Comparator

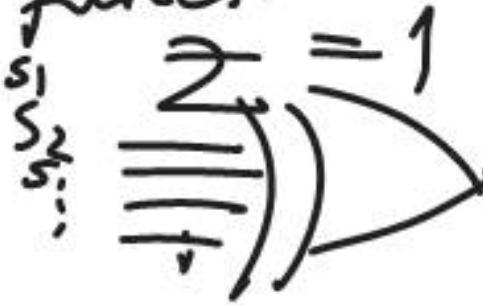
$x > y$, $x \geq y$, $x = y$, $x \leq y$, $x < y$

31.03.2011 ©

integer unsigned or signed numbers x, y

$x - y$

functions (outputs)



$Z = 1$ then $x = y$

Overflow $V = c_4 \oplus c_3$

Negative $N = s_3$

$x > y$

$\overline{Z + N \oplus V} = 1$

$x \geq y$

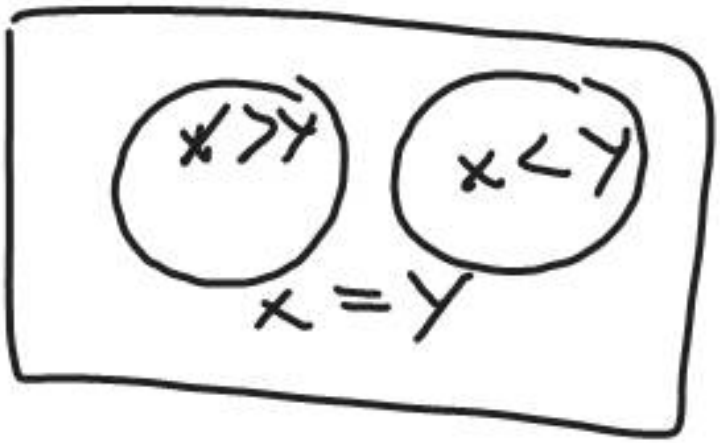
$\overline{N \oplus V} = 1$

$x < y$ $\quad N \oplus V = 1$

$x \leq y$ $\quad N \oplus V + Z = 1$

$X \geq Y \quad Z + \overline{Z + N \oplus V} = 1$

$X = Y$ OR $X > Y$



$= Z + \bar{Z} \cdot \overline{(N \oplus V)}$

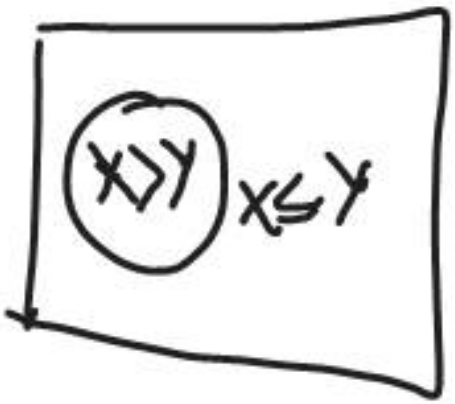
$= (Z + \bar{Z}) \cdot (Z + \overline{(N \oplus V)})$

1

$= Z + \overline{(N \oplus V)}$?

$X < Y \Rightarrow N \oplus V = 1$

$\overline{X < Y} \Rightarrow X \geq Y$



$X \geq Y \Rightarrow \overline{N \oplus V} = 1$

Behavioral Code for The Comparator

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_signed.all;
ENTITY Comparator IS
PORT(X, Y : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
     V, N, Z : OUT STD_LOGIC));
END Comparator;
```

This line enables us to write an ARITHMETIC code.

```
ARCHITECTURE Behavior OF Comparator IS
SIGNAL S: STD_LOGIC_VECTOR (4 DOWNTO 0);
BEGIN
  S <= ('0'&X) + Y;
  V <= S(4) XOR X(3) XOR Y(3) XOR S(3);
  N <= S(3);
  Z <= '1' WHEN S(3 DOWNTO 0)=0 ELSE '0';
END Behavior;
```

C(4)

(actually This is equivalent to C(3)

V = C4 ⊕ C3)

The MSB of the sum

HW: show that HW #2

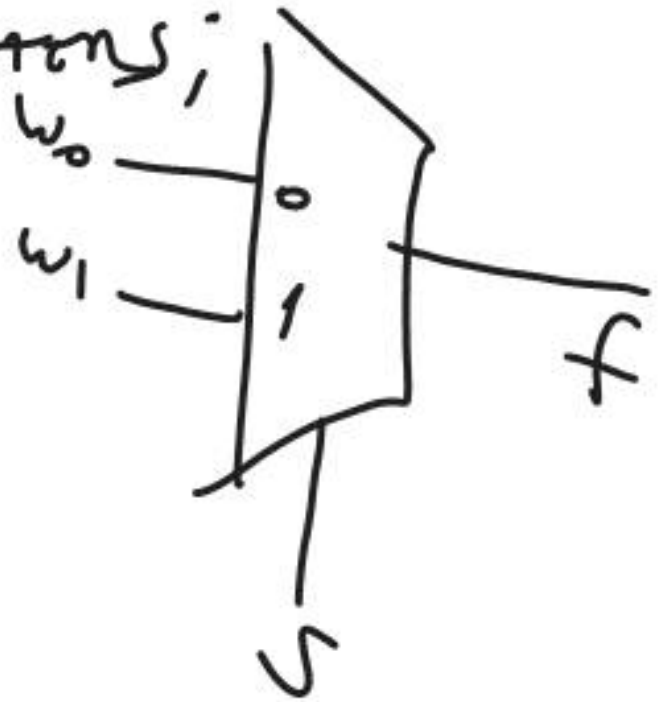
$$C(3) = X(3) \oplus Y(3) \oplus S(3)$$

ASSIGNMENT STATEMENTS in the VHDL

1. Simple Assignment statement
   f <= '1';
2. Selected Signal Assignment
3. Conditional Signal "
4. GENERATE Statements
5. IF-THEN-ELSE STATEMENTS
6. CASE STATEMENTS

② SELECTED SIGNAL ASSIGNMENT

```
ARCHITECTURE Behaviour OF MUX2TO1 IS
BEGIN
WITH S SELECT
   f <= W0 WHEN '0',
   f <= W1 WHEN OTHERS;
END Behaviour;
```



A concurrent statement
(it is not sequential)

But, we have to be very careful when we are using SELECT statement

ex: A priority encoder

w_3	w_2	w_1	w_0	y_1	y_0	z
0	0	0	0	d	d	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

don't care

not applicable

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY priority IS
PORT( W: IN STD_LOGIC_VECTOR (3 DOWNTO 0);
      Y: OUT STD_LOGIC_VECTOR (1 DOWNTO 0);
      Z: OUT STD_LOGIC);
END priority;
ARCHITECTURE Behavior OF Priority IS
BEGIN
WITH W SELECT
Y <= "00" WHEN "0001";
     "01" WHEN "0010";
     "01" WHEN "0011";
```

```
"10" WHEN "0100";
"10" WHEN "0101";
"10" WHEN "0110";
"10" WHEN "0111";

"11" WHEN OTHERS;
```

include

1 x x x

1000

1001

1010

1011

1100

1101

1110

1111

and 0000

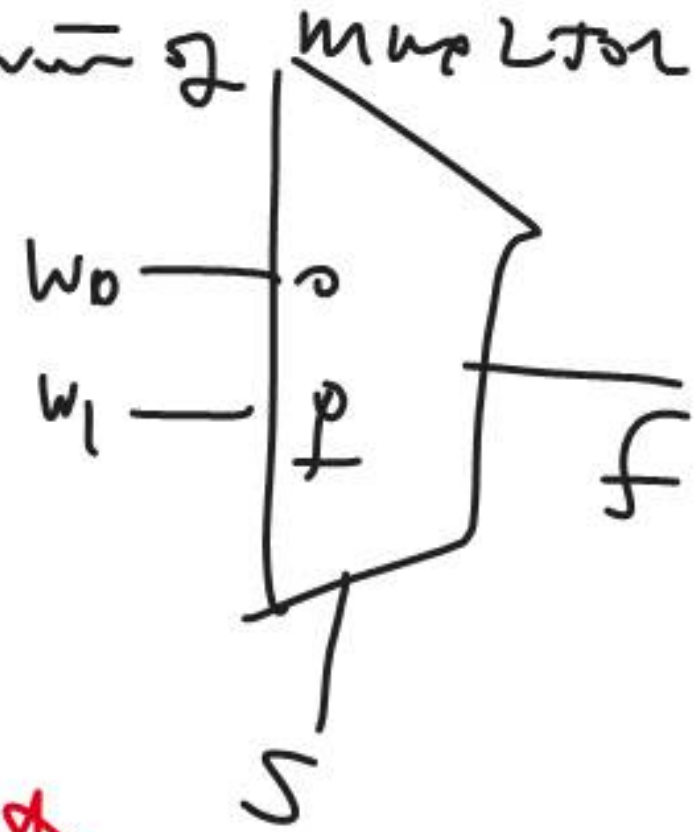
WITH W SELECT

Z <= '0' WHEN "0000"

'1' WHEN OTHERS;

③ CONDITIONAL SIGNAL ASSIGNMENT

```
ARCHITECTURE Behavior OF MUX2TO1 IS
BEGIN
f <= W0 WHEN S = '0'
     ELSE W1;
END Behavior;
```



This is also a concurrent assignment statement.

example: COMPARATOR with Arithmetic

VHDL code:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;
ENTITY Comparator IS
PORT(A,B :IN STD_LOGIC_VECTOR(3 DOWNTO 0);
AeqB, AgtB, AltB : OUT STD_LOGIC);
END Comparator;
ARCHITECTURE Behavior of Comparator IS
BEGIN
AeqB <= '1' WHEN A=B ELSE '0';
AgtB <= '1' WHEN A>B ELSE '0';
```

concurrent statements

```
AltB <= '1' WHEN A<B ELSE '0';
END Behaviour;
```

ex : Comparison with SIGNED numbers now:

```
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arithmetic.all;
ENTITY COMPARATOR IS
PORT(A,B : IN SIGNED (3 DOWNTO 0);
AeqB, AgtB, AltB : OUT STD_LOGIC);
END COMPARATOR;

-- The next one The same as above
```

④ GENERATE Statements

4a. FOR GENERATE
4b. IF GENERATE

ex: a 16-to-1 mux with 5 4-to-1 Muxes

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
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
G1: FOR i IN 0 TO 3 GENERATE;
MUX4S : MUX4TO1 PORTMAP
  w(4*i), w(4*i+1), w(4*i+2), w(4*i+3),
  S(1 DOWNTO 0), m(i));
END GENERATE;

MUX5 : MUX4TO1 PORTMAP (m(0), m(1), m(2),
       m(3), S(3 DOWNTO 2), f);

END Structure;
```