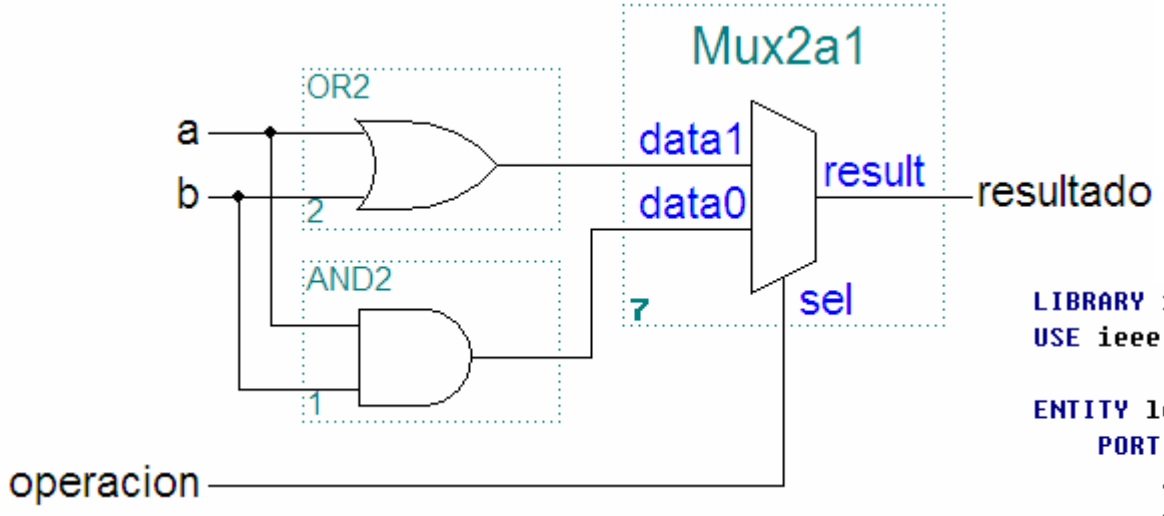
Una Unidad Lógica Aritmética en VHDL

Ing. Arturo Miguel de Priego

arturo@tourdigital.net

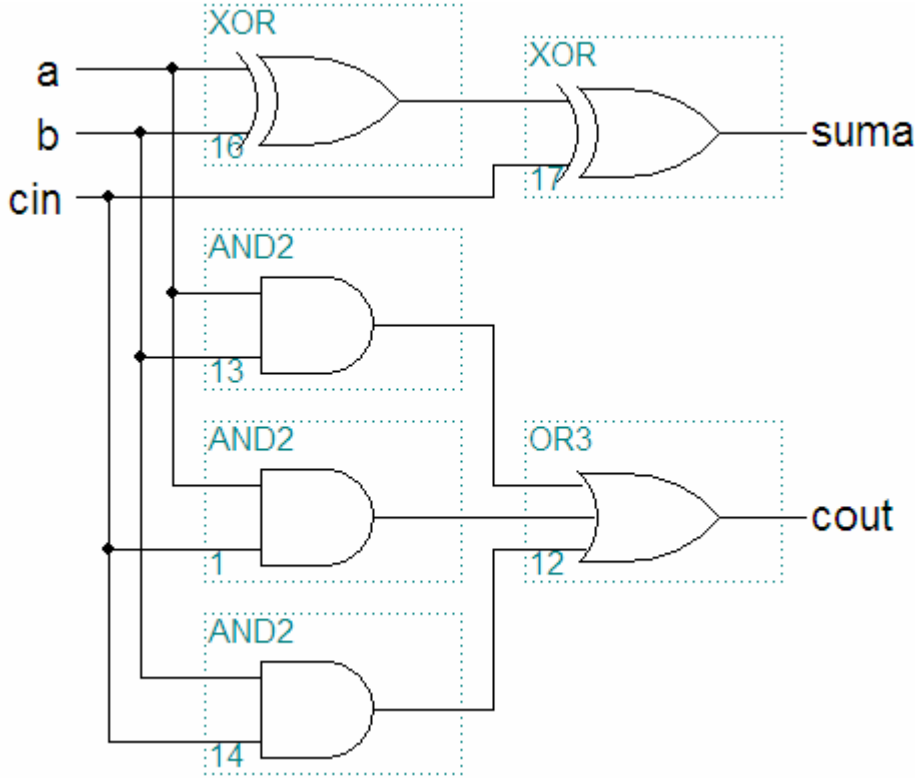
Chincha, Perú
Mayo de 2008

Unidad Lógica de 1 bit




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

ENTITY logica IS
    PORT(
        a, b,
        operacion   : IN    STD_LOGIC;
        resultado   : OUT   STD_LOGIC
    );
END logica;


ARCHITECTURE funcional OF logica IS
BEGIN

    WITH operacion SELECT
        resultado <= a AND b    WHEN '0',
                     a OR b     WHEN OTHERS;

END funcional;
```

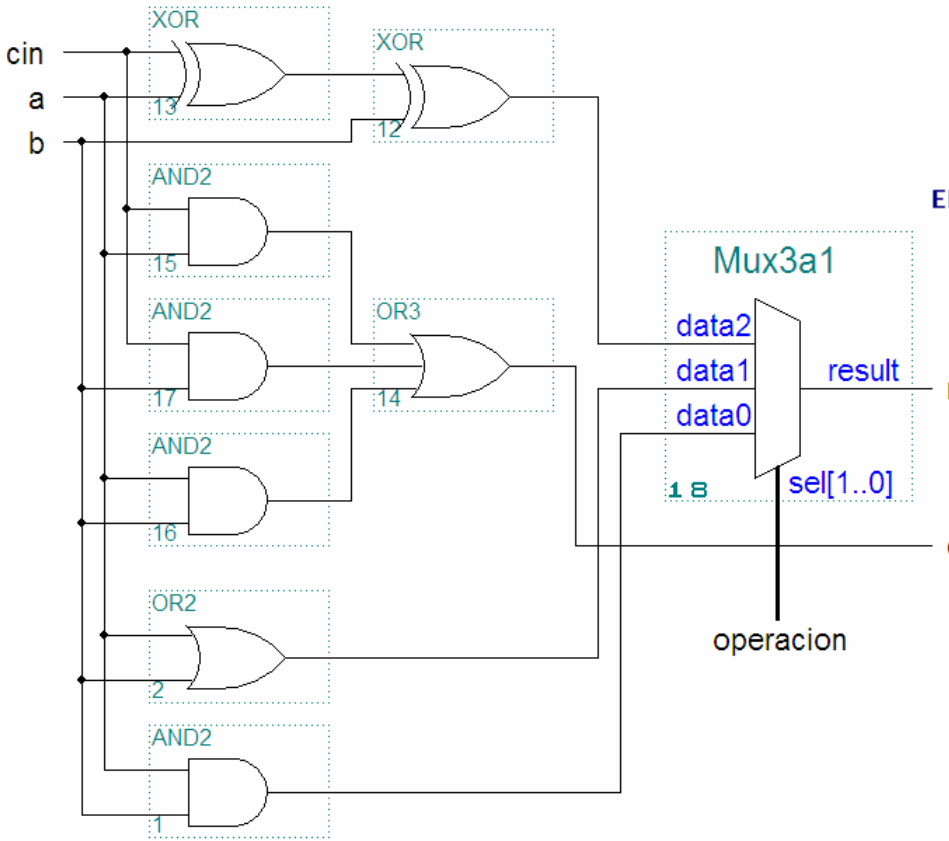
Unidad Aritmética de 1 bit




```
ENTITY Sumador1bit IS
    PORT(
        a, b,
        cin     : IN    STD_LOGIC;
        cout,
        suma    : OUT   STD_LOGIC
    );
END Sumador1bit;
```

```
ARCHITECTURE booleana OF Sumador1bit IS
BEGIN

    suma <= a XOR b XOR cin;

    cout <= (a AND b) OR (a AND cin) OR (b AND cin);

END booleana;
```

Unidad Lógico–Aritmética de 1 bit

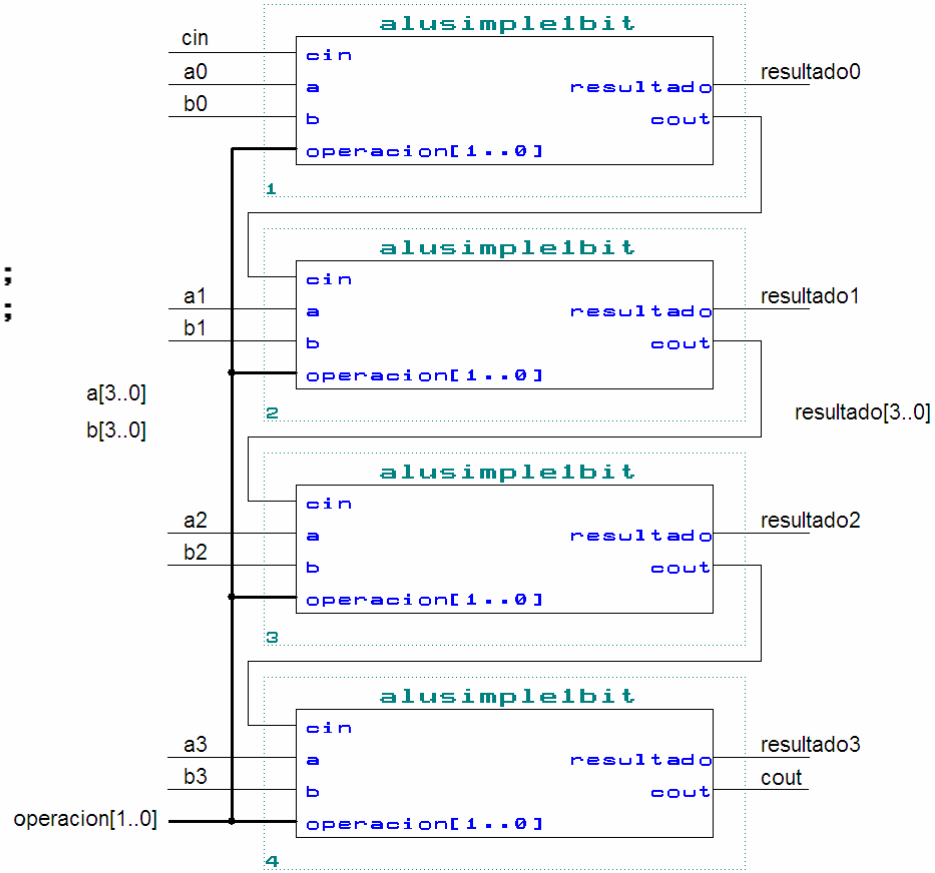



```
ENTITY ALUsimple1bit IS
    PORT(
        a, b,
        cin          : IN    STD_LOGIC;
        operacion    : IN    STD_LOGIC_VECTOR(1 DOWNTO 0);
        cout,
        resultado    : OUT   STD_LOGIC
    );
END ALUsimple1bit;
```

```
ARCHITECTURE funcional OF ALUsimple1bit IS
BEGIN

    WITH operacion SELECT
        resultado <= a AND b            WHEN "00",
                     a OR b             WHEN "01",
                     a XOR b XOR cin    WHEN "10",
                     '0'                WHEN OTHERS;

        cout <= (a AND b) OR (a AND cin) OR (b AND cin);

END funcional;
```

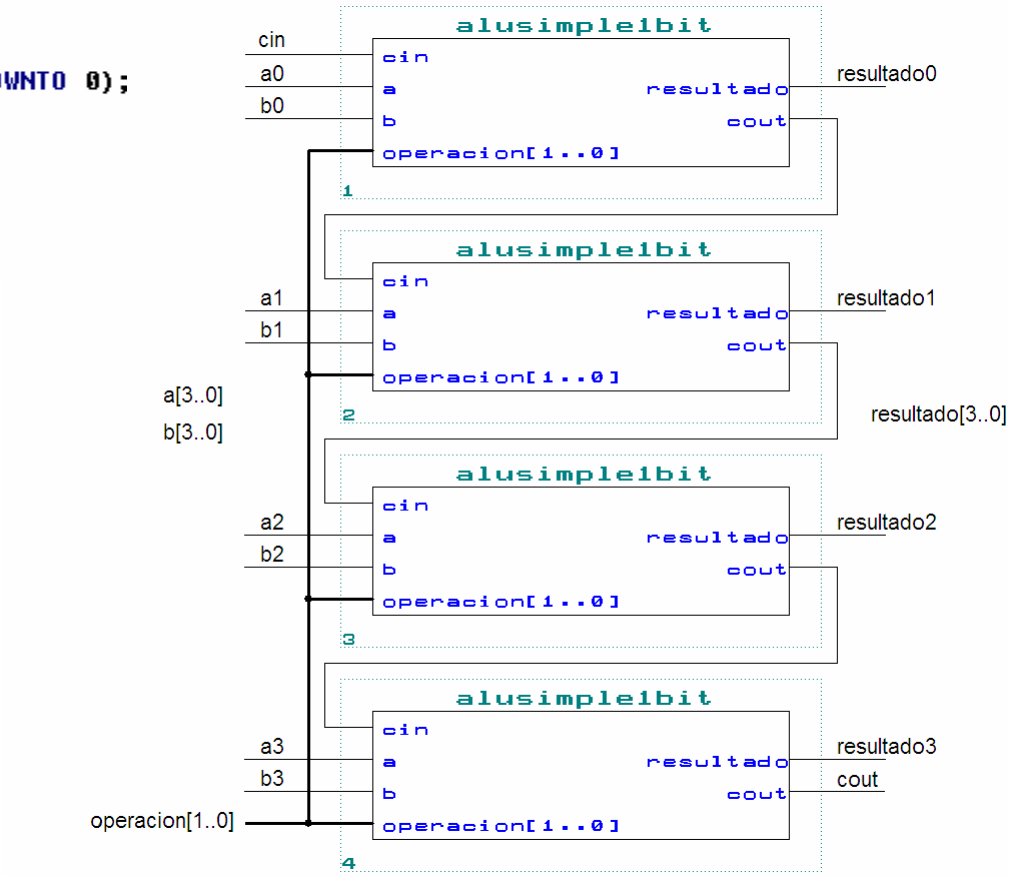
Unidad Lógico–Aritmética de 4 bits

```
ENTITY ALUsimple4bits IS
    PORT(
        cin          : IN    STD_LOGIC;
        a, b         : IN    STD_LOGIC_VECTOR(3 DOWNTO 0);
        operacion    : IN    STD_LOGIC_VECTOR(1 DOWNTO 0);
        cout         : OUT   STD_LOGIC;
        resultado    : OUT   STD_LOGIC_VECTOR(3 DOWNTO 0)
    );
END ALUsimple4bits;
```



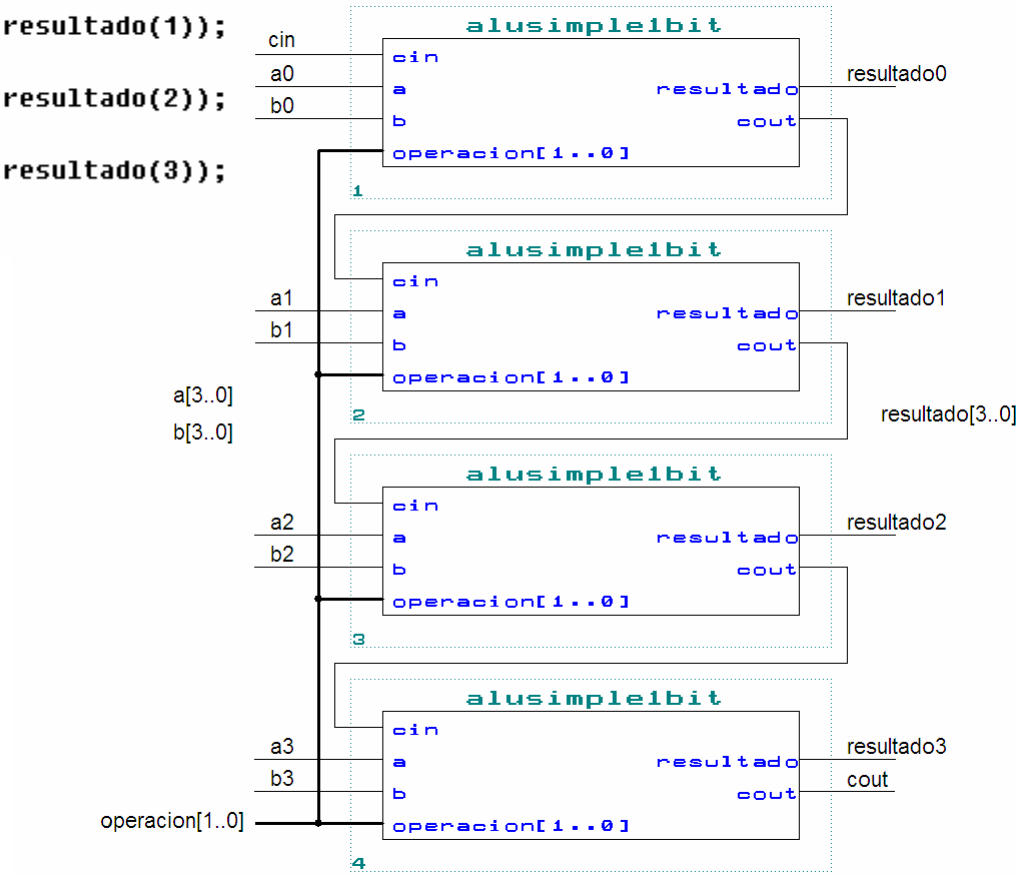
Unidad Lógico–Aritmética de 4 bits

```
ARCHITECTURE estructural1 OF ALUsimple4bits IS

    COMPONENT ALUsimple1bit
        PORT(
        a, b,
        cin          : IN    STD_LOGIC;
        operacion    : IN    STD_LOGIC_VECTOR(1 DOWNTO 0);
        cout,
        resultado    : OUT   STD_LOGIC);
    END COMPONENT;

    SIGNAL c: STD_LOGIC_VECTOR(1 TO 3);
```



Unidad Lógico–Aritmética de 4 bits

```
BEGIN

    U0: ALUsimple1bit
        PORT MAP (a(0), b(0),  cin, operacion, c(1), resultado(0));
    U1: ALUsimple1bit
        PORT MAP (a(1), b(1), c(1), operacion, c(2), resultado(1));
    U2: ALUsimple1bit
        PORT MAP (a(2), b(2), c(2), operacion, c(3), resultado(2));
    U3: ALUsimple1bit
        PORT MAP (a(3), b(3), c(3), operacion, cout, resultado(3));

END estructural1;
```



Unidad Lógico–Aritmética de 4 bits

```
    SIGNAL c: STD_LOGIC_VECTOR(1 TO 3);
BEGIN

    U0: ALUsimple1bit
        PORT MAP (cin,  a(0), b(0), operacion, resultado(0), c(1));
    U1: ALUsimple1bit
        PORT MAP (c(1), a(1), b(1), operacion, resultado(1), c(2));
    U2: ALUsimple1bit
        PORT MAP (c(2), a(2), b(2), operacion, resultado(2), c(3));
    U3: ALUsimple1bit
        PORT MAP (c(3), a(3), b(3), operacion, resultado(3), cout);

END estructural1;
```

```
    SIGNAL c: STD_LOGIC_VECTOR(0 TO 4);
BEGIN

    c(0) <= cin;
    cout <= c(4);

    U0: ALUsimple1bit
        PORT MAP (c(0), a(0), b(0), operacion, resultado(0), c(1));
    U1: ALUsimple1bit
        PORT MAP (c(1), a(1), b(1), operacion, resultado(1), c(2));
    U2: ALUsimple1bit
        PORT MAP (c(2), a(2), b(2), operacion, resultado(2), c(3));
    U3: ALUsimple1bit
        PORT MAP (c(3), a(3), b(3), operacion, resultado(3), c(4));

END estructural2;
```

Unidad Lógico–Aritmética de 4 bits

```
    SIGNAL c: STD_LOGIC_VECTOR(0 TO 4);
BEGIN

    c(0) <= cin;
    cout <= c(4);

    U0: ALUsimple1bit
        PORT MAP (c(0), a(0), b(0), operacion, resultado(0), c(1));
    U1: ALUsimple1bit
        PORT MAP (c(1), a(1), b(1), operacion, resultado(1), c(2));
    U2: ALUsimple1bit
        PORT MAP (c(2), a(2), b(2), operacion, resultado(2), c(3));
    U3: ALUsimple1bit
        PORT MAP (c(3), a(3), b(3), operacion, resultado(3), c(4));

END estructural2;
```

```
    SIGNAL c: STD_LOGIC_VECTOR(0 TO 4);
BEGIN

    c(0) <= cin;
    cout <= c(4);

    Cascada:
    FOR i IN 0 TO 3 GENERATE
        U: ALUsimple1bit
            PORT MAP (c(i), a(i), b(i), operacion, resultado(i), c(i + 1));
    END GENERATE;

END estructural;
```

Unidad Lógico–Aritmética de N bits

```
    SIGNAL c: STD_LOGIC_VECTOR(0 TO 4);
BEGIN

    c(0) <= cin;
    cout <= c(4);

    Cascada:
    FOR i IN 0 TO 3 GENERATE
        U: ALUsimple1bit
            PORT MAP (c(i), a(i), b(i), operacion, resultado(i), c(i + 1));
    END GENERATE;

END estructural;
```

```
    SIGNAL c: STD_LOGIC_VECTOR(0 TO N);
BEGIN

    c(0) <= cin;
    cout <= c(N);

    Cascada:
    FOR i IN 0 TO N - 1 GENERATE
        U: ALUsimple1bit
            PORT MAP (c(i), a(i), b(i), operacion, resultado(i), c(i + 1));
    END GENERATE;

END estructural;
```

Unidad Lógico–Aritmética de N bits

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

ENTITY ALUsimpleNbits IS
    GENERIC (N : INTEGER:= 16);
    PORT(
        cin         : IN    STD_LOGIC;
        a, b        : IN    STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
        operacion   : IN    STD_LOGIC_VECTOR(1 DOWNTO 0);
        resultado   : OUT   STD_LOGIC_VECTOR(N - 1 DOWNTO 0);
        cout        : OUT   STD_LOGIC
    );
END ALUsimpleNbits;

ARCHITECTURE estructural OF ALUsimpleNbits IS

    COMPONENT ALUsimple1bit
        PORT(
            cin,
            a, b        : IN    STD_LOGIC;
            operacion   : IN    STD_LOGIC_VECTOR(1 DOWNTO 0);
            resultado,
            cout        : OUT   STD_LOGIC);
    END COMPONENT;

    SIGNAL c: STD_LOGIC_VECTOR(0 TO N);
BEGIN

    c(0) <= cin;
    cout <= c(N);

    Cascada:
    FOR i IN 0 TO N - 1 GENERATE
        U: ALUsimple1bit
            PORT MAP (c(i), a(i), b(i), operacion, resultado(i), c(i + 1));
    END GENERATE;

END estructural;
```