

DIGITÁLIS RENDSZERTERVEZÉS II HÁZI FELADAT

7 szegmenses kijelző vezérlése gombokkal
JEGYZŐKÖNYV

Oktató: Vígh Tamás

Készítette: Pap Csaba
4-es csoport. Hétfő 11.30-13.20.
2010.11.25.

7 szegmenses kijelző vezérlése gombokkal

FELADAT SPECIFIKÁCIÓJA

Kezdetben két szegmens világít a jobb szélső 7-szegmenses kijelzőn.

A 0-ás és az 1-es nyomógomb megnyomásával lehet léptetni a két szegmenst az óramutató járásával megegyező ill. ellenkező irányban körbe-körbe (csak a kijelző szélső szegmenseit kell használni, a g szegmenst soha).

ELVI MEGOLDÁS

Elsőként a meghatároztam az állapotok számát. Ez pedig 7, mivel a két szegmens a kijelzőn nem vehet fel 7 állapotnál többet (akár balra forgatjuk, akár jobbra).

Magát a vezérlést egy process-ben valósítottam meg, amihez egy lokális, de mégis kívülről elérhető változóra volt szükség (hasznalt_seg_local). A forgatás semmi másból nem áll, mint a bitek tologatásából, ami általánosan leolvasható, ha a 6 használt bit 7 kombinációját (állapotait) leírjuk.

Az állapotok: (balra forgatás esetén)

111100 ez a kezdőállapot (bal oldaliak világítanak)

111001

110011

100111

001111

011110

111100

Látható, hogy a két nulla „balra csúszik”. Jobbra forgatás esetén, természetesen fordított az állapotok sorrendje. A „csúszást” a következőképpen oldottam meg: a 6 bit elejére fűztem a 2., 3., 4., 5., 6. biteket (0.1.2.3.4. helyi értékű bitek) valamint az első bitet pedig a végére fűztem.

A programkód a következő oldalon található.

A FELADAT VHDL PROGRAMJA

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Hazi_feladat_potzh is
    Port ( reset : in STD_LOGIC;
          clk : in STD_LOGIC;
          gomb : in STD_LOGIC_VECTOR (1 downto 0);
          hasznalt_segment : out STD_LOGIC_VECTOR (5 downto 0);
          nem_hasznalt_seg : out STD_LOGIC_VECTOR (1 downto 0);
          digit : out STD_LOGIC_VECTOR (3 downto 0));
end Hazi_feladat_potzh;

architecture Behavioral of Hazi_feladat_potzh is
    signal hasznalt_seg_local: STD_LOGIC_VECTOR (5 downto 0);

begin

    digit<="1110";
    nem_hasznalt_seg<="11";

    process (clk, reset, gomb, hasznalt_seg_local)
    begin

        if reset='1' then hasznalt_seg_local<="111100";
        else if clk'event and clk='1' then
            if (gomb(0)='1') then
                hasznalt_seg_local<=hasznalt_seg_local(4 downto 0)
                & hasznalt_seg_local(5);
            else if (gomb(1)='1') then
                hasznalt_seg_local<=hasznalt_seg_local(0) &
                hasznalt_seg_local(5 downto 1);
            else hasznalt_seg_local<=hasznalt_seg_local;
            end if;
        end if; -- ha a gomb(0) = '1' feltétel vége
        end if; -- ha van felfutó él feltétel vége

        end if; -- ha a reset = '1' feltétel vége

        hasznalt_segment<=hasznalt_seg_local;

    end process;

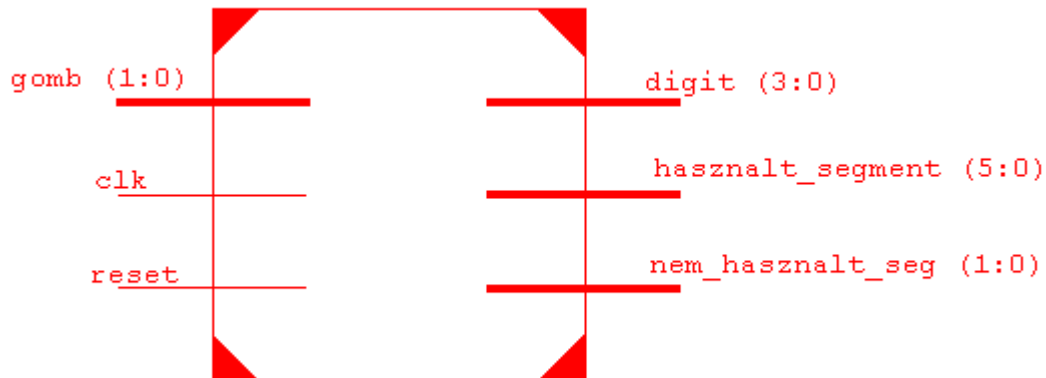
end Behavioral;
```

JELKIOSZTÁS

A jelkiosztás a SPARTAN3E gyakorló panelen a következő (a panelen az XC3S100E TQ144-es tokozású FPGA áramkör található):

NET "clk" LOC = P54;	Órajel
NET "digit[0]" LOC = P34;	jobbszélső kijelző
NET "digit[1]" LOC = P33;	
NET "digit[2]" LOC = P32;	
NET "digit[3]" LOC = P26;	
NET "gomb[0]" LOC = P69;	BTN0 (0-s gomb)
NET "gomb[1]" LOC = P48;	BTN1
NET "hasznalt_segment[0]" LOC = P25;	A szegmens
NET "hasznalt_segment[1]" LOC = P16;	B szegmens
NET "hasznalt_segment[2]" LOC = P23;	C szegmens
NET "hasznalt_segment[3]" LOC = P21;	D szegmens
NET "hasznalt_segment[4]" LOC = P20;	E szegmens
NET "hasznalt_segment[5]" LOC = P17;	F szegmens
NET "nem_hasznalt_seg[0]" LOC = P83;	G szegmens
NET "nem_hasznalt_seg[1]" LOC = P22;	DP szegmens
NET "reset" LOC = P41;	BTN3 (3-as gomb)

BLOKKSÉMA



MEGJEGYZÉS

A program a XILINX ISE 12.2 –vel készült, ebből adódóan a lábkiosztás (.ucf) fílet sem a XILINX PACE program készítette el, hanem a Design Suite-hoz tartozó **PlanAhead** nevű szoftver.

TARTALOM

7 szegmenses kijelző vezérlése gombokkal.....	2
Feladat specifikációja	2
Elvi megoldás.....	2
A feladat VHDL kódja	3
Jelkiosztás	4
Blokkséma	4
Megjegyzés.....	4