4. Saioa

4.1. Praktika: zirkuitu logikoen diseinua MSI eta ate logikoak erabiliz

Praktika honen helburua MSI zirkuituak erabiliz zirkuitu digital konplexuagoak garatzea da. Horretarako, zehaztapen hauek dituen zirkuitua garatu beharko da: BCD naturalean adierazitako bi zenbakiren arteko eragiketa aukeratzeko posibilitatea duen zirkuitua garatuko da. Ariketa posibleak bi dira. Bata, bi zenbakiren arteko batuketa (A+B), eta bestea, A zenbakia bider bi (A·2) egiten duena. Haien arteko bat aukeratzeko, E deritzogun sarrera izango du zirkuituak. A eta B zenbakiak adierazteko, ezinbestekoa izango da entrenadorearen zortzi etengailu erabiltzea, eta eragiketaren emaitza 7 segmentuko bi *display*etan aurkeztuko da. Funtzionamendu-prozedura hau izango da:

- Eragingaiak definitzen dira.
- Egin nahi den eragiketa aukeratzen da: Batuketa edo Biderketa.
- Emaitza *display*etan aurkezten da.

Saioa hasi aurretik prestatu beharreko atalak

- Bloke-diagrama marraztu behar da. Han adieraziko dira erabili behar diren zirkuitu integratu motak (ez da beharrezkoa zenbakia zehaztea) eta diseinatu behar diren zirkuituen funtzionalitatea. Halaber, argi adierazi behar dira zirkuitu bakoitzaren sarrera eta irteera-aldagai guztiak, eta blokeen arteko konexioak.
- Zirkuitu integratuak aukeratu.
- Haien ezaugarri orriak aztertu.

Laborategian garatu beharreko atalak

- Froga ezazu garatutako zirkuituaren funtzionamendua egokia dela QUARTUS II diseinurako programa erabiliz.
- Egin ezazu sistemaren muntaketa protoboard diseinurako plaka erabiliz. Zirkuituari integratu bat gehitzen zaion bakoitzean dena ondo konektatuta dagoela frogatu behar da.
- Programatu ezazu UP1 edo UP2n eskuragarri dagoen CPLDa eta froga ezazu funtzionamendua.

- Diseinua VHDL bidez deskriba ezazu, eta, QUARTUS IIa erabiliz froga ezazu programazioa zuzena dela. VHDL deskripzioa errazteko, praktika honi dagokion eranskina kontsulta dezakezu.

Erabil daitezkeen ZIak: 74284, 74147, 74148, 7400, 74185A, 7483, 7447, 7448, 74157, anodo komuneko 7 segmentuko *displaya*, eta bihurgailuak.

ERANSKINA: VHDL lengoaian diseinatzeko gida

```
-- BCD/7 SEGMENTU DESKODEGAILU ZIRKUITUA: 7447REN BALIOKIDEA
-- "WITH" SENTENTZIAREN ERABILERA

LIBRARY    ieee;
USE        ieee.std_logic_1164.all;

ENTITY BCD7segmentu IS
        PORT (
                kodea  : IN     STD_LOGIC_VECTOR(3 DOWNTO 0);
                segmentuak      : OUT STD_LOGIC_VECTOR(0 TO 6)
                );
END BCD7segmentu;

--SEGMENTUEN KITZIKAPENA 0AREKIN LORTZEN DA

ARCHITECTURE 7segmentu_bistaratzaile OF BCD7segmentu IS
BEGIN
        WITH kodea SELECT
          segmentuak <=       "0000001" WHEN "0000",
                        "1001111" WHEN "0001",
                        "0010010" WHEN "0010",
                        "0000110" WHEN "0011",
                        "1001100" WHEN "0100",
                        "0100100" WHEN "0101",
                        "1100000" WHEN "0110",
                        "0001111" WHEN "0111",
                        "0000000" WHEN "1000",
```

```
                    "0001100" WHEN "1001",
                    "1111111" WHEN OTHERS;
END 7segmentu_bistaratzaile;


-- 4TIK 1ERA ZIRKUITU MULTIPLEXADOREA

-- "WITH" SENTENTZIAREN ERABILERA

LIBRARY     ieee;

USE         ieee.std_logic_1164.all;

ENTITY mux4-1 IS

    PORT        (

            S       :  IN    STD_LOGIC_VECTOR (1 DOWNTO 0);
            X       :  IN    STD_LOGIC_VECTOR (3 DOWNTO 0);
            Y       :  OUT STD_LOGIC
            );
END mux4-1;

ARCHITECTURE multiplexadore OF mux4-1 IS

BEGIN

  WITH s SELECT

    Y <=    X(0) WHEN "00",
            X(1) WHEN "01",
            X(2) WHEN "10",
            X(3) WHEN OTHERS;
END multiplexadore;

-- 4TIK 2RA ZIRKUITU MULTIPLEXADOREA: 74153REN BALIOKIDEA

-- “CASE” SENTENTZIAREN ERABILERA “PROCESS” BATEN BARRUAN


LIBRARY     ieee;

USE         ieee.std_logic_1164.all;

ENTITY mux4-2 IS

    PORT        (

            s       : IN     STD_LOGIC_VECTOR (1 DOWNTO 0); -- Aukera
            A       : IN     STD_LOGIC_VECTOR (3 DOWNTO 0); -- A sarrera
            B       : IN     STD_LOGIC_VECTOR (3 DOWNTO 0); -- B sarrera
            Y       : OUT STD_LOGIC_VECTOR (1 DOWNTO 0) – Y irteera
            );
END mux4-2;
```

```
ARCHITECTURE multiplexadore OF mux4-2 IS
BEGIN
  PROCESS(s,A,B) – Prozesuaren aldagai-zerrenda
  BEGIN
        CASE s IS
                WHEN "00" =>
                        Y(0) <= A(0);
                        Y(1) <= B(0);
                WHEN "01" =>
                        Y(0) <= A(1);
                        Y(1) <= B(1);
                WHEN "10" =>
                        Y(0) <= A(2);
                        Y(1) <= B(2);
                WHEN OTHERS =>
                        Y(0) <= A(3);
                        Y(1) <= B(3);
        END CASE;
   END PROCESS;
END multiplexadore;

-- 3TIK 8RA ZIRKUITU DESKODEGAILUA: 74138REN BALIOKIDEA
-- "IF" ETA "WITH" SENTENTZIEN ERABILERA

LIBRARY     ieee;
USE         ieee.std_logic_1164.all;

ENTITY 3-8deskodegailua IS
        PORT (
                Sarrera: IN     STD_LOGIC_VECTOR(2 DOWNTO 0); -- dek. sarrera.
                G1,G2           : IN     STD_LOGIC; -- gaitze-sarrerak
                irteera         :  OUT  STD_LOGIC_VECTOR(7  DOWNTO  0)  –  dek.
irteera
                );
END 3-8deskodegailua;
```

```
--IRTEEREN KITZIKAPENA 0AREKIN LORTZEN DA
ARCHITECTURE deskodegailuaOF 8deskodegailua IS

SIGNAL bar_irteera : STD_LOGIC_VECTOR(7 DOWNTO 0); -- Barne aldagaia

BEGIN
        WITH sarrera SELECT
          bar_irteera <= "01111111" WHEN "000",
                         "10111111" WHEN "001",
                         "11011111" WHEN "010",
                         "11101111" WHEN "011",
                         "11110111" WHEN "100",
                         "11111011" WHEN "101",
                         "11111101" WHEN "110",
                         "11111110" WHEN OTHERS;

        PROCESS(G1,G2, bar_irteera) – Aldagai-zerrenda
        BEGIN
                IF G1 = '1' AND G2 = '0' THEN
                        irteera <= bar_irteera;
                ELSE
                        irteera <= "11111111";
                END IF;
        END PROCESS;


END deskodegailua;

-- 4TIK 16RA ZIRKUITU DESKODEGAILUA: 74154REN BALIOKIDEA
-- "WITH" SETENTZIAREN ERABILERA

LIBRARY     ieee;
USE         ieee.std_logic_1164.all;


ENTITY 4-16deskodegailua IS
        PORT (
                sarrera : IN     STD_LOGIC_VECTOR(3 DOWNTO 0);
                irteera  : OUT STD_LOGIC_VECTOR(15 DOWNTO 0)
                );
END 4-16deskodegailua;
```

```
--IRTEERAK 0 LORTUZ KITZIKATZEN DIRA
ARCHITECTURE deskodegailua OF 16deskodegailua IS
BEGIN
        WITH sarrera SELECT
           irteera <=  "0111111111111111" WHEN "0000",
                       "1011111111111111" WHEN "0001",
                       "1101111111111111" WHEN "0010",
                       "1110111111111111" WHEN "0011",
                       "1111011111111111" WHEN "0100",
                       "1111101111111111" WHEN "0101",
                       "1111110111111111" WHEN "0110",
                       "1111111011111111" WHEN "0111",
                       "1111111101111111" WHEN "1000",
                       "1111111110111111" WHEN "1001",
                       "1111111111011111" WHEN "1010",
                       "1111111111101111" WHEN "1011",
                       "1111111111110111" WHEN "1100",
                       "1111111111111011" WHEN "1101",
                       "1111111111111101" WHEN "1110",
                       "1111111111111110" WHEN OTHERS;
END deskodegailua;


-- ZIRKUITU KONPARADOREA: 7485REN BALIOKIDEA
-- "IF" ETA "WITH" SENTENTZIEN ERABILERA


LIBRARY     ieee;
USE         ieee.std_logic_1164.all;


ENTITY 4bit_konparadorea IS
        PORT (
                A       : IN    STD_LOGIC_VECTOR(3 DOWNTO 0);
                B       : IN    STD_LOGIC_VECTOR(3 DOWNTO 0);
                c_in    : IN    STD_LOGIC_VECTOR(2 DOWNTO 0); -- A<B A=B
A>B
                c_out   : OUT STD_LOGIC_VECTOR(2 DOWNTO 0) -- A<B A=B
A>B
                );
END 4bit_konparadorea;
```

```
-- IRTEERAK 0 LORTUZ KITZIKATZEN DIRA

ARCHITECTURE konparadorea OF 4bit_konparadorea IS

	SIGNAL c_out_in : STD_LOGIC_VECTOR(2 DOWNTO 0); -- berne aldagaia

BEGIN
	PROCESS(A,B) – Aldagai-zerrenda
	BEGIN
		IF A > B THEN
			c_out_in <=  "001"; -- A > B
		ELSIF A < B THEN
			c_out_in <=  "100"; -- A < B
		ELSE
			c_out_in <=  "010"; -- A = B
		END IF;
	END PROCESS;

	PROCESS(c_in,c_out_in) -- Aldagai-zerrenda
	BEGIN
		IF c_out_in = "001" OR c_out_in = "100" THEN
			c_out <= c_out_in;
		ELSE
			c_out <= c_in;
		END IF;
	END PROCESS;

END konparadorea;
```

-- BIT BAKARRA DUTEN BI ZENBAKIEN ARTEKO BATUTZAILEA "FULLADDER"

-- ERAGILE LOGIKOEN ERABIlLERA

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

ENTITY      fulladd IS
        PORT (
                cin,x,y : IN     STD_LOGIC; -- cin: carry in, sarrerak x-y
                s,cout  : OUT STD_LOGIC  -- cout: carry out, irteera s
                );
END fulladd;

ARCHITECTURE programa OF fulladd IS

BEGIN
        s <= x xor y xor cin;
        cout <= (x and y) or (cin and (x or y));


END programa;
```

-- BIT BAKARRA DUTEN BI ZENBAKIREN ARTEKO BATUTZAILEAREN ZIRKUITUA

```
-- "PACKAGE" ETA "COMPONENT" EN ERABILERA
LIBRARY     ieee;
USE         ieee.std_logic_1164.all;

PACKAGE fulladd_pakete IS
        COMPONENT fulladd
                PORT (
                        a,b     : IN     STD_LOGIC;
                        s       : OUT  STD_LOGIC;
                        cin     : IN     STD_LOGIC;
                        cout    : OUT  STD_LOGIC
                        );
        END COMPONENT;
END fulladd_pakete;
```

-- LAU BIT DUTEN BI ZENBAKIREN ARTEKO BATUTZAILEAREN ZIRKUITUA: 7483REN BALIOKIDEA

-- "COMPONENT" ETA "WITH" ETA "FOR - GENERATE" SENTENTZIEN ERABILERA

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

ENTITY      s4+4bitbatutzaile      IS
        PORT (
                X,Y     : IN     STD_LOGIC_VECTOR(3 DOWNTO 0); -- 4 bit dituzten
aldagaiak
                S       : OUT STD_LOGIC_VECTOR(3 DOWNTO 0); -- Batuketa
                cin     : IN     STD_LOGIC; -- carry in
                cout    : OUT STD_LOGIC  -- carry out
                );
END s4+4bitbatutzaile;

ARCHITECTURE programa OF s4+4bitbatutzaile IS
        SIGNAL S_int        :  STD_LOGIC_VECTOR(3  DOWNTO  0);  --  barne
aldagaiak
        SIGNAL c_int        : STD_LOGIC_VECTOR(4 DOWNTO 0);

COMPONENT fulladd
        PORT (
                cin,x,y : IN     STD_LOGIC;
                s,cout  : OUT STD_LOGIC
                );
        END COMPONENT;
BEGIN
        c_int(0) <= cin; -- carry bita esleitzen da
        batuketa:       -- X eta Yren arteko BATUKETA emango duen blokearen
etiketa.
        FOR i IN 0 TO 3 GENERATE
                batutzaile: fulladd PORT MAP (c_int(i),X(i),Y(i),S_int(i),c_int(i+1));
        END GENERATE;

        S <= S_int;
        cout <= c_int(4);
    END programa;
```