Rubén Comerón Galán

Práctica 4-Sumador RCA

El modelo VHDL del sumador compuesto por semisumadores es:

```
ENTITY RCA IS
	GENERIC (n: INTEGER:=8);
	PORT(x_in, y_in: IN BIT_VECTOR(n-1 downto 0); c_in: IN BIT; sal: OUT BIT_VECTOR(n-1 downto 0); c_out: OUT BIT);
END RCA;

ARCHITECTURE estructural OF RCA IS
	COMPONENT sumador
		PORT (x, y, c_in: IN BIT; s, c_out:OUT BIT);
	END COMPONENT;

	FOR primero, intermedios, ultimos: sumador USE ENTITY WORK.sumador(estructural);
	SIGNAL c_inside: BIT_VECTOR(n-1 downto 0);
BEGIN
	G: FOR i IN 0 TO n-1 GENERATE
		lsb: IF i = 0 GENERATE
			primero: sumador PORT MAP(x_in(0), y_in(0), c_in, sal(0), c_inside(0));
		END GENERATE lsb;

		siguientes: IF i > 0 AND i < n-1 GENERATE
			intermedios: sumador PORT MAP(x_in(i), y_in(i), c_inside(i-1), sal(i), c_inside(i));
		END GENERATE siguientes;

		msb: IF i = n-1 GENERATE
			ultimos: sumador PORT MAP(x_in(i), y_in(i), c_inside(i-1), sal(i), c_out);
		END GENERATE msb;
	END GENERATE G;
END estructural;
```

El modelo VHDL del sumador compuesto por suma de productos es:

```
ENTITY RCA IS
	GENERIC (n: INTEGER:=8);
	PORT(x_in, y_in: IN BIT_VECTOR(n-1 downto 0); c_in: IN BIT; sal: OUT BIT_VECTOR(n-1 downto 0); c_out: OUT BIT);
END ENTITY;

ARCHITECTURE estructural OF RCA IS
	COMPONENT sumador
```

```
		PORT (x, y, c_in: IN BIT; s, c_out:OUT BIT);
	END COMPONENT;
	COMPONENT and3
		PORT (e1, e2, e3: IN BIT; sal: OUT BIT);
	END COMPONENT;
	COMPONENT or4
		PORT (e1, e2, e3, e4: IN BIT; sal: OUT BIT);
	END COMPONENT;
	COMPONENT not2
		PORT (e: IN BIT; sal: OUT BIT);
	END COMPONENT;
	SIGNAL c_aux: BIT_VECTOR(n-1 downto 0);
BEGIN
	G: FOR i IN 0 TO n-1 GENERATE
		lsb: IF i = 0 GENERATE
			primeros: sumador PORT MAP(x_in(0), y_in(0), c_in, sal(0), c_aux(0));
		END GENERATE lsb;

		siguientes: IF i > 0 AND i < n-1 GENERATE
			intermedios: sumador PORT MAP(x_in(i), y_in(i), c_aux(i-1), sal(i),
c_aux(i));
		END GENERATE siguientes;

		msb: IF i = n-1 GENERATE
			primeros: sumador PORT MAP(x_in(i), y_in(i), c_aux(i-1), sal(i), c_out);
		END GENERATE msb;
	END GENERATE G;
END estructural;
```

EL test-bench utilizado para ambos sumadores es:

```
ENTITY tb_RCA IS
END tb_RCA;

ARCHITECTURE estructural OF tb_RCA IS
	COMPONENT RCA
		GENERIC (n: INTEGER:=8);
		PORT(x_in, y_in: IN BIT_VECTOR(n-1 downto 0);
		    c_in: IN BIT;
		    sal: OUT BIT_VECTOR(n-1 downto 0);
		    c_out: OUT BIT);
	END COMPONENT;

	FOR componente: RCA USE ENTITY WORK.RCA(estructural);

	CONSTANT TAMANO_PALABRA: integer := 2;

	SIGNAL a, b, r: BIT_VECTOR(TAMANO_PALABRA-1 downto 0);
```

```
        SIGNAL cin, cout, marca: BIT;

BEGIN

        componente: RCA GENERIC MAP (TAMANO_PALABRA) PORT MAP (A, B, Cin, R,
Cout);

        PROCESS
        VARIABLE valor, i, j, n: INTEGER;
        BEGIN

                valor:=0;
                i:=0;
                j:=0;
                n:=0;
                A<=(others=>'0');
                B<=(others=>'0');
                Cin<='0';
                marca<='0';
                WAIT FOR TAMANO_PALABRA*10 ns;
                A<=(others=>'1');
                Cin<='1';
                marca<='1';
                WAIT UNTIL Cout='1' AND R(TAMANO_PALABRA-1)='0';
                marca<='0';


                FOR k IN A'RANGE LOOP
                        IF A(k)='1' THEN
                                i:=i+2**k;
                        END IF;
                END LOOP;
                FOR k IN B'RANGE LOOP
                        IF B(k)='1' THEN
                                j:=j+2**k;
                        END IF;
                END LOOP;
                IF Cin='1' THEN
                        valor:=valor+1;
                        n:=1;
                ELSE
                        n:=0;
                END IF;
                FOR k IN R'RANGE LOOP
                        IF R(k)='1' THEN
                                valor:=valor+2**k;
                        END IF;
                END LOOP;

                IF Cout='1' THEN
                        valor:=valor+2**TAMANO_PALABRA;
                END IF;
```

```
        ASSERT valor=i+j REPORT "resultado incorrecto" SEVERITY ERROR;
        WAIT FOR 2 ns;
        WAIT;

    END PROCESS;
END estructural;
```

	Semisumadores	Suma de productos
RTL	$11r_{RTL}$	$9r_{RTL}$
ECL	$5r_{ECL}$	$5r_{ECL}$
TTL	$11r_{TTL}$	$9r_{TTL}$
CMOS	$10r_{CMOS}$	$9r_{CMOS}$