



Nyolcbites számláló mintaprojekt

1. Bevezető

A leírás egy nyolcbites számláló elkészítésének és tesztelésének lépéseit ismerteti. A számláló értéke az órajel felfutó élének hatására növekszik. A törlőgombbal lehet a számlálót alapértelmezett helyzetbe állítani, a beírás gombbal pedig egy új értéket lehet adni számláló kimenetére.

2. Projekt készítése

Készítsünk egy kapcsolási rajz alapú projektet és adjunk hozzá egy új kapcsolási rajz fájlt (példánkban: top.sch) és a *file_ucf_teszt.zip*-ben található ucf fájlt.

The screenshot shows the Xilinx ISE Project Navigator interface. The Design Summary window is open, displaying the following information:

top Project Status			
Project File:	clk8load.xise	Parser Errors:	No Errors
Module Name:	top	Implementation State:	New
Target Device:	xc3s100e-4cp132	Errors:	
Product Version:	ISE 13.2	Warnings:	
Design Goal:	Balanced	Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	Timing Constraints:	
Environment:		Final Timing Score:	

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report					
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

Secondary Reports		
Report Name	Status	Generated
Date Generated: 10/20/2011 - 18:15:52		



Az ucf fájlban egy **Led** és egy **sw** nevű nyolcbites port található. A ledeken a számláló bináris értéke jelenik meg, míg a kapcsolókon beállított érték a **LOAD** gomb megnyomásakor íródik a számlálóba. A **CLK** gomb adja az órajelet. A **CLEAR** gomb a számláló alaphelyzetbe állításáért felel.

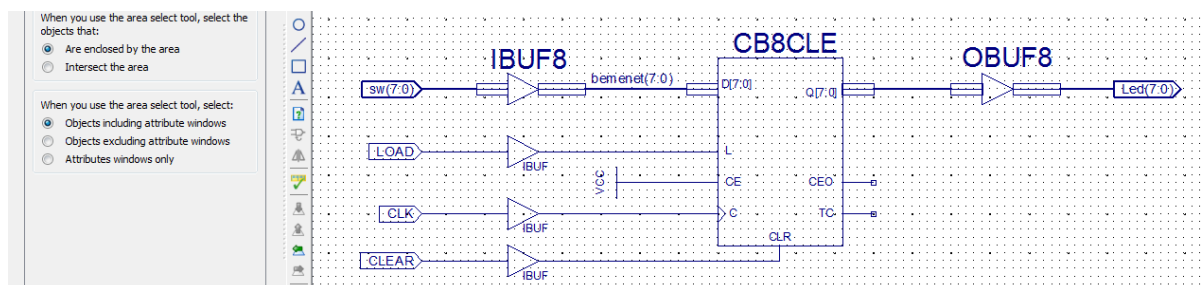
```
46
47 # Pin assignment for LEDs
48 NET "Led<7>" LOC = "G1" ; # Bank = 3, Signal name = LD7
49 NET "Led<6>" LOC = "P4" ; # Bank = 2, Signal name = LD6
50 NET "Led<5>" LOC = "N4" ; # Bank = 2, Signal name = LD5
51 NET "Led<4>" LOC = "N5" ; # Bank = 2, Signal name = LD4
52 NET "Led<3>" LOC = "F6" ; # Bank = 2, Signal name = LD3
53 NET "Led<2>" LOC = "F7" ; # Bank = 3, Signal name = LD2
54 NET "Led<1>" LOC = "M11" ; # Bank = 2, Signal name = LD1
55 NET "Led<0>" LOC = "M5" ; # Bank = 2, Signal name = LD0
56
57 # Pin assignment for SWs
58 NET "sw<7>" LOC = "N3" ; # Bank = 2, Signal name = SW7
59 NET "sw<6>" LOC = "E2" ; # Bank = 3, Signal name = SW6
60 NET "sw<5>" LOC = "F3" ; # Bank = 3, Signal name = SW5
61 NET "sw<4>" LOC = "G3" ; # Bank = 3, Signal name = SW4
62 NET "sw<3>" LOC = "B4" ; # Bank = 3, Signal name = SW3
63 NET "sw<2>" LOC = "K3" ; # Bank = 3, Signal name = SW2
64 NET "sw<1>" LOC = "L3" ; # Bank = 3, Signal name = SW1
65 NET "sw<0>" LOC = "P11" ; # Bank = 2, Signal name = SW0
66
67 //NET "UP" LOC = "A7" ; # Bank = 1, Signal name = BTN3
68 NET "CLEAR" LOC = "M4" ; # Bank = 0, Signal name = BTN2
69 NET "LOAD" LOC = "C11" ; # Bank = 2, Signal name = BTN1
70 NET "CLK" LOC = "G12" ; # Bank = 0, Signal name = BTN0
71
72 NET "CLK" CLOCK_DEDICATED_ROUTE = FALSE;
```

A kapcsolási rajzon megadott portnév

Az FPGA ki- és bemeneti lábai

Eredeti jelnév. Ezek vannak a Basys FPGA fejlesztőpanelre felírva a gombok, LED-ek mellé.

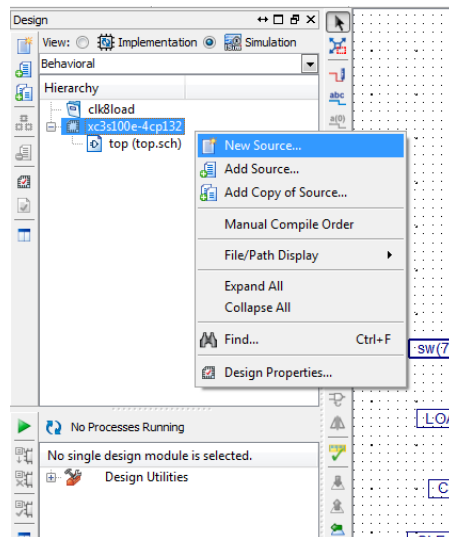
A kapcsolat elkészítése után a ki- és bemeneti portokat az alábbi ábra szerint nevezzük el. Ha más neveket adunk a portoknak, akkor az ucf fájlban annak megfelelően át kell nevezni őket.



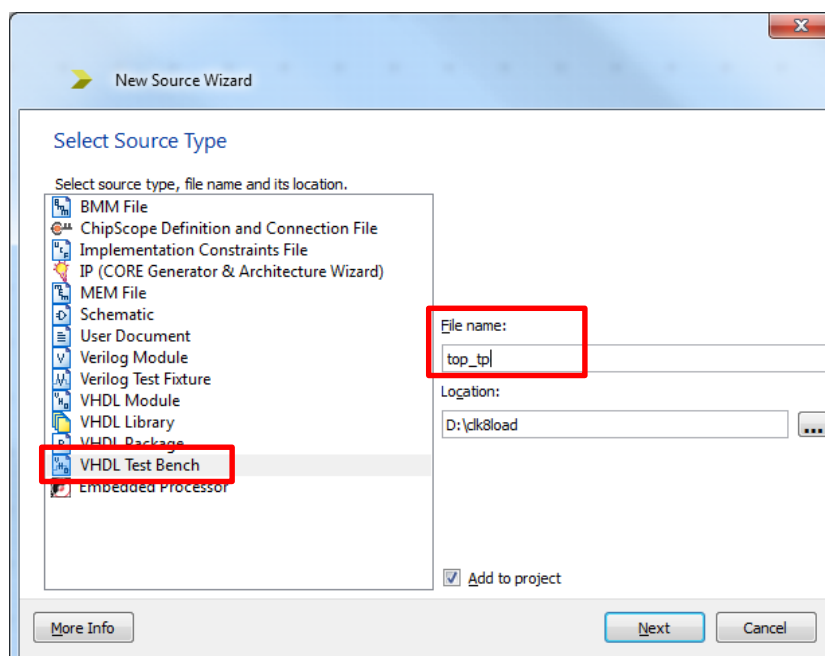


3. Tesztfájl

Az elkészült projekthez szimulációs fájl a „Design/Simulation” ablakban, az „Add New Source” opcióval lehet hozzáadni. Fontos hogy minden port nevet adjunk meg a kapcsolási rajzon a szimulációs fájl elkészítése előtt, mert a program az aktuális port neveket használja fel a sablon elkészítésére.



Ezután a „VHDL Test Bench” opciót válasszuk és adjuk meg a tesztfájl nevét.





A „Next” gomb lenyomása után a program megkérdezi melyik forrásfájlhoz készül a tesztfájl. Itt válasszuk a fő kapcsolási rajzot (példánkban: top.sch). Az ISE által generált sablon fájl tartalmazza a ki- és bemeneti portokat és a VHDL fájl keretét. Ehhez a sablonhoz kell hozzáadni a vizsgáló jelek definícióit.

```
13 -- stimulus for your design.
14 --
15 LIBRARY ieee;
16 USE ieee.std_logic_1164.ALL;
17 USE ieee.numeric_std.ALL;
18 LIBRARY UNISIM;
19 USE UNISIM.Vcomponents.ALL;
20 ENTITY top_top_sch_tb IS
21 END top_top_sch_tb;
22 ARCHITECTURE behavioral OF top_top_sch_tb IS
23
24     COMPONENT top
25     PORT( Led      : OUT  STD_LOGIC_VECTOR (7 DOWNTO 0);
26           sw       : IN   STD_LOGIC_VECTOR (7 DOWNTO 0);
27           LOAD     : IN   STD_LOGIC;
28           CLK      : IN   STD_LOGIC;
29           CLEAR    : IN   STD_LOGIC);
30     END COMPONENT;
31
32     SIGNAL Led      : STD_LOGIC_VECTOR (7 DOWNTO 0);
33     SIGNAL sw       : STD_LOGIC_VECTOR (7 DOWNTO 0);
34     SIGNAL LOAD     : STD_LOGIC;
35     SIGNAL CLK      : STD_LOGIC;
36     SIGNAL CLEAR    : STD_LOGIC;
37
38 BEGIN
39
40     UUT: top PORT MAP(
41         Led => Led,
42         sw  => sw,
43         LOAD => LOAD,
44         CLK => CLK,
45         CLEAR => CLEAR
46     );
47
48 -- *** Test Bench - User Defined Section ***
49 tb : PROCESS
50 BEGIN
51     WAIT; -- will wait forever
52 END PROCESS;
53 -- *** End Test Bench - User Defined Section ***
54
55 END;
```

Portok megadása

Jelek, konstansok...

#1 -- Órajel periódusidejének definiálása
constant clk_period : time := 1 us;

#1 -- Órajel előállítás
clk_process : process
begin
 CLK <= '0';
 wait for clk_period/2;
 CLK <= '1';
 wait for clk_period/2;
end process;

#2 Ide kerül a szimuláló jelek előállítása

Órajel előállítás az „Órajel periódusidejének definiálása” és az „Órajel előállítás” kódrészek megfelelő helyre történő beírásával lehetséges. Ez a fenti ábrán #1-el jelölve látható. Példánkban a *constant clk_period : time := 1 us;* egy mikro szekundumos órajel periódusidőt definiál. A *clk_process : process begin ... end* az órajel tényleges előállítása. Amennyiben nem CLK az órajel port neve, módosítani kell az általunk használt névre.



A fenti kiegészítéseket elvégezve, a szimuláció során a CLK bemeneti porton egy 1 us periódusidejű órajel jelenik meg. A kiegészítések az alábbi ábrán pirossal bekeretezve láthatóak.

```
35 SIGNAL LOAD : STD_LOGIC;  
36 SIGNAL CLK : STD_LOGIC;  
37 SIGNAL CLEAR : STD_LOGIC;  
38 -- Clock period definitions  
39 constant clk_period : time := 1 us;  
40  
41 BEGIN  
42  
43 UUT: top PORT MAP(  
44     Led => Led,  
45     sw => sw,  
46     LOAD => LOAD,  
47     CLK => CLK,  
48     CLEAR => CLEAR  
49 );  
50  
51 -- Clock process definitions  
52 clk_process :process  
53 begin  
54     CLK <= '0';  
55     wait for clk_period/2;  
56     CLK <= '1';  
57     wait for clk_period/2;  
58 end process;  
59  
60 -- *** Test Bench - User Defined Section ***  
61 tb : PROCESS  
62 BEGIN
```

A következő ábrán látható kiegészítésekkel előállítjuk a **LOAD**, **sw** és **CLEAR** bemenetekre adott szimulációs jeleket.

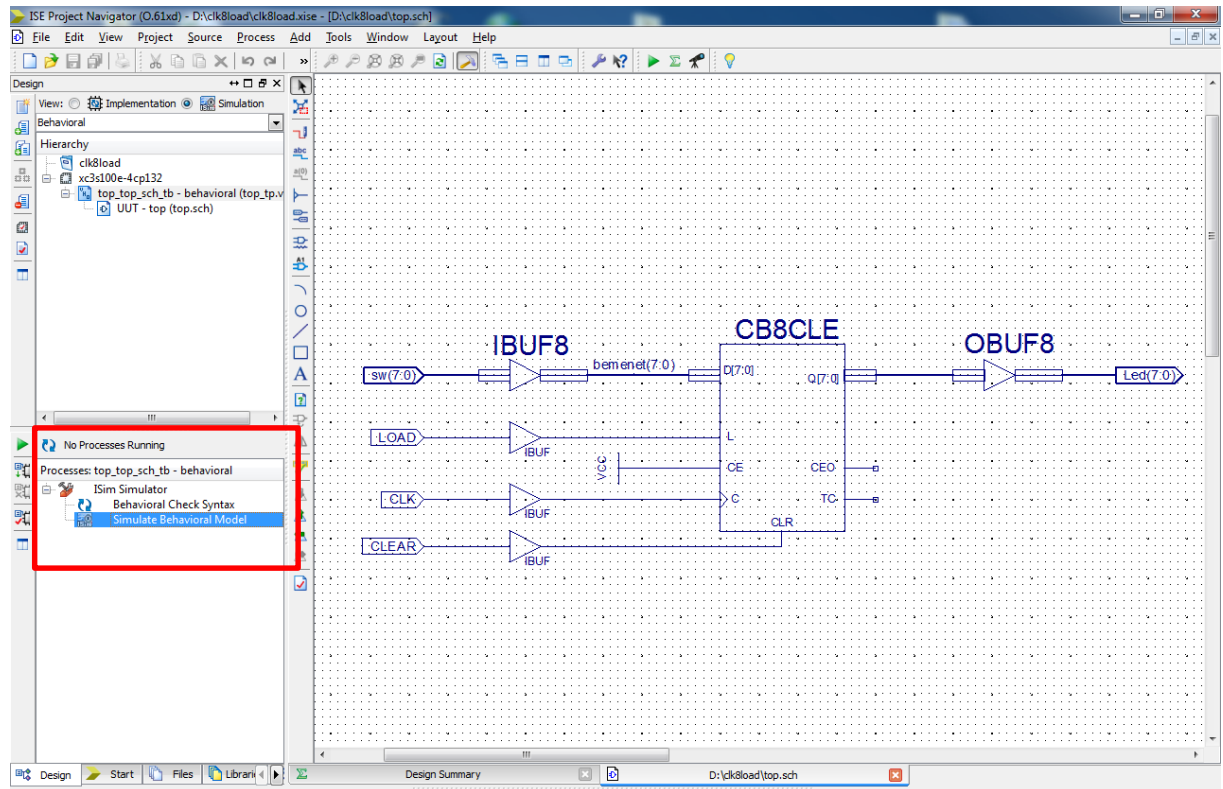
```
61 -- *** Test Bench - User Defined Section ***  
62 tb : PROCESS  
63 BEGIN  
64  
65     LOAD <= '0'; --a load jel 0  
66     sw <= "000000000"; --a párhuzamos adatbemenet 0  
67     CLEAR <= '0'; --nincs törlés  
68  
69     wait for 5 us; --5 mikroszekundum szünet  
70  
71     CLEAR <= '1'; --a számláló törlése  
72     wait for 1 us; --1us idejig lesz a CLEAR 1-es  
73     CLEAR <= '0';  
74  
75     wait for 8 us; --5 mikroszekundum szünet  
76  
77     sw <= "00000010"; --a párhuzamos adatbemenet 2  
78     wait for 1 us;  
79     LOAD <= '1'; --a load jel 1, beíródik a bemenet  
80     wait for 1 us;  
81     LOAD <= '0'; --a load jel 1us ideig volt 1-es  
82  
83     wait for 20 us; --5 mikroszekundum szünet  
84  
85     WAIT; -- will wait forever  
86 END PROCESS;  
87 -- *** End Test Bench - User Defined Section ***  
88  
89 END;
```

A jelek aktuális időpontbeli értékét a „<=” utasítással lehet megadni. Buszvezetékeknél „...”-ok, szimpla jelvezetékeknél ’.’-ok között kell megadni a jel értékét. A jelek beállítása után a **wait for** paranccsal adhatjuk meg mennyi ideig maradjanak meg a beállított értékek.

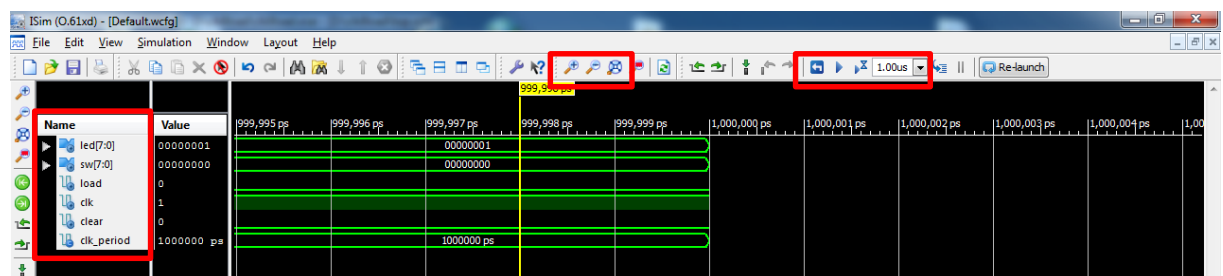


4. Szimuláció

A szimulációs fájl (tesztbench, példánkban: top_top_sch_tb) kiválasztása után a szimulációt a „Simulate Behavioral Model” parancssal lehet indítani.

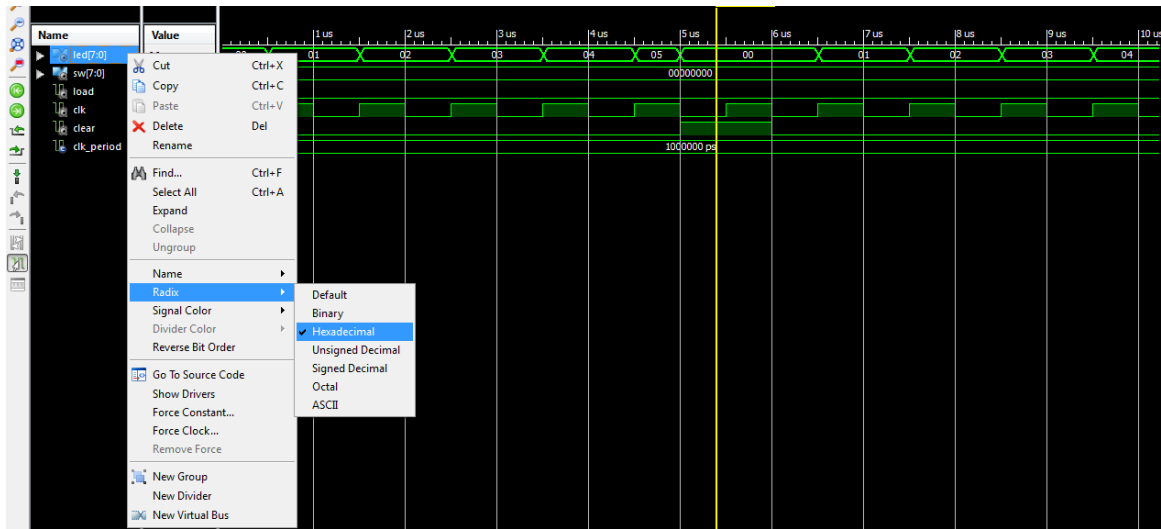


A szimuláció elindítás után a tesztfájlban szereplő portok, jelek láthatóak. A **Zoom In**, **Zoom Out**, **Zoom to Full View** ikonokkal (az alábbi ábrán a 2. piros téglalappal jelölt rész) tudjuk a megjelenített időtartományt módosítani. A **Restart** ikon újraindítja a szimulátort, törli a megjelenített idődiagramot, **Run for the time specified on toolbar** ikon lenyomásakor a szimuláció a beállított időegységnyit lép előre. (az alábbi ábrán a 3. piros téglalappal jelölt részben)





A busz értékek kijelzési formáját (hexadecimális, bináris, stb.) az alábbi ábrán látható módon lehet megváltoztatni.



Lehetőség van a szimuláció során egyéb jelek megjelenítésére (pl. belső jelek, alkatrész portok...). Az **Instance and Process Name** ablakban ki lehet választani egy alkatrészt, és az **Object** ablakban annak egy jelét. Ezeket a segédablakokat a **View/Panel** menüben lehet hozzáadni a szimulációs felülethez.

Az **Object** ablakban kiválasztott jelet a szimulációs ablakhoz adhatjuk (drag and drop).

