تصميم النظم المنطقية باستخدام الدارات المنطقية المبرمجة

المحاضرة التاسعة

د.م. خولة حموي
khawla.hamwi@gmail.com

العام الدراسي: 2023-2024

عناوين المحاضرة

- النمذجة البنيوية Structural modeling

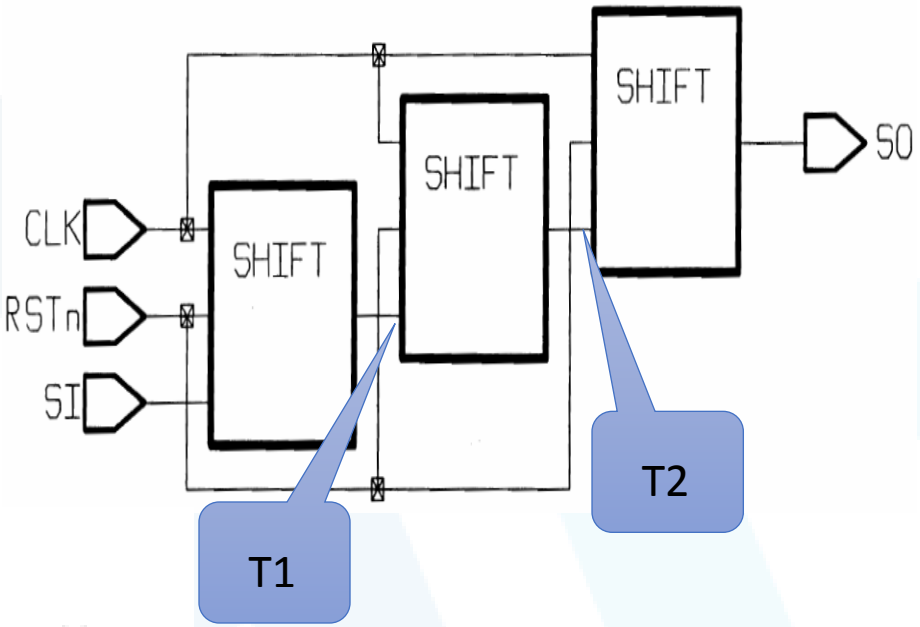
1. الحزم Packages
2. المكون Component
3. عبارة Port Map
4. عبارة Generic Map

أمثلة Examples

أمثلة

مسجل إزاحة 24 بت باستخدام ثلاثة مسجلات إزاحة 8 بت

```
1   entity SHIFT24 is
2      port (
3         RSTn, CLK, SI : in  bit;
4         SO            : out bit);
5   end SHIFT24;
6   architecture RTL5 of SHIFT24 is
7      component SHIFT
8      port (
9         RSTn, CLK, SI : in  bit;
10        SO            : out bit);
11     end component;
12     signal T1, T2 : bit;
13  begin
14     stage2 : SHIFT
15        port map (RSTn => RSTn, CLK => CLK, SI => SI, SO => T1);
16     stage1 : SHIFT
17        port map (RSTn => RSTn, CLK => CLK, SI => T1, SO => T2);
18     stage0 : SHIFT
19        port map (RSTn => RSTn, CLK => CLK, SI => T2, SO => SO);
20  end RTL5;
```

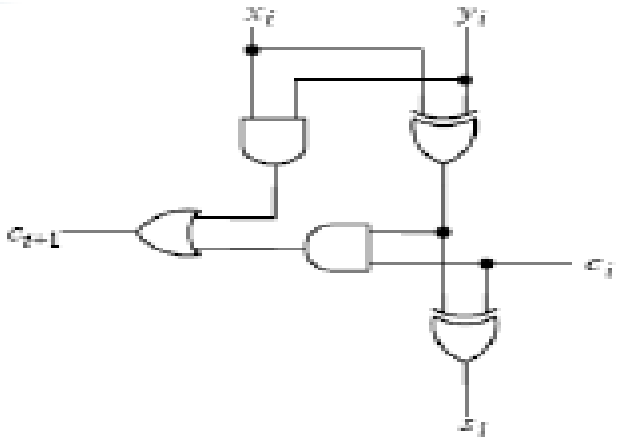


أمثلة

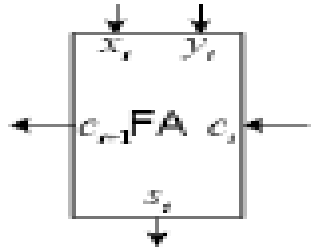
دارة جامع كامل

x_i	y_i	c_i	c_{i-1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

(a)



(b)



(c)

Full adder: (a) truth table; (b) circuit; (c) logic symbol.

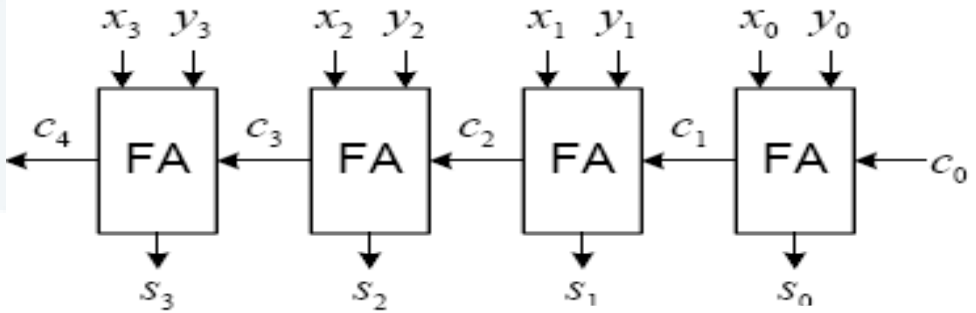
$$S = x \oplus y \oplus c_{in}$$
$$C_{out} = xy + c_{in}(x \oplus y)$$

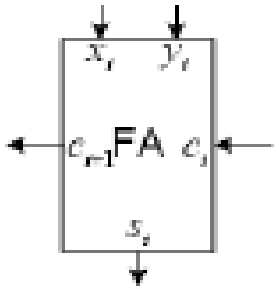
```
ENTITY FA IS PORT (
   ci, xi, yi: IN BIT;
   co, si: OUT BIT);
END FA;

ARCHITECTURE Dataflow OF FA IS
BEGIN
   co <= (xi AND yi) OR (ci AND (xi XOR yi));
   si <= xi XOR yi XOR ci;
END Dataflow;
```

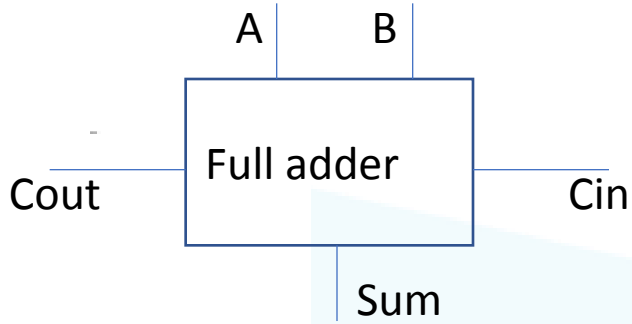
Dataflow VHDL code for a 1-bit full adder.

أمثلة

دارة جامع كامل





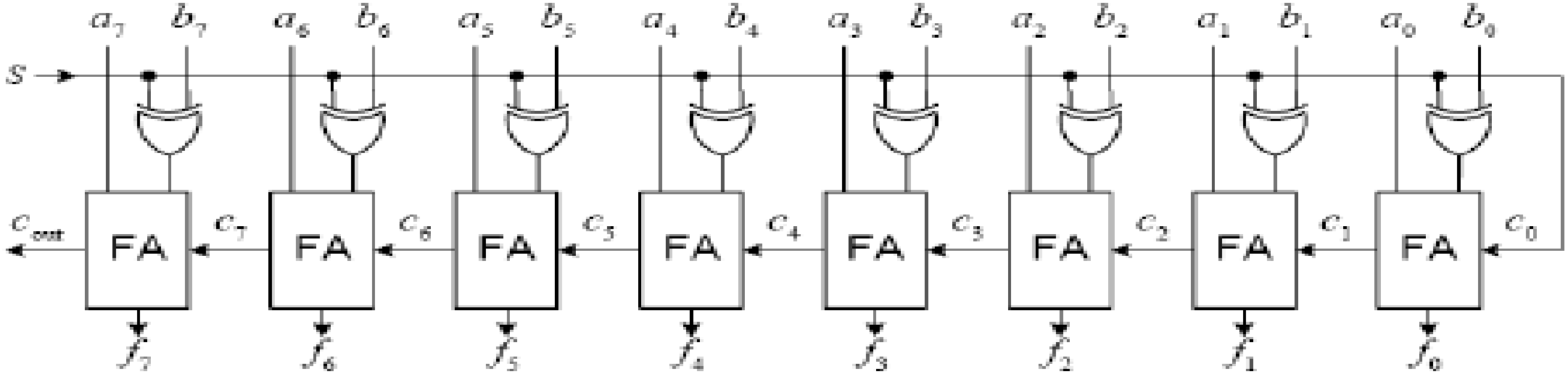



```
ENTITY Adder4 IS
    PORT (Cin: IN BIT;
          A, B: IN BIT_VECTOR(3 DOWNTO 0);
          Cout: OUT BIT;
          SUM: OUT BIT_VECTOR(3 DOWNTO 0));
END Adder4;
ARCHITECTURE Structural OF Adder4 IS
    COMPONENT FA  IS
          PORT (ci, xi, yi: IN BIT;
                co, si: OUT BIT);
          END COMPONENT;
    SIGNAL Carryv: BIT_VECTOR(4 DOWNTO 0);
BEGIN
    Carryv(0) <= Cin;
    Adder: FOR k IN 3 DOWNTO 0 GENERATE
          FullAdder: FA PORT MAP (Carryv(k), A(k), B(k), Carryv(k+1), SUM(k));
    END GENERATE Adder;
    Cout <= Carryv(4);
END Structural;
```

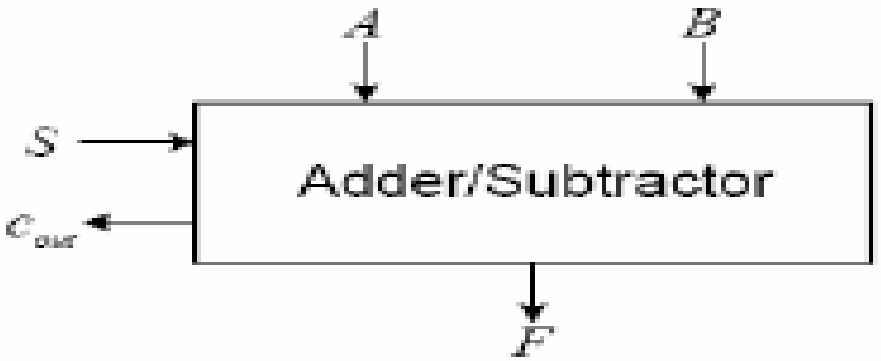
VHDL code for a 4-bit ripple-carry adder using a FOR-GENERATE statement.

أمثلة

دارة جامع / طارح



S	Function	Operation
0	Add	$F = A + B$
1	Subtract	$F = A + B' + 1$



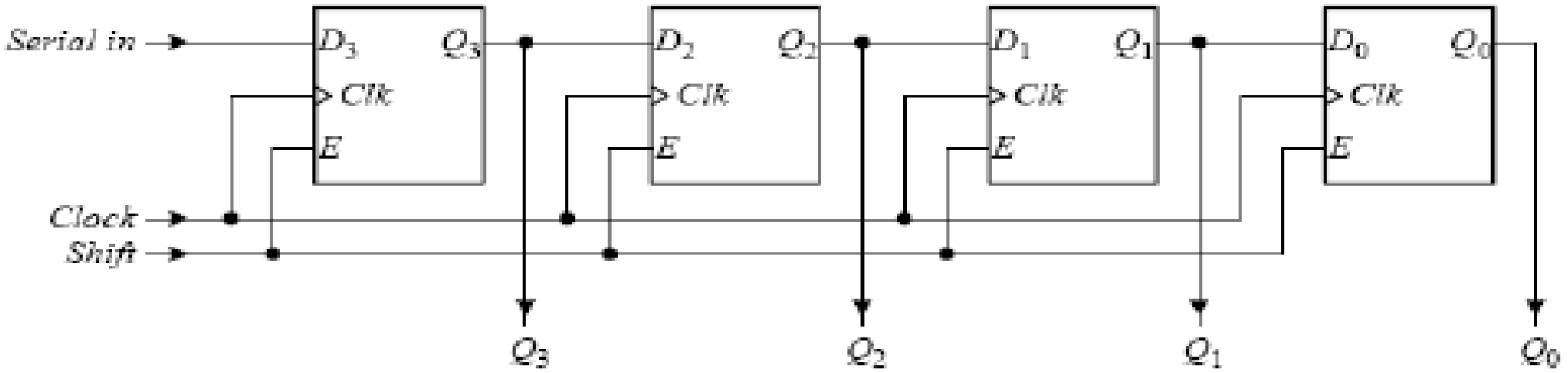
أمثلة

دارة جامع / طارح

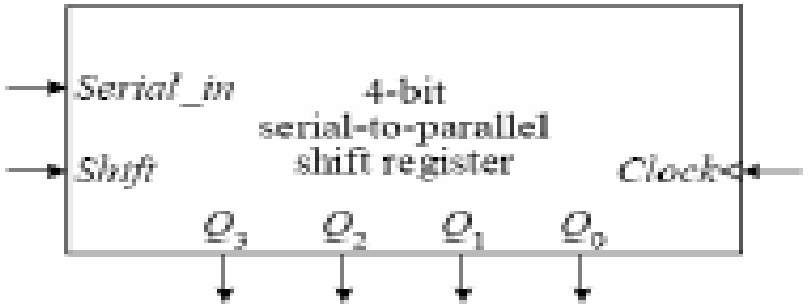
```
1.  LIBRARY ieee;
2.  USE ieee.std_logic_1164.ALL;
3.  USE ieee.std_logic_arith.ALL;
4.  USE ieee.std_logic_unsigned.ALL;
5.  ENTITY AddSub IS
6.  GENERIC(n: NATURAL :=8);                                    -- default number of bits = 8
7.          PORT(A: IN std_logic_vector(n-1 downto 0);
8.                  B: IN std_logic_vector(n-1 downto 0);
9.                  subtract: IN std_logic;
10.                 carry: OUT std_logic;
11.                 sum: OUT std_logic_vector(n-1 downto 0));
12. END AddSub;
13. ARCHITECTURE Behavioral OF AddSub IS
14.         -- temporary result with one extra bit for carry
15.         SIGNAL result: std_logic_vector(n downto 0);
16. BEGIN
17.         PROCESS(subtract, A, B)
18.         BEGIN
19.                 IF (subtract = '0') THEN                    -- addition
20.                 --add the two operands with one extra bit for carry
21.                         result <= ('0' & A)+('0' & B);
22.                         sum <= result(n-1 downto 0);        -- extract the n-bit result
23.                         carry <= result(n);                 -- extract the carry bit from result
24.                 ELSE                                        -- subtraction
25.                         result <= ('0' & A)-('0' & B);
26.                         sum <= result(n-1 downto 0);        -- extract the n-bit result
27.                         carry <= result(n);                 -- extract the borrow bit from result
28.                 END IF;
29.         END PROCESS;
30. END Behavioral;
```

أمثلة

مسجل إزاحة بدخل تسلسلي وخرج تفرعي



Shift	Operation
0	No change
1	One bit from *Serial_in* is shifted in



أمثلة

مسجل إزاحة بدخل تسلسلي وخرج تفرعي

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY ShiftReg IS
      PORT(Serialin, Clock, Shift : IN STD_LOGIC;
                                    Q : OUT STD_LOGIC_VECTOR(3 downto 0));
END ShiftReg;

ARCHITECTURE Structural OF ShiftReg IS
      SIGNAL N0, N1, N2, N3 : STD_LOGIC;
      COMPONENT D_flipflop PORT (D, Clock, E : IN STD_LOGIC;
                                               Q : OUT STD_LOGIC);
      END COMPONENT;

BEGIN
     U1: D_flipflop PORT MAP (Serialin, Clock, Shift, N3);
     U2: D_flipflop PORT MAP (N3, Clock, Shift, N2);
     U3: D_flipflop PORT MAP (N2, Clock, Shift, N1);
     U4: D_flipflop PORT MAP (N1, Clock, Shift, N0);
     Q(3) <= N3;
     Q(2) <= N2;
     Q(1) <= N1;
     Q(0) <= N0;
END Structural;
```

```
-- D flip-flop with enable
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY D_flipflop IS
     PORT(D, Clock, E : IN STD_LOGIC;
                        Q : OUT STD_LOGIC);
END D_flipflop;
ARCHITECTURE Behavior OF D_flipflop IS
BEGIN
     PROCESS(Clock)
     BEGIN
          IF Clock'EVENT AND Clock = '1' THEN
               IF E = '1' THEN
                    Q <= D;
               END IF;
          END IF;
     END PROCESS;
END Behavior
```