

Sistema Digitalak Diseinatzeko Oinarriak (SDDO)

3. laborategia

Adei Arias eta Ander Prieto – 7. taldea

1) Sarrera

Praktika honetan, XC6SLX16 zirkuitu integratuaren bidez, BCD kodetik 7-segment display baten seinaleetara bihurtzea egiten duen kodetzaile baten inplementazioa garatu dugu. Praktika hau ordenagailuz egin dugun lehenengoa izan da. Gainera, kurtsoko hirugarrena denez, egingo dugun azken aurrekoa da.

Praktika burutzeko, Xilinx ISE izeneko software programa erabili dugu. Programa honek egiten duena VHDL kode hizkuntzan idatzitako funtzioa garatu eta simulatzea da.

2) Ariketa

2.1) Kodea

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

ENTITY convert IS
PORT( e: IN STD_LOGIC_VECTOR (3 DOWNTO 0);
      leds : OUT STD_LOGIC_VECTOR(1 TO 7) ) ;

END convert ;

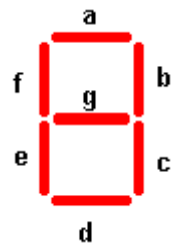
ARCHITECTURE a OF convert IS
BEGIN
    PROCESS ( e )
    BEGIN
        CASE e IS
            -- abcdefg
            WHEN "0000" => leds<= "0000001" ;
            WHEN "0001" => leds<= "1001111" ;
            WHEN "0010" => leds<= "0010010" ;
            WHEN "0011" => leds<= "0000110" ;
            WHEN "0100" => leds<= "1001100" ;
            WHEN "0101" => leds<= "0100100" ;
            WHEN "0110" => leds<= "0100000" ;
            WHEN "0111" => leds<= "0001111" ;
            WHEN "1000" => leds<= "0000000" ;
            WHEN "1001" => leds<= "0001100" ;
            WHEN OTHERS => leds<= "1111111" ;
        END CASE ;
    END PROCESS ;
END a ;
```

2.2) Azalpena

Kodean ikusi ahal dugun bezala, kontagailuaren LEDak piztuko dira 4 biteko zenbaki bat programa honetan sartzeko unean. "ARCHITECTURE" atal barruan ohartzen bagara, argi ikusi dezakegu displayaren LEDak baldintza batzuekin piztuko direla.

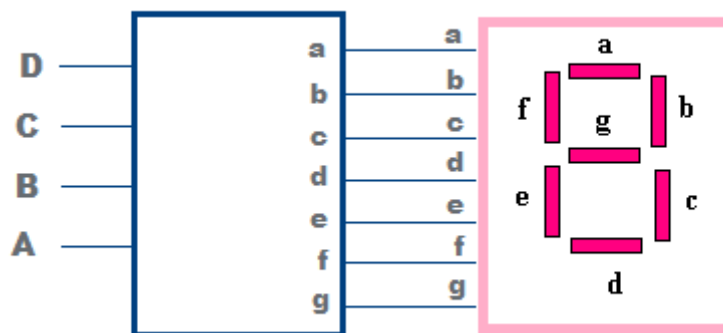
Hasteko, guk sartutako zenbakia BCD kodearen zenbaki bat izan behar da (hau da, 0-tik 9 arte dagoen zenbaki bat). Beste edozein zenbaki sartzen badugu (10-tik 15 arte) gure displayak ez du inolako zenbakirik ezta LEDik aurkeztuko.

Gero, zenbakia pizteko garaian, 7 biteko zenbaki bitarrak izango ditugu. Bit bakoitza kontagailuaren letra bat adieraziko du, hau da, 1010101 zenbakia badugu lehenengo bita ezkerretik hasita, 1 da eta honek “a” letra adieraziko du, gero bigarren bita “b” eta horrela zenbakia bukatu arte.



Zenbakiak irudikatzean, zenbaki bitarretan “0” zenbakia duten posizioak LED piztuko dute eta besteak (“1” zenbakia duten posizioa) ez dute LEDik piztuko. Adibidez, 0001111 zenbakiak, 7 zenbakia irudikatzen du (a, b eta c LEDetan 0 daudelako).

Zirkuitu guzti hau muntatzeko kodetzaile bat erabili dugu programa barruan. Kodetzaile baten bidez, zirkuitu integratuak egiteko gai izango gara.



Honela konektatuko zen display bat kodetzaile batekin, ikusi ahal dugun bezala zazpi marra (displayaren marra kopurua) lotzen dituzte bi aparatu hauek. Gero kodetzaileak lau sarrera ditu non lau biteko zenbakia sartu ahal dugun eta zazpi irteera ditu. Funtzio honen egia taula hurrengoa izango litzateke:

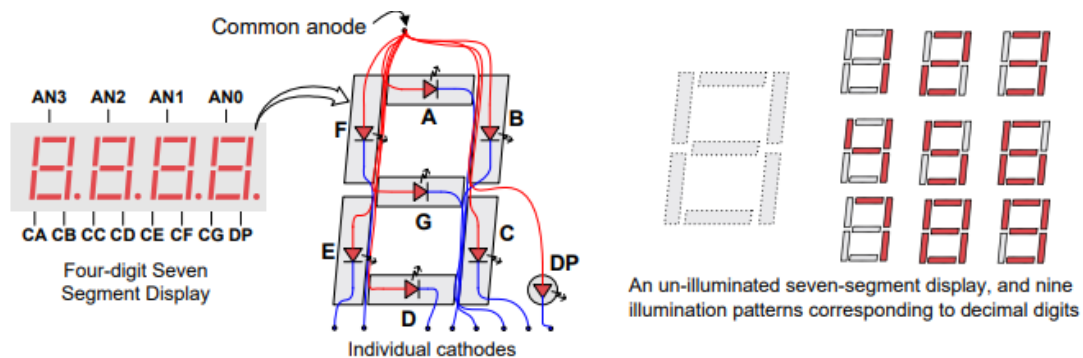
| A | B | C | D | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | x | x | x | x | x | x | x |
| 1 | 0 | 1 | 1 | x | x | x | x | x | x | x |
| 1 | 1 | 0 | 0 | x | x | x | x | x | x | x |
| 1 | 1 | 0 | 1 | x | x | x | x | x | x | x |
| 1 | 1 | 1 | 0 | x | x | x | x | x | x | x |
| 1 | 1 | 1 | 1 | x | x | x | x | x | x | x |

Lehen esan dudan moduan, BCD kodea, 1tik 9arte dauden zenbakiak bitarrean jartzea da. Beste zenbakiak (10tik 15arte) ez dira kontatzen, horregatik, "X" ipiniko dugu.

Egia taula hau oso luzea da. Honen ondorioz, funtzio hauek sinplifikatzeko "*Karnaugh-en mapa*" erabiliko dugu. Guztira 7 egin behar genituzke, bat segmentu bakoitzeko. BCD kodetik kanpo dauden zenbakiak "X" bat dutenez, "*Karnaugh-en mapa*" egiterako orduan, "mintermino" edo "maxtermino" bezala hartuko ditugu balioak (guk adostuko dugu zein hartzen dugun).

Bukatzeko, aurreko guztia garatu ostean, ate logikoen bidezko adierazpena egin behar litzateke.

Guk hau guztia Xilinx ISE programaren bidez gauzatu genuen, honen ondorioz ez genuen arazorik izan.



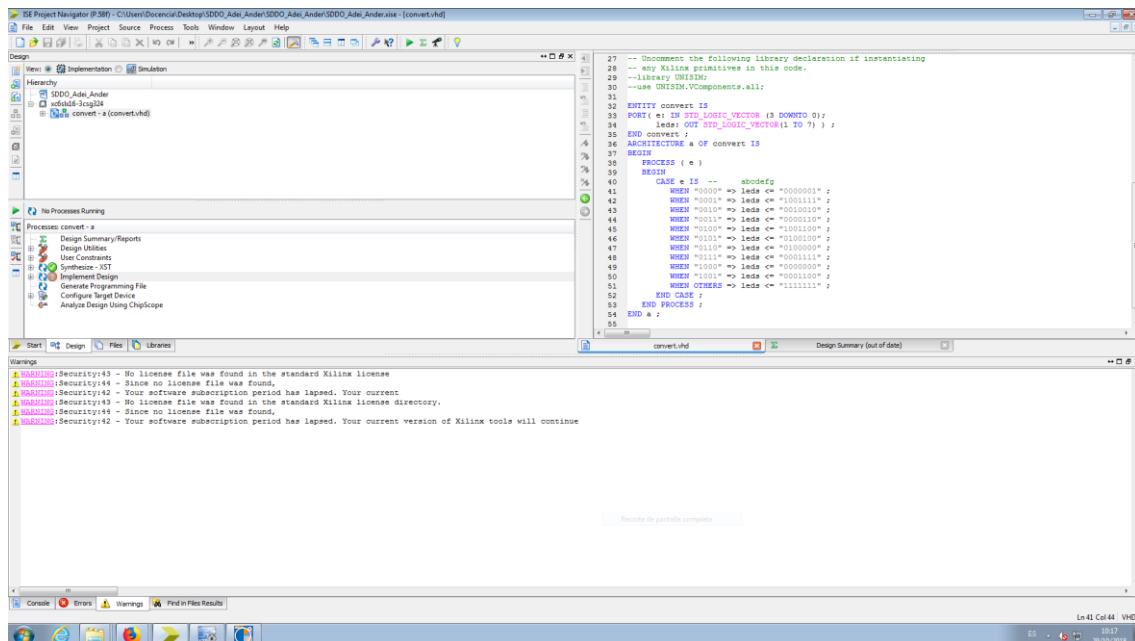
Marrazki honetan oso argi ikusi dezakegu nola funtzionatuko lukeen programatzen ari garen display-a. Alde batetik anodo komun bat sartuko da gure aparatuan eta gero kable bakoitza display-era heltzen denean, ukapen batekin anodo hauek katodoetan bihurtuko dira eta indibidualki aterako dira kanporantz.

Spartan3 izan zen erabili genuen programaren barruan erabili genuen plaka, prozesu hau gauzatzeko. Hurrengo atalean, Spartan3-aren diagrama aurkituko da.

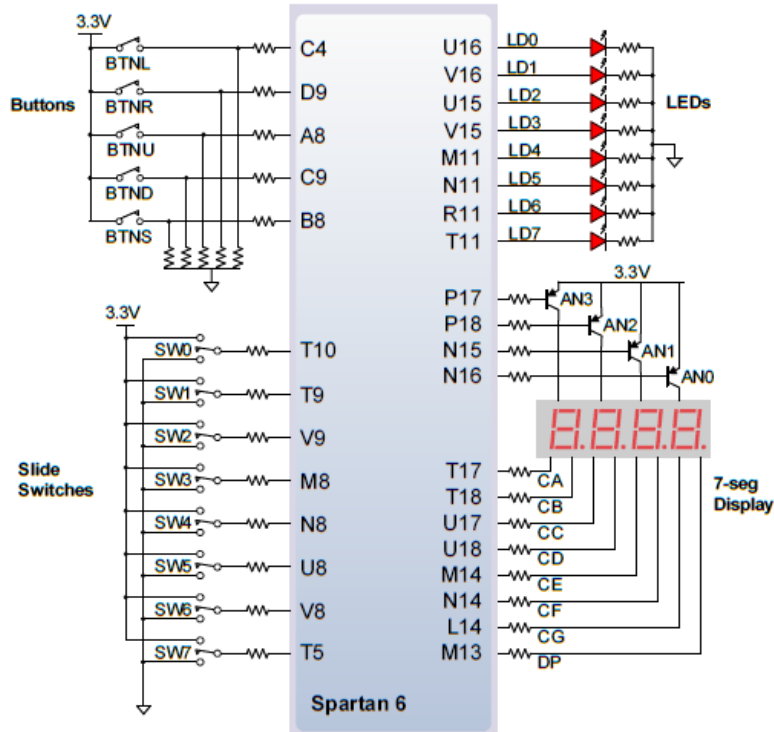
2.3) Programa

Sarrerari esan dugun bezala, Xilinx ISE programa erabili dugu. Programa konplexua den arren, ez da zaila maneiatzen jakitea.

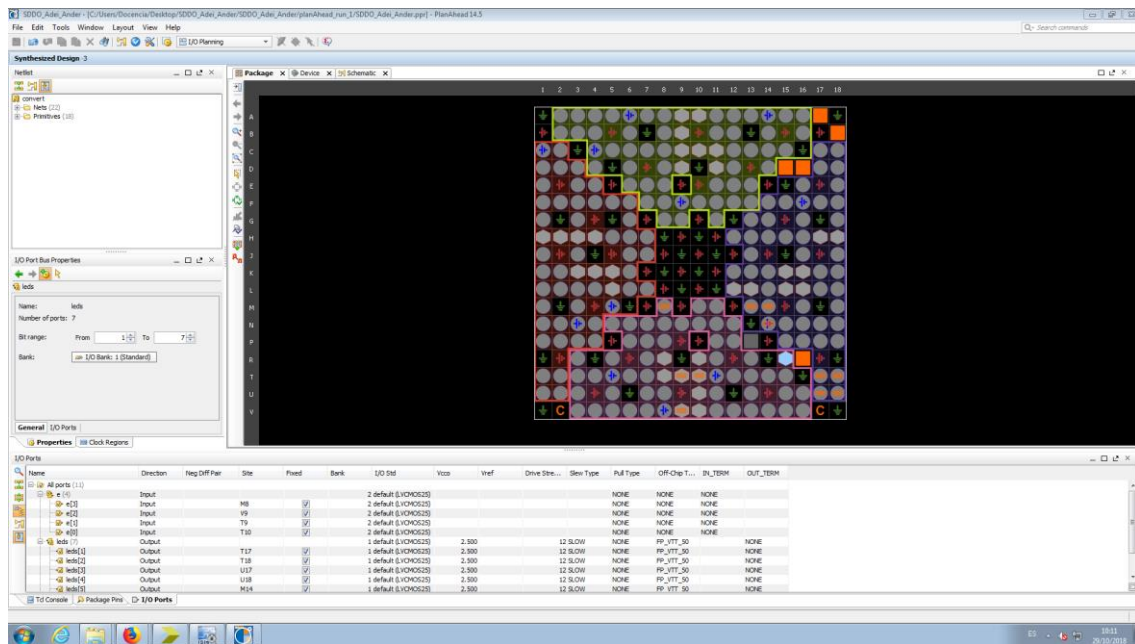
Beheko irudi honetan softwarearen interfazea daukagu. Eskumako laukitxo hartan kodea idazten dugu; hain zuzen, gorago aipatutakoa. Kodean idatzitakoa garatzeko, hasteko sintetizatu eta geroago diseinua inplementatu beharko da. Ondoren, beste software baten bidez plakara pasatzeko aukera izango dugu.



Irudian ikus daitekenez, diseinua implementatzeko dago. Honetarako, beste software ezberdin bat erabiltzen da: PlanAhead. Hemen, gure diseinua implementatuko dugu. Hau da, plakaren barruan egingo diren konexioak lotuko ditugu. Beheko kapturan ikus daitekenez, sarrera eta irteera bakoitzari letra eta zenbaki determinatu bat ematean, lotuta direla interpretatuko du softwareak. Letra eta zenbakia Spartan bakoitzarentzat ezberdina da, eta kasu honetan hurrenekoa da, gorago ere aipatu duguna:



Konexioak burututa daudenean, holako antza izango du PlanAhead softwarea:



Hau bukatzean, Xilinx programa nagusira itzuli eta plakara bidaltzeko prest egongo da.

3) Ondorioak

Praktika hau burutu ondoren hainbat gauza ondoriozta ditzakegu:

- Orokorrean 7-segment display baten funtzionamendua ikasi dugu. Orain, iratzargailuak, mikrouhinak... nola funtzionatzen duten badakigu.
- VHDL kodearekin edozein zirkuitu irudika daiteke. Horregatik, praktika onuragarria izan zaigu kode hizkuntza hau ezagutzeko