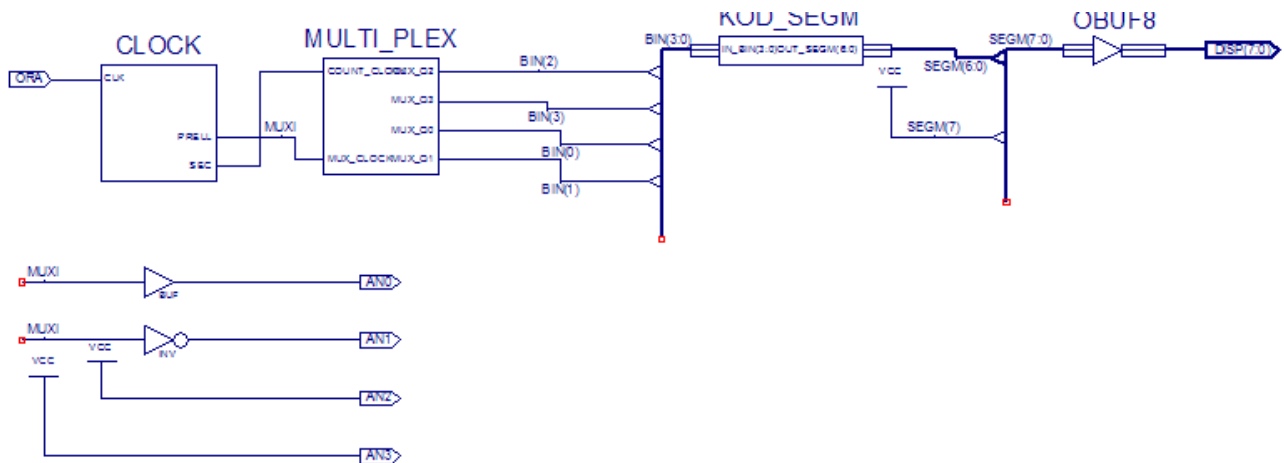


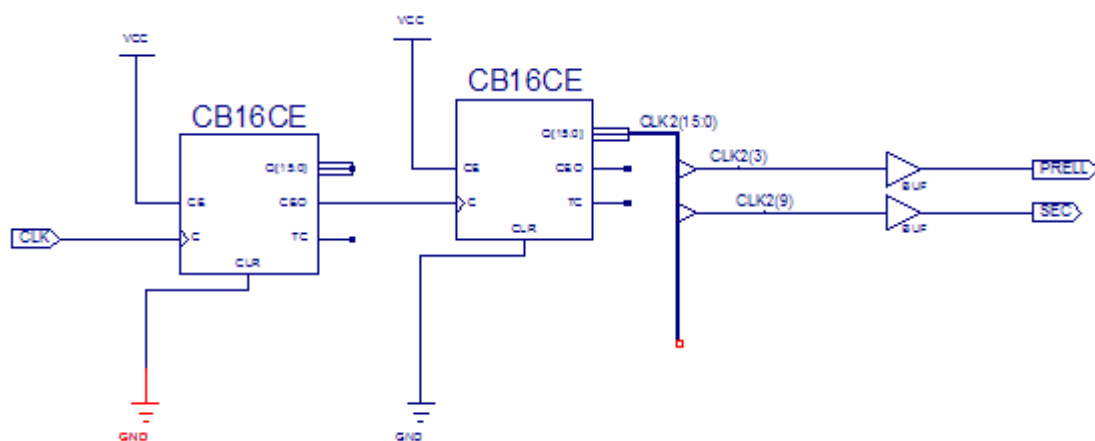
Feladatok rövid ismertetése

1, Multiplexelés a kijelzőkön:

a, Két 7-szegmens kijelzőre való multiplex üzemű kiíratás megvalósítása, áramkörtí szerkesztő segítségével:

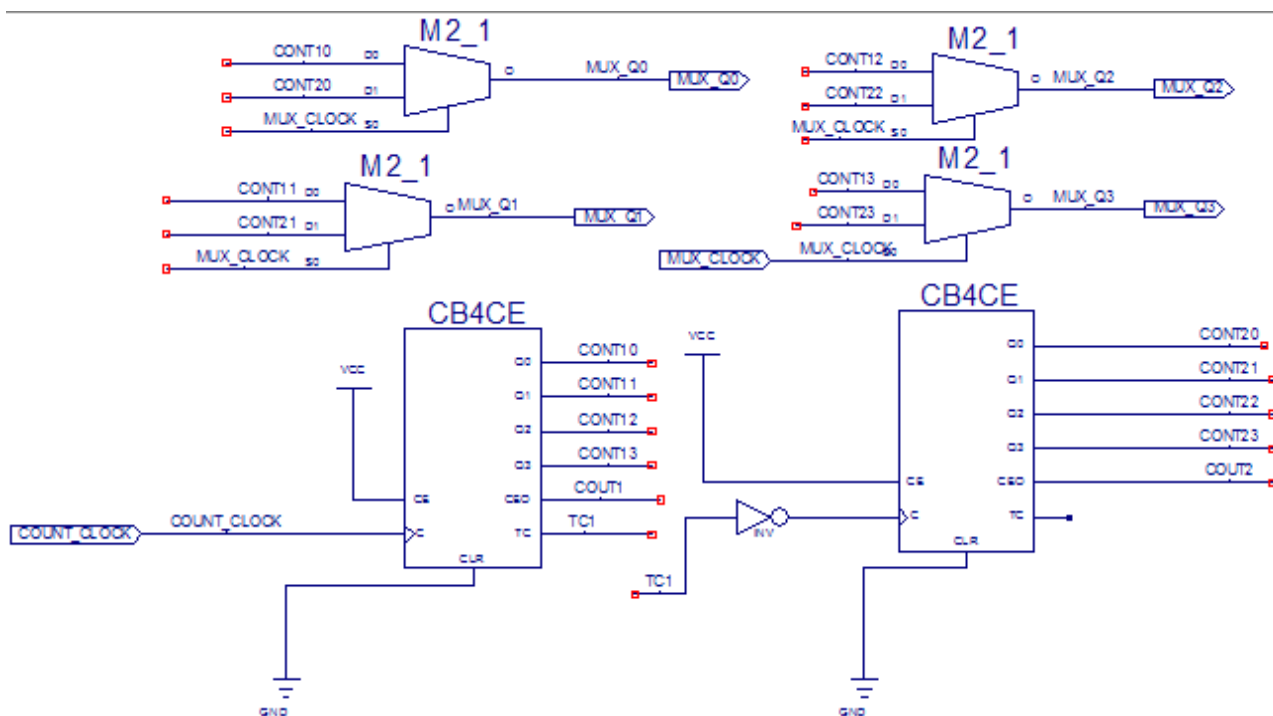


Clock modul:



A CLK bemenetre kössük az 50 MHz-es órajelet. A Prell kimenet egy gyors (ms-os) frekvenciát ad, melyet felhasználhatunk a későbbiek során nyomógombok, kapcsolók pergésmentesítésére. A SEC kimenet kb. 1 Hz-es órajelet szolgáltat.

Multi_Plex modul:



A Count_clock bemenet a számlálás, míg a Mux_clock a multiplexálás (a két kijelző váltogatása) frekvenciáját határozza meg. Természetesen a modul könnyen bővíthető x kijelzőre. A multiplexáláshoz nyugodtan használhatjuk az Clock modul Prell kimenetét, vagy akár létrehozhatunk egy újabb leágazást a Clock modulból.

Kod_Segm modul:

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

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity KOD_SEGM is
    Port ( IN_BIN : in STD_LOGIC_VECTOR (3 downto 0);
          OUT_SEGM : out STD_LOGIC_VECTOR (6 downto 0));
end KOD_SEGM;

architecture Behavioral of KOD_SEGM is

begin

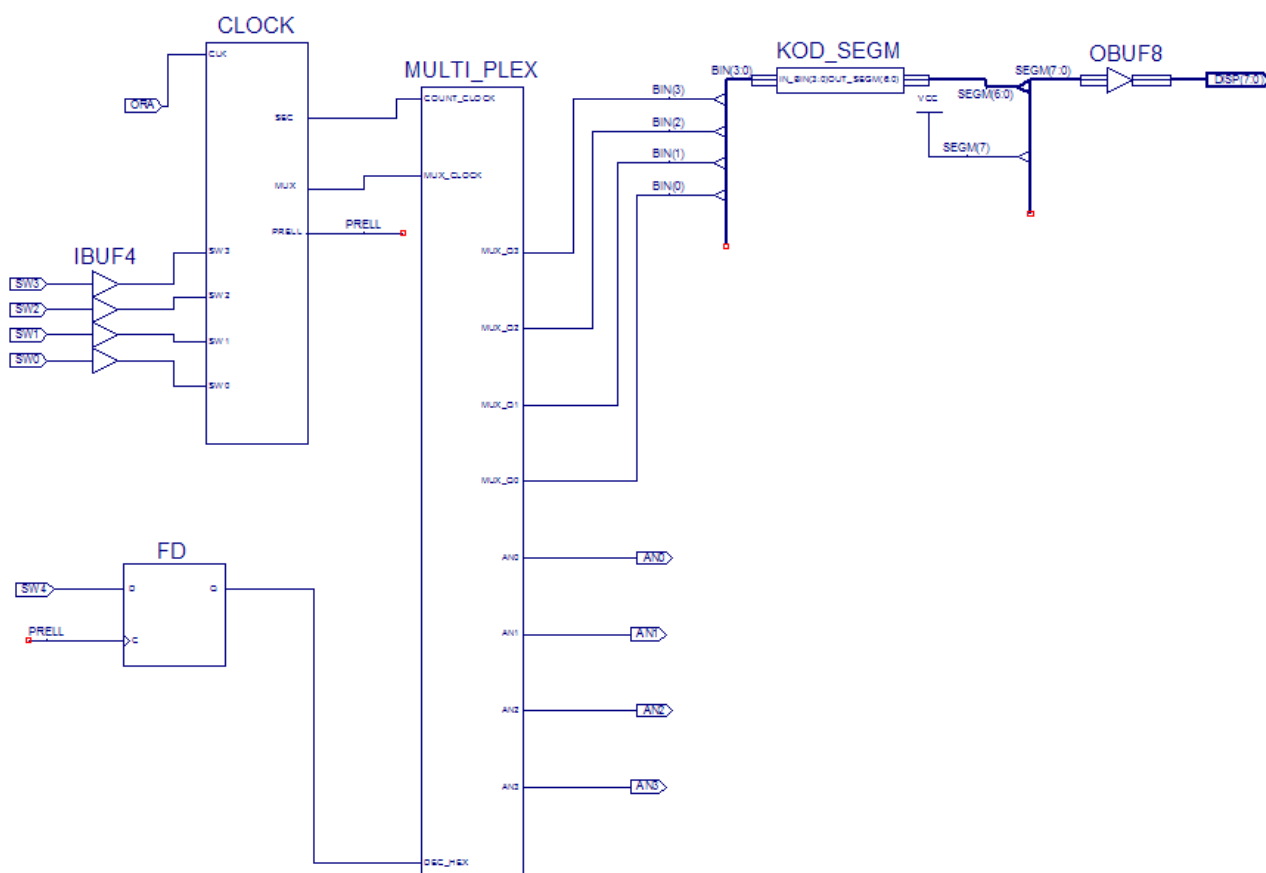
with IN_BIN SElect

OUT_SEGM<=  "1111001"    when "0001", --1
            "0100100"    when "0010", --2
            "0110000"    when "0011", --3
            "0011001"    when "0100", --4
            "0010010"    when "0101", --5
            "0000010"    when "0110", --6
            "1111000"    when "0111", --7
            "0000000"    when "1000", --8
            "0010000"    when "1001", --9
            "0001000"    when "1010", --A
            "0000011"    when "1011", --B
            "1000110"    when "1100", --C
            "0100001"    when "1101", --D
            "0000110"    when "1110", --E
            "0001110"    when "1111", --F
            "1000000"    when others; --0

end Behavioral;
```

b, Számláló áramkör négy kijelzőre, kapcsolóval választhatóan decimális, ill. hexa kijelzés.

Top Level:



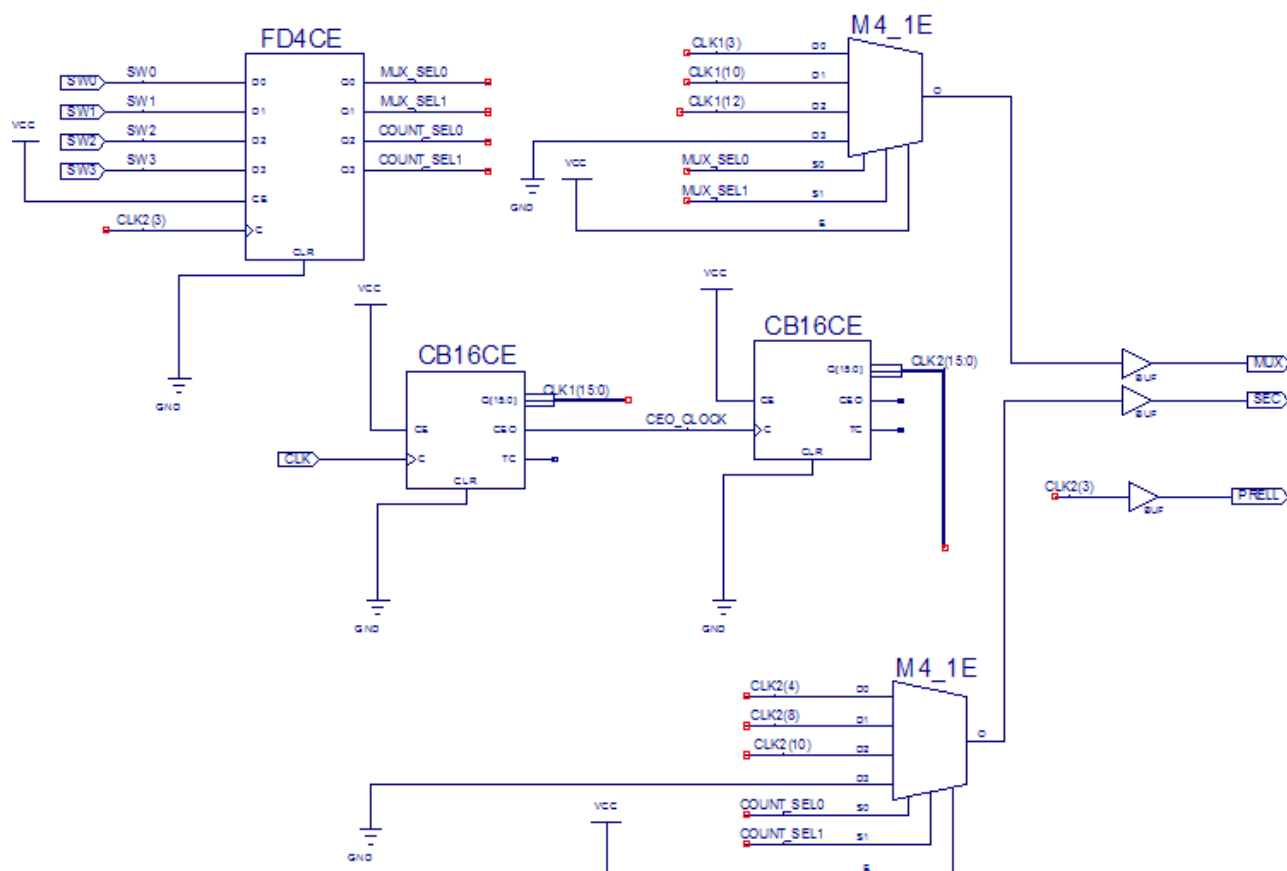
Az FD jelzésű egység egy Flip-Flop, mellyel könnyen elvégezhető egy nyomógomb, vagy kapcsoló pergesmentesítése.

A Clock modul szolgáltatja a leosztott órajelet a számláláshoz, ill. a multiplexáláshoz.

A Multi_plex modul szolgáltatja a kijelzéshez szükséges bináris kódot, valamint az egyes kijelzők „váltogatását”.

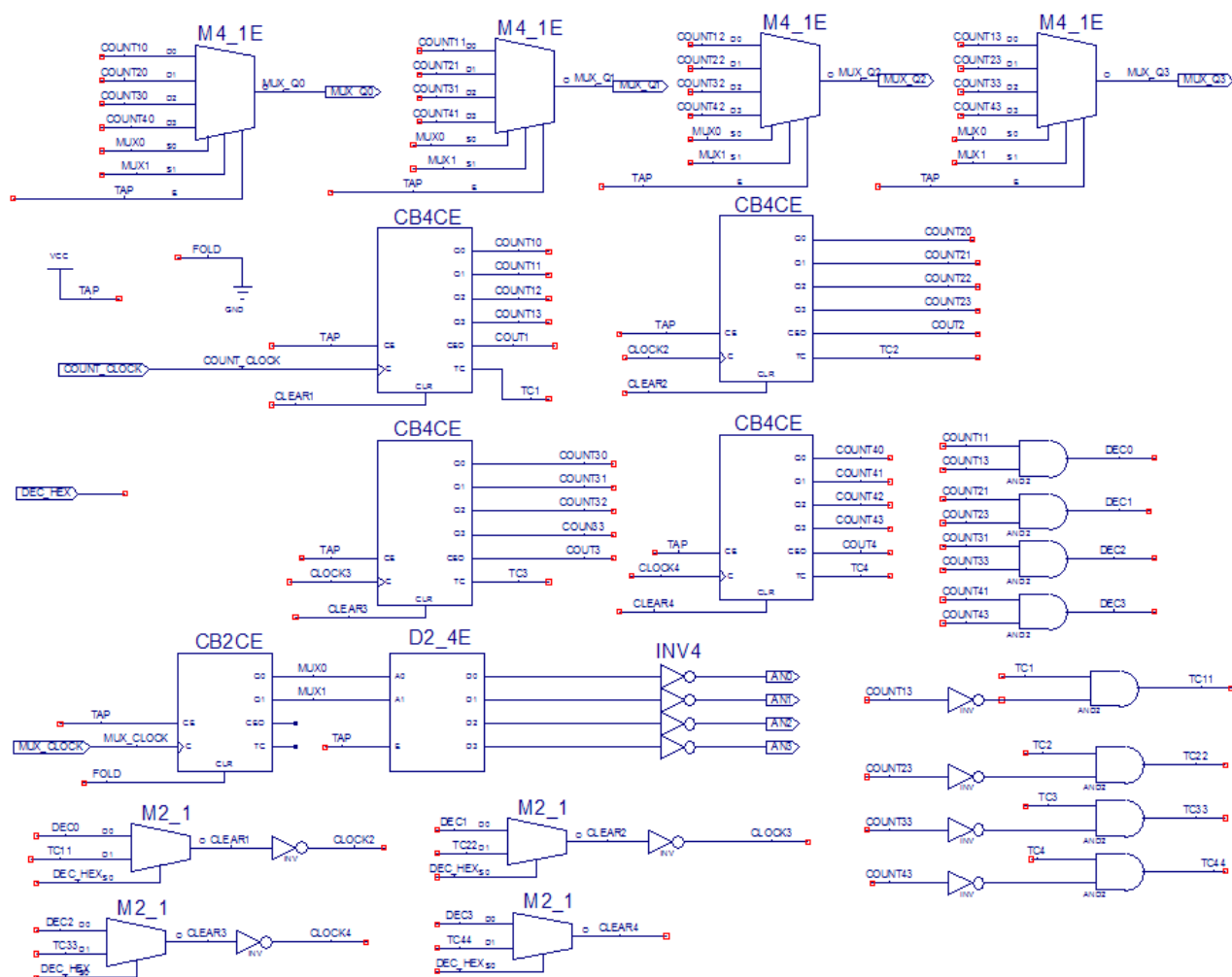
A Kod_Segm modul a beérkező bináris kódot átalakítja a 7-szegmens kijelzőhöz használt kódra.

Clock modul:



A program funkcionális működéséhez nem szükséges ilyen bonyolult órajel modul. Jelen esetben lehetőségünk van mind a multiplexálás (SW0;SW1), mind a számlálás (SW2;SW3) frekvenciájának állítására. Érdeemes megfigyelni, milyen problémát okoz a túlságosan nagy, ill. túlságosan kis multiplexálási frekvencia. Itt is megtalálható a pergésmentesítés (FD4CE), ezt elhelyezhetjük a belső rétegekben (modulokban), vagy a top szinten is – esetleg külön modult hozhatunk létre, melynek feladata kizárólag a prell megszüntetése.

Multi_Plex modul:



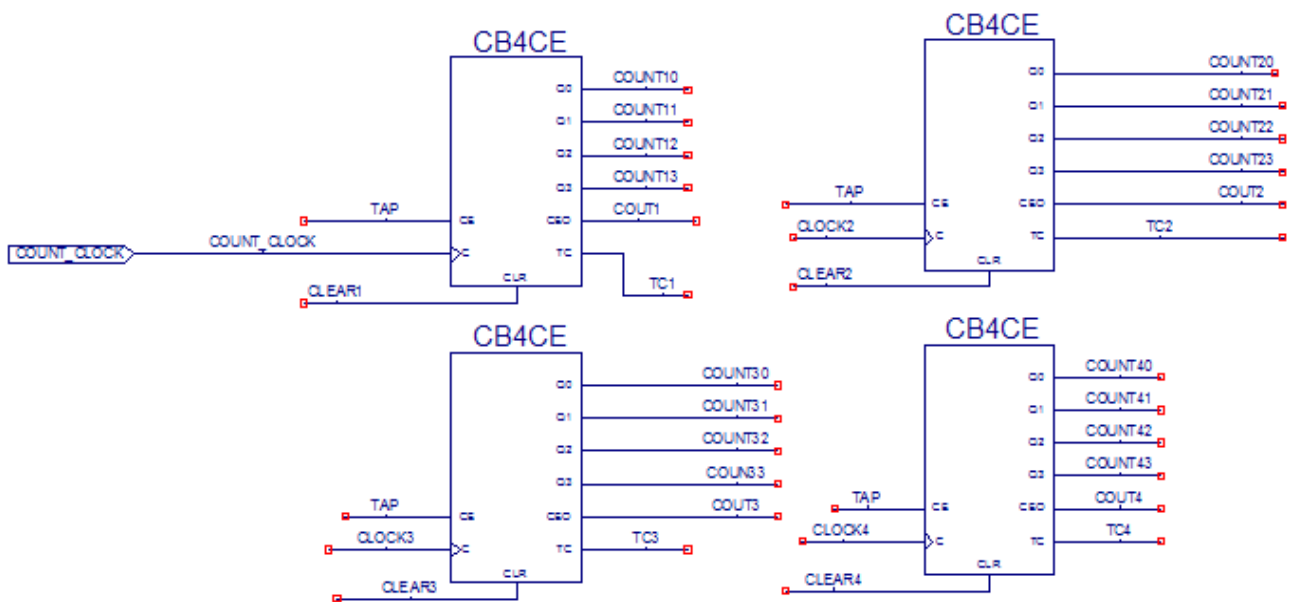
A Multi_Plex modul a program lelke, ezért ezt részletesebben ismertetjük. A modult a következő egységekre bonthatjuk fel:

– Táp-föld blokk:



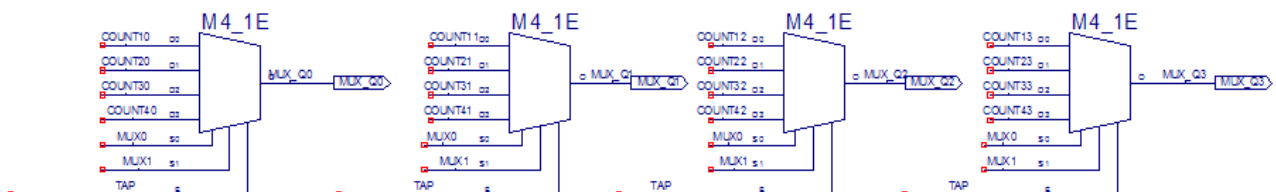
Bonyolultabb (sok alkatrész tartalmazó) hálózat esetén célszerű elnevezni egy Gnd, ill. egy Vcc vezeték, így megspórolhatjuk a sok Gnd, Vcc jelölések lerakását, melyek csak a helyet foglalják.

– Számláló blokk:



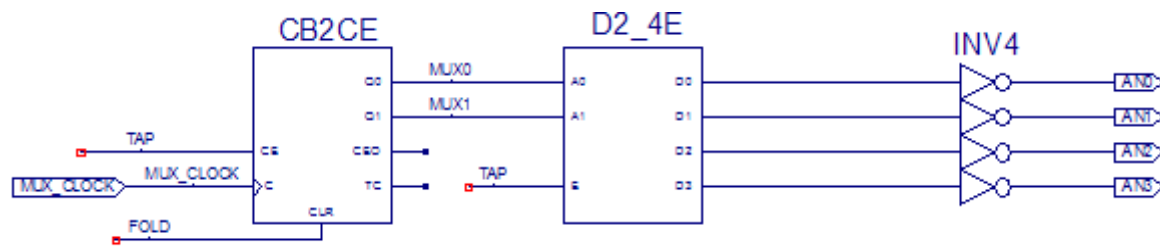
A négy darab 7-szegmens kijelző meghajtásához szükségünk van 4 számlálóra, melyeknek a be és kimeneteit célszerű elnevezni későbbi felhasználás céljára – az átláthatóság érdekében nem ajánlatos a „direktbe kötés”.

– Multiplexer blokk:



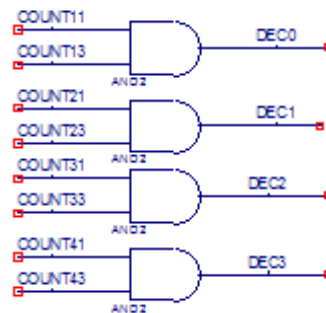
Mivel egy buszra van rákötve mind a 4 kijelző, ezért szükségünk van egy áramkörre, mely egyszer az egyik, egyszer a másik stb. adatot küldi ki a buszra.

– **Mux-kijelző vezérlő blokk:**



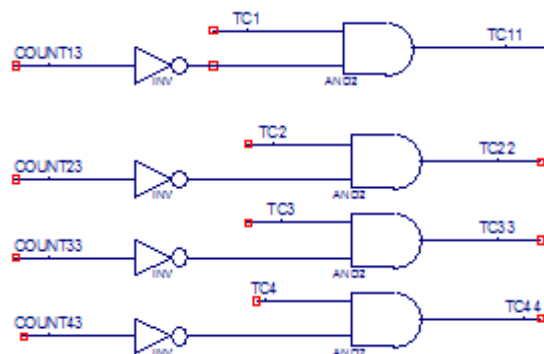
A MUX0, MUX1 vonalak csatlakozna az S0, S1 multiplexer vezérlő bemenetekre, ennek függvényében (ti. Hogy épp melyik számláló kimeneteit csatlakoztattuk a buszra) kell be/kikapcsolni az AN0-3 lábakkal az egyes kijelzőket. Fontos, egyszerre csak egy kijelző lehet aktív, és mivel közös anódos kijelző van a gyakorlópanelen, ezért aktív nullás mind a vezérlés, mind a buszra történő kiírás.

– **Decimális carry képző blokk:**



A részáramkör Q2, Q4 ÉS kapcsolatával létrehozza számunkra a decimális kijelzést lehetővé tévő Carry-t.

– **Hazárd kompenzáló blokk:**

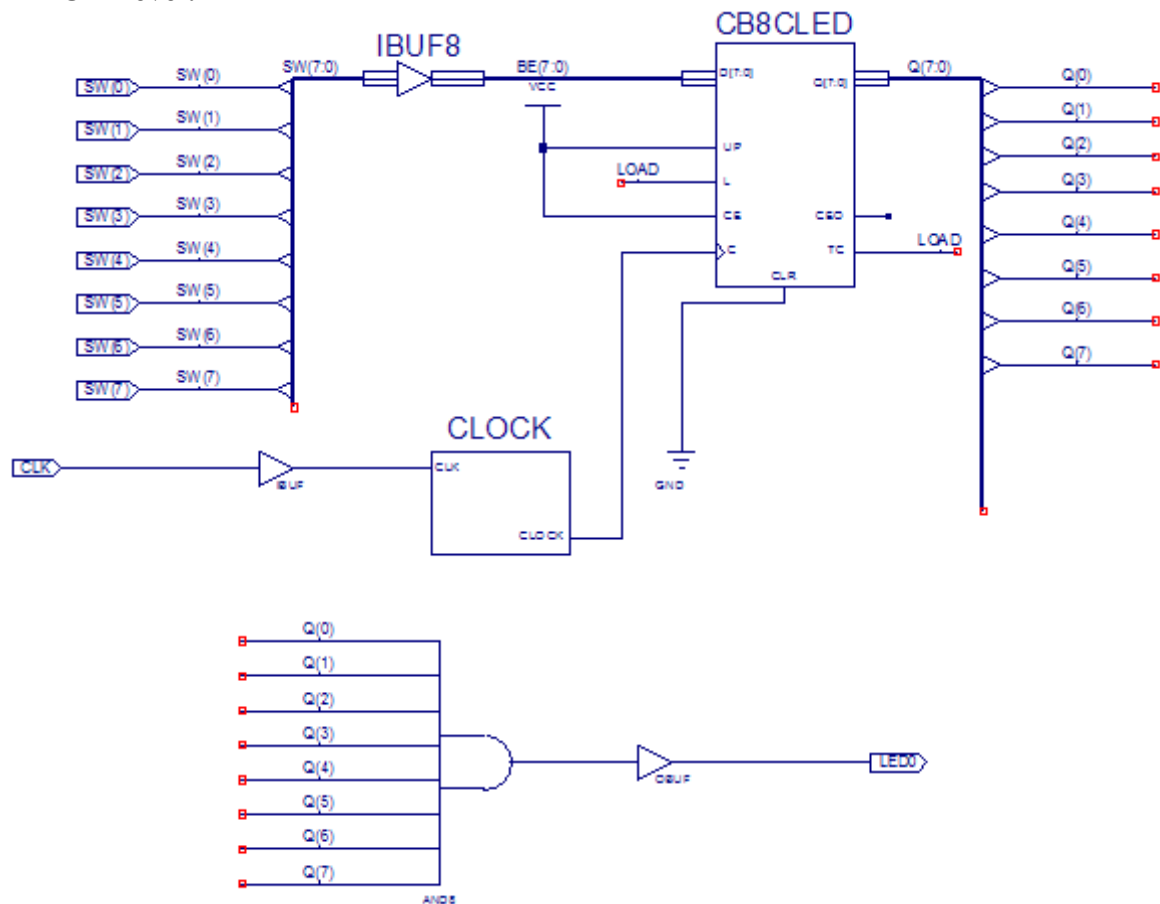


Az áramkör csak akkor engedi érvényesülni a Carry-t, ha a Q3 kimenet nem aktív. Amennyiben az áramkört gyakorlati elemekből raktuk volna össze nem kellene ez a részáramkör, hiszen Carry nem érkezhetsz, csak F-ből 0-ba váltáskor (ekkor egy fél órajelet másol a Carry kimenetre a számláló). Elméletben a TC kimeneteket közvetlenül beköthetnénk a Decimális/Hexa váltó blokkba, azonban ha ezt tesszük E-ből F-be történő váltáskor megjelenik egy „álcarry” - egy hazard. A probléma az FPGA jellegéből adódik. Az egyes alkatrészeket (tárolók, számlálók, multiplexerek, stb.) mind Nand kapuk megfelelő összekötésével hozzuk létre. Ezen alkatrészek legegyszerűbb formában hazardmentesítve kerülnek implementálásra. Amikor egyes gyári, vagy saját alkatrészeket összekötünk gyakran lehetőség nyílik optimalizálásra, egyszerűsítésre, melyet a fordító el is végez. A kialakult optimalizált áramkör hazardmentesítése azonban már nem történik meg!

A fenti áramkör részlet egy a gyakorlatban előszeretettel alkalmazott probléma megoldási sémát mutat be. Nem a hiba okát szüntettük ugyanis meg, hanem annak következményét. Ez mindig egyszerűbb, és gyorsabb mint a hibát elemezni, a lehetséges okokat számba venni, valamint azokat kiküszöbölni. Amennyiben az áramkör részletet nem alkalmazzuk a gyakorlópanelen azt vesszük észre, hogy a számláló váltás nem F-nél, hanem E-nél történik. Rövid gondolkodás, valamint próbálkozás után (melynek során a multiplexert kihagyva közvetlenül kötjük a TC1 kimenetet a CLOCK2 bemenetre), aránylag gyorsan rájövünk a probléma mibenlétére, és annak egyszerű semlegesítésére. A hiba okának elemzését azonban nem szabad „megspórolni”. Gondoljunk bele, milyen nehéz dolgunk lenne egy ennél sokkal bonyolultabb áramkör esetén - ott már elképzelhető, hogy ilyen egyszerű megoldásokat nem alkalmazhatunk, illetve maga a hiba, és az okok felderítése is sokkalta nehezebb feladat lenne.

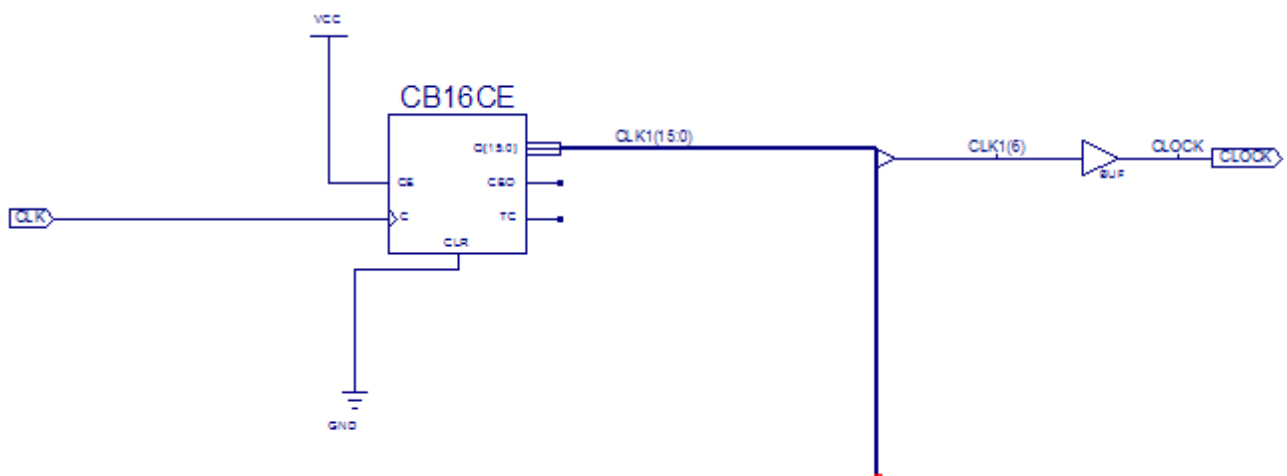
PWM:

TOP Level:



Az impulzushosszt az SW0-SW7 kapcsolókkal állíthatjuk bináris súlyozással.

Clock modul:



Az órajelosztásnál vegyük figyelembe, hogy a TOP Levelben lévő számláló még 256-al (8 bittel) osztja le a kapott órajelet a magas szintre amennyiben egyik kapcsoló se aktív, és a LED-nek még ilyen esetben se szabad villognia.