

# AURREPROIEKTUA

Lan honen helburua proiektu bat garatzea izango da, bi aukera eman zizkiguten hau garatzeko, bietan helburua berdina zen; makina saltzaile baten funtzionamendua diseinatzea. Guk, lehenengo aukera hartu dugun non EHU-rako makina bat garatu beharko dugun.

Lehenik eta behin, proiektu hau garatzeko XILINX ISE programa erabili dugu non "VHDL" lengoaia erabiliko den guztia programatzeko. Ondoren, programa guztia bukatu ondoren, programa honek duen elektronika plaka batean, lana funtzionatzeko gai izan behar da.

Hasteko, gure proiektua hurrengo egiteko programa garatuko dugu; txanponak sartu ditugu eta hauen prezioen arabera, makina barruan dagoen produktu bat edota beste bat lortuko dugu. Hiru txanpon bakarrik erabilgarriak izango dira makina honetan, "5", "10", "20" txanponak bereziki.

Honekin batera, txanpon kantitate zehatza sartzen ez bada, makinak automatikoki txanponak itzuliko ditu eta ez da ezer erosiko. Aurreko esaldian azaldu den informazioarekin ondoriozta genezake makinak ez duela kanbiorik emango.

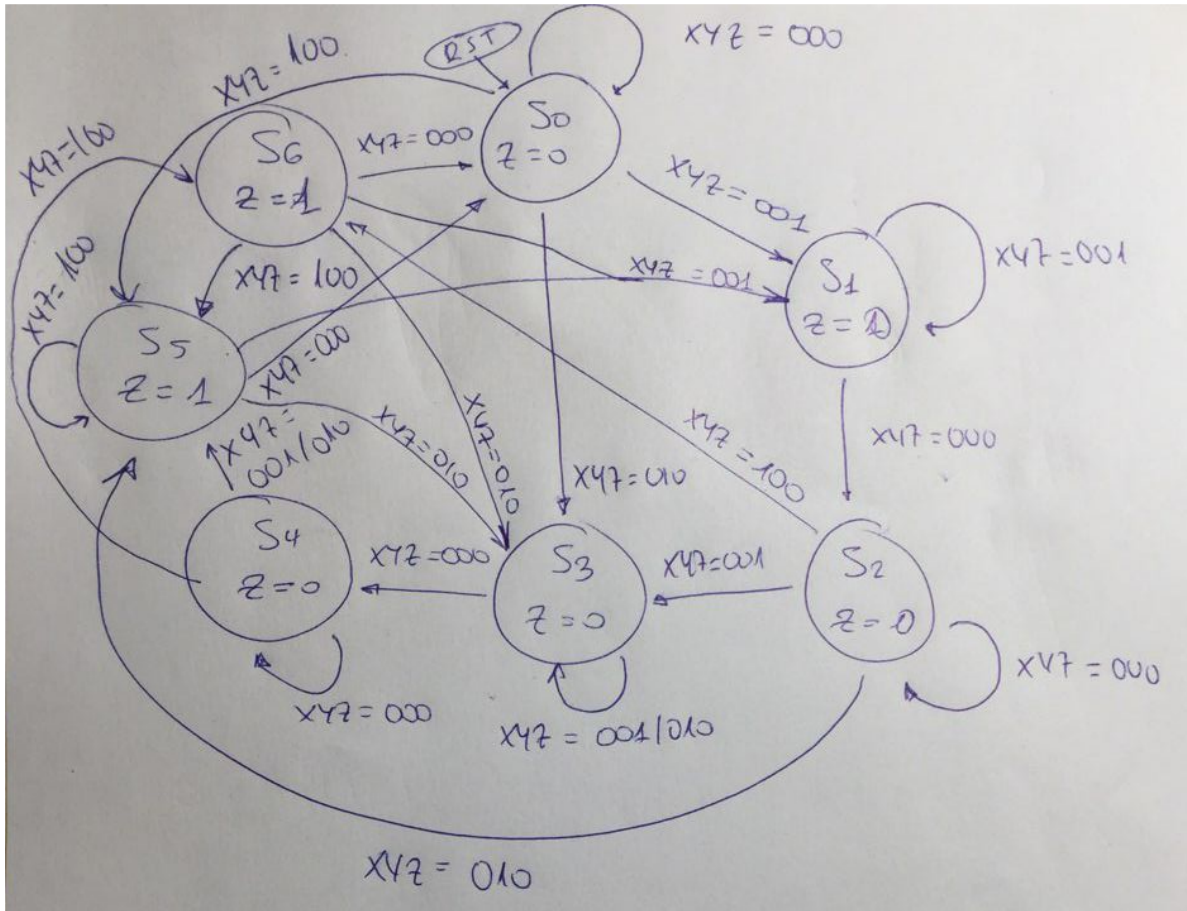
Amaitzeko funtzionamendua hurrengo izango da, 15 zentimotik, 25 zentimo artean dagoen diru kopurua sartu ondoren intsignia bat lortuko dugu. Bestalde, 25 zentimo baino gehiago sartzerakoan pompondun artilezko txano bat aterako da. Bukatzeko, 15 zentimo baino gutxiago sartu ez badugu makinak ez digu ezer eskainiko.

Lehen esan den bezala proiektua egiteko VHDL lengoaia erabiliko da, honen ondorioz, egoera diagrama bat erabiliko dugu proiektuaren funtsa azaltzeko, ulermena errazagoa izan dadin.

Gure proiektuak, 7 egoera ditu honen ondorioz, 3 biteko egoerak erabiliko ditugu.

$2^2 < 7 < 2^3$

## EGOERA DIAGRAMA:



$$xyz = x_2x_1x_0$$

## EGOERA TAULA:

<b>Egoera</b>	$x_2x_1x_0=000$ (0 zent)	$x_2x_1x_0=001$ (5 zent)	$x_2x_1x_0=010$ (10 zent)	$x_2x_1x_0=100$ (20 zent)	<b>Irteera</b> $Z_1Z_0$
<b>s0</b>	s0	s1	s3	s5	<b>00</b>
<b>s1</b>	s2	s1	ez dago	ez dago	<b>00</b>
<b>s2</b>	s2	s3	s5	s6	<b>00</b>
<b>s3</b>	s4	s3	s3	ez dago	<b>00</b>
<b>s4</b>	s4	s5	s5	s6	<b>00</b>
<b>s5</b>	s0	s1	s3	s5	<b>01</b>
<b>s6</b>	s0	s1	s3	s5	<b>10</b>

## FUNTZIONAMENDUA:

Gure diseinuak 7 egoera ditu non azken biak produktuak diren. Produkturen bat eskuratzean, bi egoera haiek, lehengo egoera bezala balitz erabiliko ditugu eta berriro diru kantitate batera heltzean, berriro ere produktua emango dute.

Goiko taulan, gauza interesgarri bat daukagu, lehengo eta hirugarren egoerak, beste egoetara igarotzeko orduan ez daukate nora joan. Guzti hau azalpen bat du noski, bi egoera hauek pasuko egoerak dira, hau da, 0 egoeratik 1 egoerara pasa dadin lehenik eta behin 5 zentimoko txanpon bat sartu behar dugu. Ondoren, pausuko egoera sartuko da jokoan, hau da, pausuko egoerak egingo duena txanpon hori kontatzea da. Adibidez, guk 5 zentimoko txanpon bat sartzen dugu makinan eta honek egingo duena hurrengoa da; txanpon hori hartuko du kontuan (kontatuko du) geroago beste txanpon bat sartzeko(edozein) eta kontaketa egiteko. Horregatik, pausuko egoeran 5 zentimoko txanpon bat badago egoera horretan jarraituko du, kontaketa egiten ari delako. Beste edozein txanponarekin (0-rekin bakarrik, beste biak 10 eta 20 beste egoeratan sartuko direlako) bigarren egoerara pasatuko da.

Aurrekoaren berdina gertatuko da hirugarren egoerarekin, baina kasu honetan, 10 zentimoko txanpona sartzen dugunez, makinan 5 edo 10-ko txanponak sartzean egoera horretan geldituko da kontaketa egiteko. Beste edozein kasutan (0-rekin) hurrengo egoerara pasatuko da, hau da, 4-ra.

Aurretik azaldu den bezala bi egoera daude non produktuak ateratzeko gai izango garen, 7 eta 8 egoerak. Honen ondorioz, produktua atera ondoren makina berriro ere beste produktu bat ateratzeko gai izan beharko zen. Horregaitik, bi egoera hauek 1 egoera bezala jokatuko dute.

Ondoren, Reset bat dugu 1 egoeran, hau egingo duena, diru kopuru okerra sartzen dugunean, makinak berriro ere dirua itzuliko digu eta berriro hasteko aukera izango dugu. "RESET" hau berriro hasteko aukera hori eskainiko digu.

Egoera taularen eskuineko aldean, irteera aldagaia daukagu non honek 1 izango da produktua atera denean eta 0 oraindik diru kantitate beharrezkoa sartu ez dugunean.

Beheko aldean VHDL kodea utzi dugu, kodea egiterako garaian, egoera diagrama hartu dugu kontuan.

Orduan, adibide bat jarriko dugu; lehenik eta behin 5 zentimoko txanpona sartuko dugu eta lehenengo egoerara eramango digu, hor bertan 5 zentimoko txanpona sartuz egoera horretan mantenduko gara eta ez badugu txanponik sartzen bigarren egoerara eramango gaitu. Hor berriro ere guk esango dugu nora joan nahi dugun, kasu honetan, 10 zentimo sartuko ditugu eta kasu honetan insignia lortuko dugu. Bukatzeko momentu honetatik aurrera, egoera hau (kasu honetan 5) lehenengo egoera duen jarrera izango du. Kasu honetan Z0 irteera aktibatuko da.

## VHDL kodea:

```
'glibrary IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
-- Balioak gehitzeko (+) funtzioa duen liburutegia.
use IEEE.STD_LOGIC_arith.ALL;
entity proiektua is PORT(
    clk: IN std_logic;
    reset: IN std_logic_vector(0 DOWNT0 0);
    dirua: IN std_logic_vector(2 DOWNT0 0);
    irteerak: OUT std_logic_vector(2 DOWNT0 0)
);
end proiektua;
```

```
ARCHITECTURE Behavioral of proiektua is
    TYPE estados IS(s0,s1,s2,s3,s4,s7,s8);
    SIGNAL kontua: estados;
    SIGNAL kontadore : integer := 1;
        SIGNAL clk2 : std_logic;
```

```
BEGIN
    PROCESS(clk2,reset)
    BEGIN
        -- Reset etengailua aktibatzean edozein momentuan guren zentimoak Ora itzuliko dira
        IF reset = "1" THEN
            kontua <= s0;
        ELSIF (clk2'EVENT AND clk2 = '1') THEN
            CASE kontua IS
                -- 0
                WHEN s0 =>
                    IF dirua = "000" THEN kontua <= s0; -- Egoera mantentzen du
                    ELSIF dirua = "001" THEN kontua <= s1; -- +5
                    ELSIF dirua = "010" THEN kontua <= s3; -- +10
                    ELSIF dirua = "100" THEN kontua <= s7; -- +20
                    END IF;4t
                -- 5 (pasuko egoera)
                WHEN s1 =>
                    IF dirua = "001" THEN kontua <= s1; -- Egoera mantentzen du
                    ELSIF dirua = "000" THEN kontua <= s2; --Hurrengo egoerara pasatzen da
                    END IF;
                -- 5 (benetazko egoera)
                WHEN s2 =>
                    IF dirua = "000" THEN kontua <= s2; -- Egoera mantentzen du
                    ELSIF dirua = "001" THEN kontua <= s3; -- +5
                    ELSIF dirua = "010" THEN kontua <= s7; -- +10
                    ELSIF dirua = "100" THEN kontua <= s8; -- +20
                    END IF;
                -- 10 (pasuko egoera)
                WHEN s3 =>
                    IF dirua = "000" THEN kontua <= s4; -- Hurrengo egoerara pasatzen da
                    ELSIF dirua = "001" THEN kontua <= s3; -- Egoera mantentzen du
                    ELSIF dirua = "010" THEN kontua <= s3; -- Egorea mantentzen du
                    END IF;
```

```

-- 10 (benetazko egoera)
WHEN s4 =>
    IF dirua = "000" THEN kontua <= s4; -- Egoera mantentzen du
    ELSIF dirua = "001" THEN kontua <= s7; -- +5
    ELSIF dirua = "010" THEN kontua <= s7; -- +10
    ELSIF dirua = "100" THEN kontua <= s8; -- +20
    END IF;
-- 15
WHEN s7 => -- Insigniaren Egoera
    kontua <= s7;

-- 25
WHEN s8 => -- Txanoaren Egoera
    kontua <= s8;
    -- 500000000 tick = 5 segundu (100000kHz-ko clk bat daukagu, beraz 100000000 tick/seg)
    IF (kontadore = 500000000) THEN
        kontadore <= 1;
        kontua <= s0;
    ELSE
        kontadore <= kontadore + 1; -- +100000000 tick/seg
    END IF;
END CASE;
END IF;
END PROCESS;
PROCESS(kontua)
BEGIN
    CASE kontua IS
        --15
        WHEN s7 =>
            irteerak <= "001";
        --25
        WHEN s8 =>
            irteerak <= "010";

        -- Beste edozein kasurako LEDak itzalita egongo dira
        WHEN OTHERS =>
            irteerak <= "000";
    END CASE;
END PROCESS;

PROCESS(clk)
BEGIN
    IF (clk'EVENT AND clk = '1') THEN
        IF (kontadore = 250000000) THEN
            clk2 <= '1';
        ELSIF (kontadore = 500000000) THEN
            kontadore <= 1;
            clk2 <= '0';
        ELSE
            kontadore <= kontadore + 1; -- +100000000 tick/seg
        END IF;
    END IF;
END PROCESS;
END Behavioral;

```