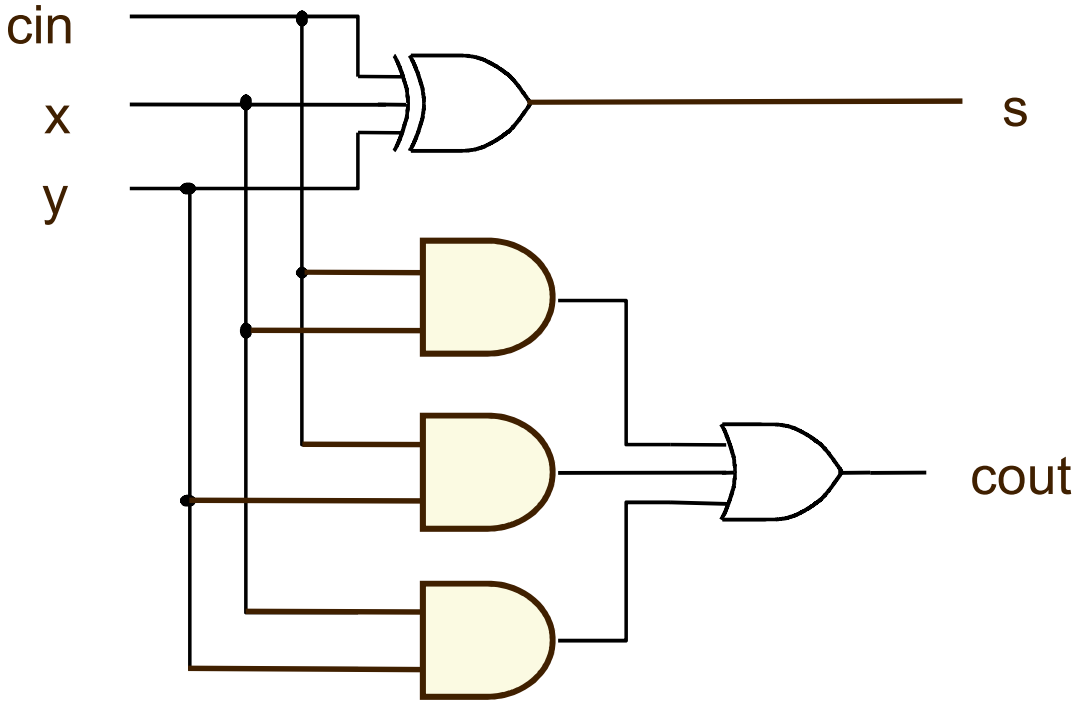
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Παρουσίαση 2η: Συνδυαστικά μπλοκ στη VHDL

2022-2023

Data-flow VHDL Example 1: Full adder




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

ENTITY fulladd IS PORT (
 x    : IN BIT;
 y    : IN BIT;
 cin  : IN BIT;
 s    : OUT BIT;
 cout : OUT BIT);
END fulladd ;
```

```
ARCHITECTURE dataflow OF fulladd IS
BEGIN
 s    <=  x XOR y XOR cin;
 cout <= (x AND y) OR (cin AND x) OR (cin AND y);
END dataflow ;
```

Operators

- Relational operators

=	/=	<	<=	>	>=

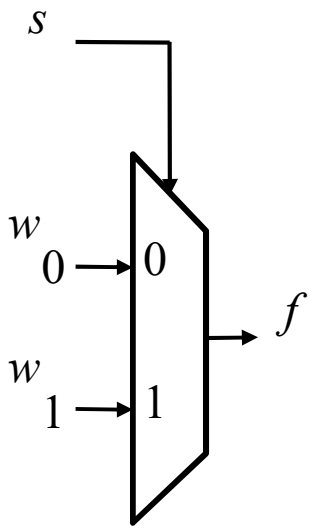
- Logic and relational operators precedence

Highest

Lowest

		not			
=	/=	<	<=	>	>=
and	or	nand	nor	xor	xnor

Χρήση της δομής when: 2-to-1 Multiplexer



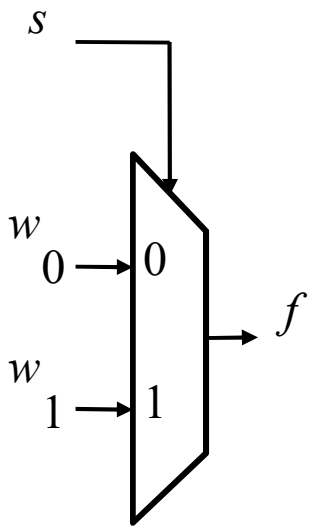
Graphical symbol

s	f
0	w_0
1	w_1

Truth table

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

ENTITY mux2to1 IS PORT (
 w0, w1, s: IN STD_LOGIC;
 f: OUT STD_LOGIC);
END mux2to1 ;

ARCHITECTURE dataflow OF mux2to1 IS
BEGIN
    f <= w0 WHEN s = '0' ELSE w1;
END dataflow;
```

Χρήση της δομής when: 2-to-1 Multiplexer



Graphical symbol

s	f
0	w_0
1	w_1

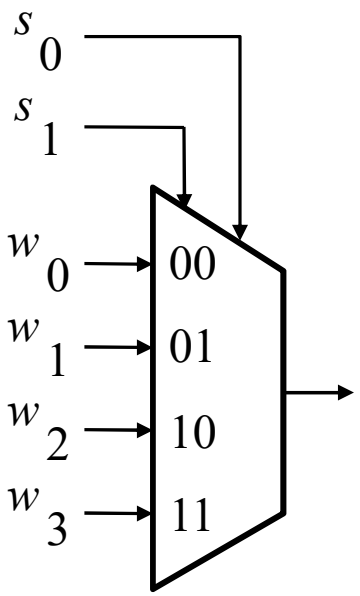
Truth table

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

ENTITY mux2to1 IS PORT (
 w0, w1, s: IN STD_LOGIC;
 f: OUT STD_LOGIC);
END mux2to1 ;

ARCHITECTURE dataflow OF mux2to1 IS
BEGIN
    f <= w0 WHEN s = '0' ELSE w1;
END dataflow;
```

Άσκηση 1: Να φτιαχτεί 2-bit multiplexer 2-σε-1 σε VHDL

Άσκηση 2: Να φτιαχτεί 32-bit multiplexer 2-σε-1 σε VHDL

Δομή with select: 4-to-1 Multiplexer



Graphic symbol

s_1	s_0	f
0	0	w_0
0	1	w_1
1	0	w_2
1	1	w_3

operation table

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

ENTITY mux4to1 IS PORT (
 w0, w1, w2, w3:  IN BIT;
 s: IN BIT_VECTOR(1 DOWNTO 0);
 f: OUT BIT);
END mux4to1 ;

ARCHITECTURE dataflow OF mux4to1 IS
BEGIN
 WITH s SELECT
  f <= w0 WHEN "00",
       w1 WHEN "01",
       w2 WHEN "10",
       w3 WHEN OTHERS;
END dataflow ;
```

Ποια είναι τα others?

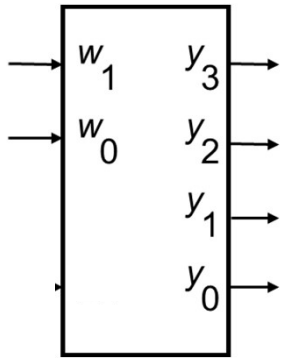
Άσκηση 1: Να φτιαχτεί 2-bit multiplexer 4-σε-1 σε VHDL

Άσκηση 2: Να φτιαχτεί 2-bit multiplexer 8-σε-1 σε VHDL

2-to-4 Decoder (without enable)

w_1	w_0	y_3	y_2	y_1	y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

(a) Truth table



(b) Graphical symbol

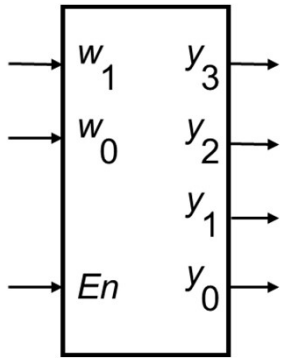
```
LIBRARY ieee ;
USE ieee.std_logic_1164.all;

ENTITY dec2to4 IS PORT (
 w : IN BIT_VECTOR(1 DOWNTO 0);
 y : OUT BIT_VECTOR(3 DOWNTO 0));
END dec2to4;

ARCHITECTURE dataflow OF dec2to4 IS
BEGIN
WITH w SELECT
 y <= "0001" WHEN "00",
      "0010" WHEN "01",
      "0100" WHEN "10",
      "1000" WHEN "11";
END dataflow ;
```

Άσκηση 1: Να φτιαχτεί 3-to-8 decoder σε VHDL

2-to-4 Decoder with enable

En	w_1	w_0	y_3	y_2	y_1	y_0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0
0	x	x	0	0	0	0

(a) Truth table



(b) Graphical symbol

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

ENTITY dec2to4 IS PORT (
 w : IN BIT_VECTOR(1 DOWNTO 0);
 En: IN BIT;
 y : OUT BIT_VECTOR(3 DOWNTO 0));
END dec2to4;

ARCHITECTURE dataflow OF dec2to4 IS
SIGNAL Enw: BIT_VECTOR(2 DOWNTO 0);
BEGIN
 Enw <= En & w ;
 WITH Enw SELECT
 y <= "0001" WHEN "100",
      "0010" WHEN "101",
      "0100" WHEN "110",
      "1000" WHEN "111",
      "0000" WHEN OTHERS ;
END dataflow ;
```

Άσκηση 1: Να φτιαχτεί 3-to-8 decode with enable

Άσκηση:

1. Να σχεδιαστεί 3-to-8 decoder χρησιμοποιώντας decoders 2-το-4

2. Να σχεδιαστεί 4-to-16 decoder χρησιμοποιώντας decoders 2-το-4