Rubén Comerón Galán

# Sesión 7-NEG-C2

Activo a partir del primer ‘1’:

Bloque negador:

```
ENTITY Bloque_neg IS
        PORT(x, e, c_in: IN BIT; r, c_out: OUT BIT);
END ENTITY;

ARCHITECTURE estructural OF Bloque_neg IS

COMPONENT and2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
COMPONENT or2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
COMPONENT xor2 IS
        GENERIC (retardo: TIME:= 3 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;

SIGNAL e_and_c: BIT;

BEGIN

        c_and_e: and2 PORT MAP(e, c_in, e_and_c);
        salida: xor2 PORT MAP(x, e_and_c, r);
        siguiente_c: or2 PORT MAP(x, c_in, c_out);

END estructural;
```

Negador:

```
ENTITY NEG_C2 IS
```

```
	GENERIC(n:INTEGER :=16);
	PORT(x, e: IN BIT_VECTOR(n-1 DOWNTO 0); r: OUT BIT_VECTOR(n-1 DOWNTO 0);
des: OUT BIT);
END ENTITY;

ARCHITECTURE estructural OF NEG_C2 IS

COMPONENT Bloque_neg IS
	PORT(x, e, c_in: IN BIT; r, c_out: OUT BIT);
END COMPONENT;
COMPONENT not2 IS
	GENERIC (retardo: TIME:= 1 ns);
	PORT (e: IN BIT; sal: OUT BIT);
END COMPONENT;
COMPONENT and2 IS
	GENERIC (retardo: TIME:= 1 ns);
	PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
SIGNAL c: BIT_VECTOR(n-1 DOWNTO 0);
SIGNAL d: BIT;

BEGIN

	G: FOR i IN 0 TO n-1 GENERATE
		primero: IF i = 0 GENERATE
			primer_bit: Bloque_neg PORT MAP(x(i), e(i), x(i), r(i), c(i));
		END GENERATE primero;
		intermedios: IF i > 0 GENERATE
			siguientes: Bloque_neg PORT MAP(x(i), e(i), c(i-1), r(i), c(i));
		END GENERATE intermedios;
	END GENERATE G;
	desbor: not2 PORT MAP(c(n-2), d)
	desbordamiento: and2 PORT MAP(x(n-1), d, des);


END estructural;
```

A partir del primer ‘0’:

Bloque negador:

```
ENTITY Bloque_neg IS
	PORT(x, e, c_in: IN BIT; r, c_out: OUT BIT);
```

```
END ENTITY;

ARCHITECTURE estructural OF Bloque_neg IS

COMPONENT and2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
COMPONENT or2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
COMPONENT xor2 IS
        GENERIC (retardo: TIME:= 3 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
COMPONENT not2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e: IN BIT; sal: OUT BIT);
END COMPONENT;

SIGNAL e_and_c, x_negada: BIT;

BEGIN

        neg_x: not2 PORT MAP(x, x_negada);
        c_and_e: and2 PORT MAP(e, c_in, e_and_c);
        salida: xor2 PORT MAP(x, e_and_c, r);
        siguiente_c: or2 PORT MAP(x_negada, c_in, c_out);

END estructural;
```

Negador:

```
ENTITY NEG_C2 IS
        GENERIC(n:INTEGER :=16);
        PORT(x, e: IN BIT_VECTOR(n-1 DOWNTO 0); r: OUT BIT_VECTOR(n-1 DOWNTO 0);
des: OUT BIT);
END ENTITY;

ARCHITECTURE estructural OF NEG_C2 IS

COMPONENT Bloque_neg IS
        PORT(x, e, c_in: IN BIT; r, c_out: OUT BIT);
END COMPONENT;
COMPONENT not2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e: IN BIT; sal: OUT BIT);
```

```
END COMPONENT;
COMPONENT or2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
SIGNAL c, cal_d: BIT_VECTOR(n-1 DOWNTO 0);
SIGNAL d, de, x_neg_0: BIT;

BEGIN

        x_neg: not2 PORT MAP(x(0), x_neg_0);
        G: FOR i IN 0 TO n-1 GENERATE
                primero: IF i = 0 GENERATE
                        primer_bit: Bloque_neg PORT MAP(x(i), e(i), x_neg_0, r(i), c(i));
                END GENERATE primero;
                intermedios: IF i > 0 GENERATE
                        siguientes: Bloque_neg PORT MAP(x(i), e(i), c(i-1), r(i), c(i));
                END GENERATE intermedios;
        END GENERATE G;
        B: FOR i IN 1 TO n-2 GENERATE
                primer: IF i = 1 GENERATE
                        primeros_bit: or2 PORT MAP(x(i), x(i-1), cal_d(i));
                END GENERATE primer;
                intermedio: IF i > 1 AND i < n-2 GENERATE
                        siguiente: or2 PORT MAP(x(i), cal_d(i-1), cal_d(i));
                END GENERATE intermedio;
                final: IF i = n-2 GENERATE
                        ultimo: or2 PORT MAP(x(i), cal_d(i-1), cal_d(i));
                END GENERATE final;
        END GENERATE B;
        d_negada: not2 PORT MAP(x(n-1), d);
        desborde:  or2 PORT MAP(d, cal_d(n-2), de);
        desbor:  not2 PORT MAP(de, des);


END estructural;
```

Hasta primer ‘1’:

Bloque negador:

```
ENTITY Bloque_neg IS
        PORT(x, e, c_in: IN BIT; r, c_out: OUT BIT);
END ENTITY;
```

```
ARCHITECTURE estructural OF Bloque_neg IS

COMPONENT and2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
COMPONENT or2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
COMPONENT xor2 IS
        GENERIC (retardo: TIME:= 3 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
COMPONENT not2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e: IN BIT; sal: OUT BIT);
END COMPONENT;

SIGNAL e_and_c, x_negada: BIT;

BEGIN

        neg_x: not2 PORT MAP(x, x_negada);
        c_and_e: and2 PORT MAP(e, c_in, e_and_c);
        salida: xor2 PORT MAP(x, e_and_c, r);
        siguiente_c: and2 PORT MAP(x_negada, c_in, c_out);

END estructural;
```

Negador:

```
ENTITY NEG_C2 IS
        GENERIC(n:INTEGER :=16);
        PORT(x, e: IN BIT_VECTOR(n-1 DOWNTO 0); r: OUT BIT_VECTOR(n-1 DOWNTO 0);
des: OUT BIT);
END ENTITY;

ARCHITECTURE estructural OF NEG_C2 IS

COMPONENT Bloque_neg IS
        PORT(x, e, c_in: IN BIT; r, c_out: OUT BIT);
END COMPONENT;
COMPONENT and2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
```

```
COMPONENT not2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e: IN BIT; sal: OUT BIT);
END COMPONENT;
COMPONENT or2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
SIGNAL c: BIT_VECTOR(n-1 DOWNTO 0);
SIGNAL x_neg_0: BIT;

BEGIN

        x_neg: not2 PORT MAP(x(0), x_neg_0);
        G: FOR i IN 0 TO n-1 GENERATE
                primero: IF i = 0 GENERATE
                        primer_bit: Bloque_neg PORT MAP(x(i), e(i), x_neg_0, r(i), c(i));
                END GENERATE primero;
                intermedios: IF i > 0 GENERATE
                        siguientes: Bloque_neg PORT MAP(x(i), e(i), c(i-1), r(i), c(i));
                END GENERATE intermedios;
        END GENERATE G;
        desborde: and2 PORT MAP(x(n-1), c(n-2), des);

END estructural;
```

Hasta primer ‘0’:

Bloque negador:

```
ENTITY Bloque_neg IS
        PORT(x, e, c_in: IN BIT; r, c_out: OUT BIT);
END ENTITY;

ARCHITECTURE estructural OF Bloque_neg IS

COMPONENT and2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
COMPONENT or2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
```

```
END COMPONENT;
COMPONENT xor2 IS
        GENERIC (retardo: TIME:= 3 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;

SIGNAL e_and_c: BIT;

BEGIN

        c_and_e: and2 PORT MAP(e, c_in, e_and_c);
        salida: xor2 PORT MAP(x, e_and_c, r);
        siguiente_c: and2 PORT MAP(x, c_in, c_out);

END estructural;
```

Negador:

```
ENTITY NEG_C2 IS
        GENERIC(n:INTEGER :=16);
        PORT(x, e: IN BIT_VECTOR(n-1 DOWNTO 0); r: OUT BIT_VECTOR(n-1 DOWNTO 0);
des: OUT BIT);
END ENTITY;

ARCHITECTURE estructural OF NEG_C2 IS

COMPONENT Bloque_neg IS
        PORT(x, e, c_in: IN BIT; r, c_out: OUT BIT);
END COMPONENT;
COMPONENT not2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e: IN BIT; sal: OUT BIT);
END COMPONENT;
COMPONENT or2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
COMPONENT and2 IS
        GENERIC (retardo: TIME:= 1 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END COMPONENT;
SIGNAL c, cal_d: BIT_VECTOR(n-1 DOWNTO 0);
SIGNAL x_neg_c: BIT;

BEGIN

        G: FOR i IN 0 TO n-1 GENERATE
```

```
            primero: IF i = 0 GENERATE
                primer_bit: Bloque_neg PORT MAP(x(i), e(i), x(i), r(i), c(i));
            END GENERATE primero;
            intermedios: IF i > 0 GENERATE
                siguientes: Bloque_neg PORT MAP(x(i), e(i), c(i-1), r(i), c(i));
            END GENERATE intermedios;
        END GENERATE G;
        B: FOR i IN 1 TO n-2 GENERATE
            primer: IF i = 1 GENERATE
                primeros_bit: or2 PORT MAP(x(i), x(i-1), cal_d(i));
            END GENERATE primer;
            intermedio: IF i > 1 AND i < n-2 GENERATE
                siguiente: or2 PORT MAP(x(i), cal_d(i-1), cal_d(i));
            END GENERATE intermedio;
            final: IF i = n-2 GENERATE
                ultimo: or2 PORT MAP(x(i), cal_d(i-1), cal_d(i));
            END GENERATE final;
        END GENERATE B;
        neg_x_c: not2 PORT MAP(cal_d(n-2), x_neg_c);
        desbordamiento: and2 PORT MAP(x_neg_c, x(n-1), des);

END estructural;
```

El test-bench es:

```
ENTITY tb_NEG_C2 IS
END tb_NEG_C2;

ARCHITECTURE estructural OF tb_NEG_C2 IS
        COMPONENT NEG_C2 IS
            GENERIC (n: INTEGER:=4);
            PORT(x, e: IN BIT_VECTOR(n-1 downto 0);
                r: OUT BIT_VECTOR(n-1 downto 0);
                des: OUT BIT);
        END COMPONENT;

        FOR componente: NEG_C2 USE ENTITY WORK.NEG_C2(estructural);

        CONSTANT TAMANO_PALABRA: integer := 4;

        SIGNAL a, b, r: BIT_VECTOR(TAMANO_PALABRA-1 downto 0);
        SIGNAL marca, d: BIT;

BEGIN

        componente: NEG_C2 GENERIC MAP (TAMANO_PALABRA) PORT MAP (A, B, R, D);

        PROCESS
        VARIABLE valor, limite, cuenta_carry, sol: INTEGER;
```

```
	BEGIN

		limite:= 2**TAMANO_PALABRA-1;

		FOR i IN 0 TO limite LOOP
			valor:=i;

			FOR k IN B'REVERSE_RANGE LOOP
				A(k)<=BIT'VAL(valor REM 2);--cambiamos A
				valor:=valor/2;
			END LOOP;

			FOR j IN 0 TO limite LOOP
				valor:=j;

				FOR k IN B'REVERSE_RANGE LOOP
					B(k)<=BIT'VAL(valor REM 2);--cambiamos B
					valor:=valor/2;
				END LOOP;
				WAIT FOR 20 ns;
			END LOOP;
		END LOOP;
	WAIT;
	END PROCESS;
END estructural;
```