Rubén Comerón Galán

Sesión 8 Sumador-restador

El código para el sumador restador es:

```
ENTITY SUMADOR_RESTADOR IS
	GENERIC(c: INTEGER:=4);
	PORT(X, Y: IN BIT_VECTOR(c-1 DOWNTO 0); SUMA_RESTA: IN BIT; R: OUT
BIT_VECTOR(c-1 DOWNTO 0); AC, DES, SIG, CERO: OUT BIT);
END SUMADOR_RESTADOR;

ARCHITECTURE estructural OF SUMADOR_RESTADOR IS
	COMPONENT RCA IS
		GENERIC (c: INTEGER:=2);
		PORT(x_in, y_in: IN BIT_VECTOR(c-1 downto 0); c_in: IN BIT; sal, c_out: OUT
BIT_VECTOR(c-1 downto 0));
	END COMPONENT;
	COMPONENT NEG_C2 IS
		GENERIC(n:INTEGER :=4);
		PORT(x: IN BIT_VECTOR(n-1 DOWNTO 0); e: IN BIT; r: OUT BIT_VECTOR(n-1
DOWNTO 0); des: 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;

	FOR ALL: RCA USE ENTITY WORK.RCA(estructural);
	FOR ALL: NEG_C2 USE ENTITY WORK.NEG_C2(estructural);
	FOR ALL: or2 USE ENTITY WORK.or2(estructural);
	FOR ALL: xor2 USE ENTITY WORK.xor2(comportamiento);
	FOR ALL: not2 USE ENTITY WORK.not2(comportamiento);

	CONSTANT TAMANO_PALABRA: INTEGER:=4;
	SIGNAL Y_SIGNED, R_aux, C_out, det_0: BIT_VECTOR(c-1 DOWNTO 0);
	SIGNAL Des_cambio_signo: BIT;
BEGIN
	negador: NEG_C2 PORT MAP(Y, SUMA_RESTA, Y_SIGNED, Des_cambio_signo);
	componente: RCA PORT MAP(X, Y_SIGNED, SUMA_RESTA, R_aux, C_out);
	G: FOR I IN 0 TO TAMANO_PALABRA-2 GENERATE
		primero: IF I=0 GENERATE
			detector_0: or2 PORT MAP(R_aux(i), R_aux(i+1), det_0(i));
		END GENERATE;
		siguientes: IF I>0 AND I<TAMANO_PALABRA-1 GENERATE
			detect_0: or2 PORT MAP(det_0(i-1), R_aux(i+1), det_0(i));
		END GENERATE;
	END GENERATE;
	deteccion_0: not2 PORT MAP(det_0(TAMANO_PALABRA-2), CERO);
	SIG<=R_aux(TAMANO_PALABRA-1);
	desbordamiento: xor2 PORT MAP(C_out(TAMANO_PALABRA-1), C_out(TAMANO_PALABRA-
2), DES);
	acarreo: xor2 PORT MAP(SUMA_RESTA, C_out(TAMANO_PALABRA-1), AC);

END ARCHITECTURE;
```

El código para su test-bench es:

```
ENTITY tb_sumador_restador IS
END tb_sumador_restador;

ARCHITECTURE estructural OF tb_sumador_restador IS
	COMPONENT SUMADOR_RESTADOR IS
		GENERIC(c: INTEGER:=4);
		PORT(X, Y: IN BIT_VECTOR(c-1 DOWNTO 0); SUMA_RESTA: IN BIT; R: OUT
BIT_VECTOR(c-1 DOWNTO 0); AC, DES, SIG, CERO: OUT BIT);
	END COMPONENT;

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

	CONSTANT TAMANO_PALABRA: INTEGER := 4;
	CONSTANT TAMANO_PALABRA_RCA: INTEGER := 2;

	SIGNAL A, B, R: BIT_VECTOR(TAMANO_PALABRA-1 downto 0);
	SIGNAL sum_sub, ac, des, sig, cero: BIT;

BEGIN

	componente: sumador_restador GENERIC MAP (TAMANO_PALABRA) PORT MAP (A, B,
sum_sub, R, ac, des, sig, cero);

	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;
                                        FOR carry IN 0 TO 1 LOOP
                                                sum_sub<=BIT'VAL(carry REM 2);
                                                WAIT FOR 20 ns;
                                        END LOOP;
                                END LOOP;
                        END LOOP;
                WAIT;
                END PROCESS;
        END estructural;
```