Proiektu honetan, sartzen genuen diruaren arabera zerbait ematen zuen makina bat diseinatu behar dugu VDHL lengoaian.

Horretarako, hiru txanpon mota dauzkagu: 5, 10 eta 20 zentimokoak. Txanponak zer eratan sartzen ditugun arabera, hiru irteera mota lortuko ditugu.

Hiru irteera hauek izozkia, haizagailua, eta biak batera izango dira. Txanponak sartzerakoan hiruretako irteera bat lortuko dugu, nahi eta nahi ez, baina diru gehiago sartu badugu, ez digu itzuliko.

Gainera, haizagailuaren aukera lortzean minutu bat egon behar da piztuta, eta orduan berriro hasierako egoerara bueltatu. Honetarako VDHL lengoaia erabiliz kontagailu bat egin behar dugu.

Erabiliko dugun hardware-a Nexys3 plaka programagarria izango da. Erabiliko ditugun osagaiak hauek izango dira:

* 3 led gure hiru irteerak adierazteko.
* 3 botoi txanponak adierazteko.
* Switch bat, reset seinalea izango dena, edozein momentutan hasierako egoerara bueltatu ahal izateko.

Sarrerak:

001: 5 zentimo.

010: 10 zentimo.

100: 20 zentimo.

s0 egoeran hasieratuko gara eta 3 txanpon mota desberdin sartzeko aukera izango dugu: 5, 10 eta 20 zentimokoak.

Irteerak:

Txanpon konbinazioen bidez irteera desberdina lortuko dugu.

001(s5): izozkia.

010(s6): haizagailua.

100(s7): izozkia eta haizagailua.

**Egia-Taula**

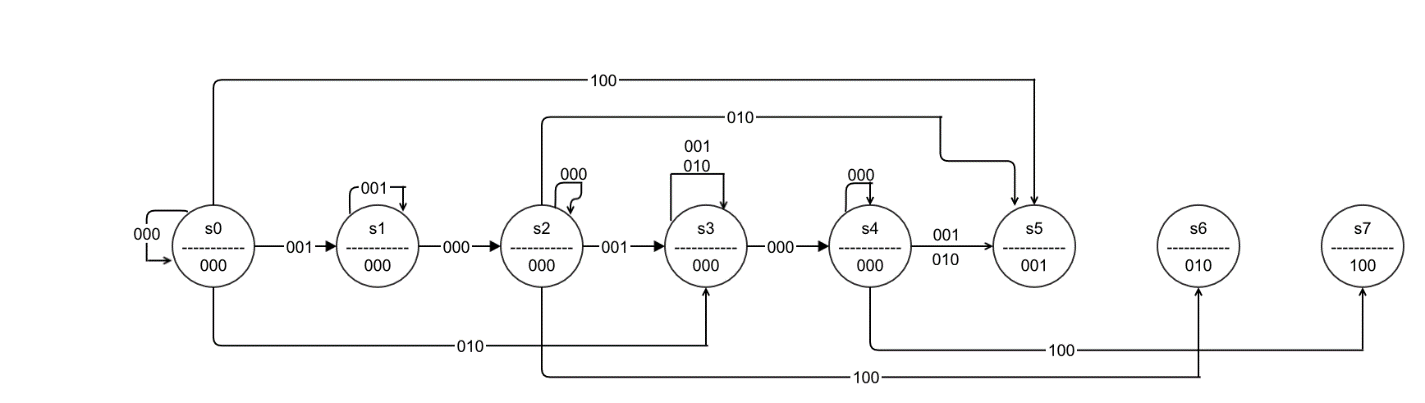
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Egoera | Dirudik ez | 5 zentimo | 10 zentimo | 20 zentimo | Irteera  Z2Z1Z0 |
| s0 | s0 | s1 | s3 | s5 | 000 |
| s1 | s2 | s1 | - | - | 000 |
| s2 | s2 | s3 | s5 | s6 | 000 |
| s3 | s4 | s3 | s3 | - | 000 |
| s4 | s4 | s5 | s5 | s7 | 000 |
| s5 | - | - | - | - | 001 |
| s6 | - | - | - | - | 010 |
| s7 | - | - | - | - | 100 |

5 zentimo sartzen baditugu s1 egoerara pasako gara. s1 egoera laguntzaile bat da s2 egoerara bide ematen duena. s2 egoeran 5 zentimo izango ditugu. Kasu honetan 5 zentimo gehiago, 10 zentimo edo 20 zentimo sar ditzakegu. 5 zentimo sartuz gero s3 egoera laguntzailera igaroko gara, aldiz, 10 zentimo sartuta s5 irteera egoerara helduko gara(izozkia, 001) eta 20 zentimo sartzean s6 irteerako egoerara pasa eta 010 egoera izango dugu(haizagailua, 010).

s0 egoeran 10 zentimo sartzen baditugu, s3 egoera laguntzailera pasako gara eta bertatik s4 egoerara. s4 egoeran 3 aukera izango ditugu: 5 zentimo, 10 zentimo edo 20 zentimo gehiago sartu. 20 zentimo sartzean s7 irteera egoera lortuko dugu(100,haizagailua eta izozkia). Bestalde, 10 edo 5 zentimo sartzen baditugu s5 irteera egoerara helduko gara. s5 001 irteera izango da, izozkiarena.

Edozein egoeratan reset botoia sakatzean s0 egoerara bueltatuko gara, zikloa berriz hasteko.

s6 eta s7 egoeratan, aldiz, minutu bat pasatzerakoan (5 segundo gure kasuan) reset automatikoa egingo du, s0 egoerara berriro bueltatuz.

**Egoera-Diagrama**

Reset automatikoa egiteko, Nexys 3 plakaren erlojua erabiliko dugu. Erloju hau 100000kHz dabil, hau da 100000000 tick segundo bakoitzeko. Honek esan nahi du, kodea maiztasun horrekin eguneratuko dela.

Gure kasuan, minutu bat ez itxarotearren, 5 segundo irauten duen kontagailu bat eraiki dugu. Hau burutzeko, aldagai berri bat deklaratu dugu, integer motatakoa, eta 1 balioa duena. Kontadorea martxan jartzeko, s6 edo s7 egoeretan egon behar gara. Behin bata edo bestean egon, integer hori, daukan balioa + 1 egingo du erlojuaren tick bakoitzeko.

5 segundoak deklaratu dugun aldagaia 500000000 baliora iristean lortuko ditugu. Behin balio hau dugula, integer aldagaia 1ean eta programa s0 egoeran, hasierakoa, ipiniko ditugu.

Aipatzekoa da “use IEEE.STD\_LOGIC\_arith.ALL” liburutegia beharrezkoa dela “+” aritmetikoa erabiltzeko.

**UCF artxiboa**

NET "clk" LOC = "V10" | IOSTANDARD = "LVCMOS33";

Net "clk" TNM\_NET = sys\_clk\_pin;

TIMESPEC TS\_sys\_clk\_pin = PERIOD sys\_clk\_pin 100000 kHz;

**## Leds**

NET "irteerak<0>" LOC = "U16" | IOSTANDARD = "LVCMOS33";

NET "irteerak<1>" LOC = "V16" | IOSTANDARD = "LVCMOS33";

NET "irteerak<2>" LOC = "U15" | IOSTANDARD = "LVCMOS33";

**## Switches**

NET "reset<0>" LOC = "T10" | IOSTANDARD = "LVCMOS33";

**## Buttons**

NET "dirua<0>" LOC = "D9" | IOSTANDARD = "LVCMOS33";

NET "dirua<1>" LOC = "B8" | IOSTANDARD = "LVCMOS33";

NET "dirua<2>" LOC = "C4" | IOSTANDARD = "LVCMOS33";

**VHD artxiboa**

library 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 Plentziapolis\_Main is PORT(

clk: IN std\_logic;

reset: IN std\_logic\_vector(0 DOWNTO 0);

dirua: IN std\_logic\_vector(2 DOWNTO 0);

irteerak: OUT std\_logic\_vector(2 DOWNTO 0)

);

end Plentziapolis\_Main;

ARCHITECTURE Behavioral of Plentziapolis\_Main is

TYPE estados IS(s0,s1,s2,s3,s4,s5,s6,s7);

SIGNAL kontua: estados;

SIGNAL kontadore : integer := 1;

BEGIN

PROCESS(clk, reset)

BEGIN

kontua <= s0;

**-- Reset etengailua aktibatzean edozein momentuan guren zentimoak 0ra itzuliko dira**

IF reset = "1" THEN

kontua <= s0;

ELSIF (clk'EVENT AND clk = '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 <= s5; **-- +20**

END IF;

**-- 5 (pasuko egoera)**

WHEN s1 =>

IF dirua = "001" THEN kontua <= s1; **-- Egoera mantentzen du**

ELSIF dirua = "000" THEN kontua <= s2; **--Hurrengo egoerara**

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 <= s5; **-- +10**

ELSIF dirua = "100" THEN kontua <= s6; **-- +20**

END IF;

**-- 10 (pasuko egoera)**

WHEN s3 =>

IF dirua = "000" THEN kontua <= s4; **-- Hurrengo egoerara**

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 <= s5; **-- +5**

ELSIF dirua = "010" THEN kontua <= s5; **-- +10**

ELSIF dirua = "100" THEN kontua <= s7; **-- +20**

END IF;

**-- 15**

WHEN s5 => **-- Izozkiaren Egoera**

kontua <= s5;

**-- 25**

WHEN s6 => **-- Haizagailuaren Egoera**

kontua <= s6;

**-- 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;

**-- 30**

WHEN s7 => **-- Haizagailu**

kontua <= s7;

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 s5 =>

irteerak <= "001";

**--25**

WHEN s6 =>

irteerak <= "010";

**--30**

WHEN s7 =>

irteerak <= "100";

**-- Beste edozein kasurako LEDak itzalita egongo dira**

WHEN OTHERS =>

irteerak <= "000";

END CASE;

END PROCESS;

end Behavioral;