

Limbaj de descriere hardware

27.11.2024

Cu HS +

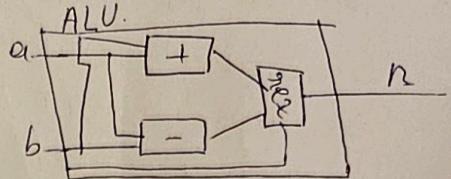
(continuare curs trecut) !!!

Simplificare 12: a două verificări

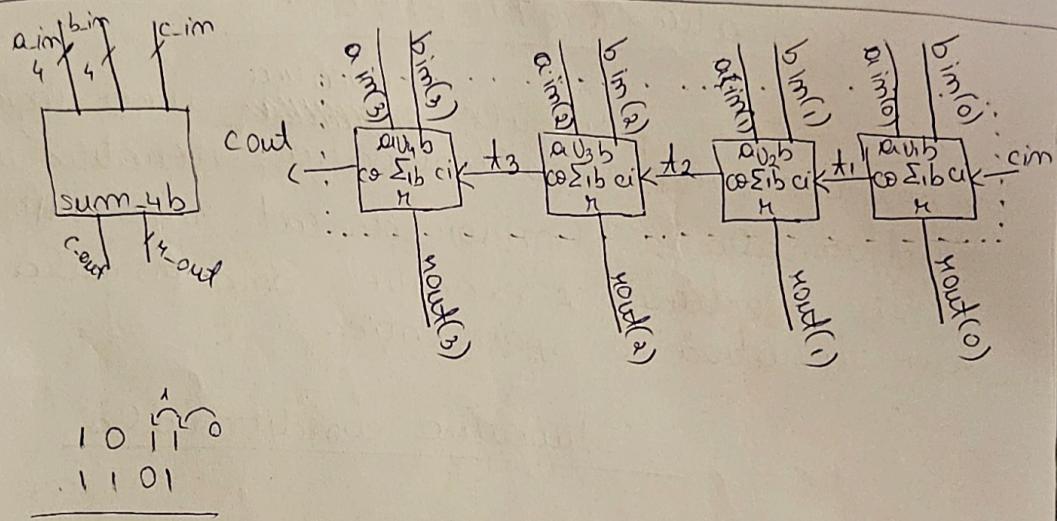
COMPONENT nume_comp
PORT (lista porturi);

END COMPONENT;

SIGNAL.....



Eticheta: nume_comp PORTMAP (lista conexiuni)



LIBRARY IEEE

USE IEEE std_logic_1164.all;

ENTITY sum_4b IS

PORT(a_im, b_im : IN std_logic_vector(3 DOWNTO 0);
cin : IN std_logic;

sum_out : OUT std_logic_vector(3 DOWNTO 0);
cout : OUT std_logic;

END sum_4b;

ARCHITECTURE desch OF sum_4b;

COMPONENT sum_1b

PORT(a, b, ci : IN std_logic;

```

    M, CO : OUT STD-LOGIC;
END COMPONENT;
SIGNAL T1, T2, T3 : STD-LOGIC;
BEGIN
    U1: SUM-1B PORT MAP(AIM(0), BIM(0), CIP, T1, HOLD(0), T1);
    U2: SUM-1B PORT MAP(AIM(1), BIM(1), T2, HOLD(1), T2);
    U3: SUM-1B PORT MAP(AIM(2), BIM(2), T2, HOLD(2), T3);
    U4: SUM-1B PORT MAP(AIM(3), BIM(3), T3, HOLD(3), Cout);
END DESCR;

```

Specificația GENERATE:

GENERATE permite ~~crearea~~ unei logici repetitive prin multiplicarea iterativă sau conditională a liniilor de cod susțină VHDL și nu afectează o execuție conditională sau iterativă a specificației.

Structura conditională

eticheta IF condiție GENERATE

- zonă descrie concurentă

END GENERATE;

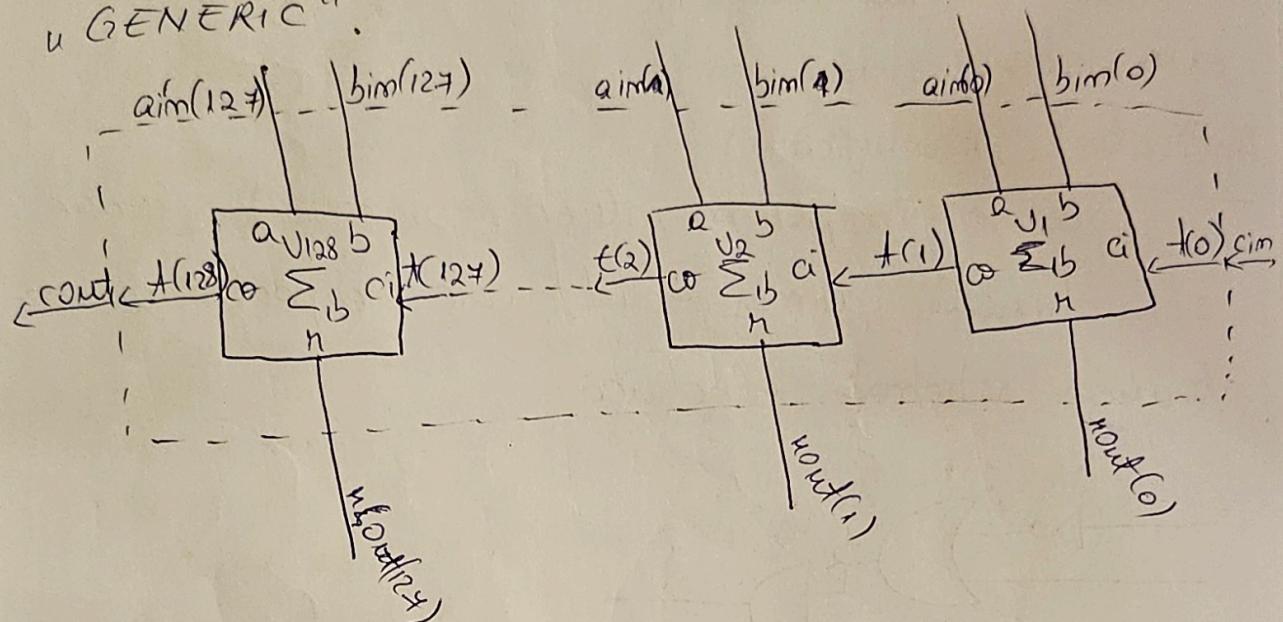
eticheta FOR parametru IN interval GENERATE

- zonă descrie concurentă

END GENERATE;

licatie:

Pentru a lucra cu sum. pe 4 biti, să se realizeze un sum. pe 128 biti folosind specificația "GENERIC", de asemenea sumaționul să fie adaptabil la operatorii de intrare și cu altă dimensiune a lastimii acestora folosind specificația "GENERIC".



LIBRARY IEEE

USE IEEE std_logic_1164.all;

ENTITY sum_128 IS

~~GENERIC~~(latmag:INTEGER:=128);
PORT(a_im,b_im:IN std_logic_vector(latmag-1 DOWNTO 0);
cim:IN std_logic;
kout:OUT std_logic_vector(latmag-1 DOWNTO 0);
cout:OUT std_logic);

END sum_128;

ARCHITECTURE desc OF sum_128 IS

COMPONENT sum_4.

PORT(a,b,ci:IN std_logic;
y,c:OUT std_logic;

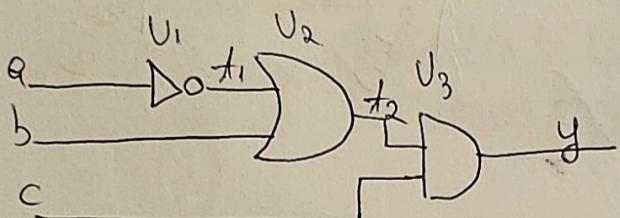
```

END COMPONENT;
SIGNAL t: std_logic_vector(latmag DOWNTO 0);
BEGIN
FOR k IN (latmag-1) GENERATE
    U: sum_1b PORTMAP (ain(k), bin(k), t(k), cout(k));
END GENERATE;
t(0) <= cin;
cout <= t(latmag);
END descr;

```

Aplicație / pt. verificare

Să se implementeze printr-o descriere structurală un circuit care respectă următoarea schemă electrică.



PAS 1: se realiză o descriere a funcției porturii logice în parte prin către un program VHDL.

```

LIBRARY IEEE;
USE IEEE std_logic_1164.all;
ENTITY inverSOR IS
    PORT (a:IN std_logic;
          y:OUT std_logic);
END inverSOR;
ARCHITECTURE descr OF inverSOR IS

```

BEGIN

$y \leftarrow \text{NOT } Q;$

END descr;

LIBRARY IEEE

USE IEEE std_logic-1164.all;

ENTITY poarte_sau IS

PORT (a, b: IN std_logic;
 y : OUT std_logic);

[SAU]

END poarte_sau;

ARCHITECTURE descr OF poarte_sau IS

BEGIN

$y \leftarrow a \text{ OR } b;$

END descr;

(uma sub alta)

LIBRARY IEEE

USE IEEE std_logic-1164.all;

ENTITY poarte_si IS

PORT (a, b: IN std_logic;
 y : OUT std_logic);

[Si]

END poarte_si;

ARCHITECTURE descr OF poarte_si IS

BEGIN

$y \leftarrow a \text{ AND } b;$

END descr;

PAS 2^a de creare modul digital prim utilizat
compom. descrie la punctul 1 si constarea
acestui după schema electrică din
figură.

LIBRARY IEEE;

USE IEEE std_logic_1164.all;

ENTITY circuit IS

PORT (a, b, c: IN std_logic;
y: OUT std_logic);

END circuit;

ARCHITECTURE desc OF circuit IS

COMPONENT inverson

PORT (a: IN std_logic;
y: OUT std_logic);

END COMPONENT;

COMPONENT paralelo_sau

PORT (a, b: IN std_logic;
y: OUT std_logic);

END ~~paralelo~~ COMPONENT;

COMPONENT paralelo_si

PORT (a, b: IN std_logic;
y: OUT std_logic);

END COMPONENT;

SIGNAL t1, t2: std_logic;

BEGIN

U1: inverson PORTMAP (a, t1);

U2: paralelo_sau PORTMAP (t1, b, t2);

U3: paralelo_si PORTMAP (t2, c, y);

END desc;