



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Elektronikus Eszközök Tanszéke

MIKROELEKTRONIKA LABORATÓRIUMI SEGÉDANYAG

2. rész

Tervezőrendszerek használata

Dr. Horváth Péter, Dr. Takács Gábor,
Dr. Szabó Péter Gábor, Dr. Timár András



2023

Tartalomjegyzék

Tartalomjegyzék.....	2
1. Bevezetés.....	3
2. Szintézis FPGA technológiára.....	4

1. Bevezetés

Az alábbi elektronikus jegyzet a Mikroelektronika című tárgy laborfoglalkozásaihoz tartozó pontos mérési utasításokat tartalmazza. A laborok helyszíne az Elektronikus Eszközök Tanszéke, azon belül a VLSI IC tervező labor (QB310-QB311). A konkrét feladatok ismertetésén túlmenően néhány irányelvet és tanácsot is összegyűjtöttünk a számítógépes laborban való munkavégzéssel kapcsolatban:

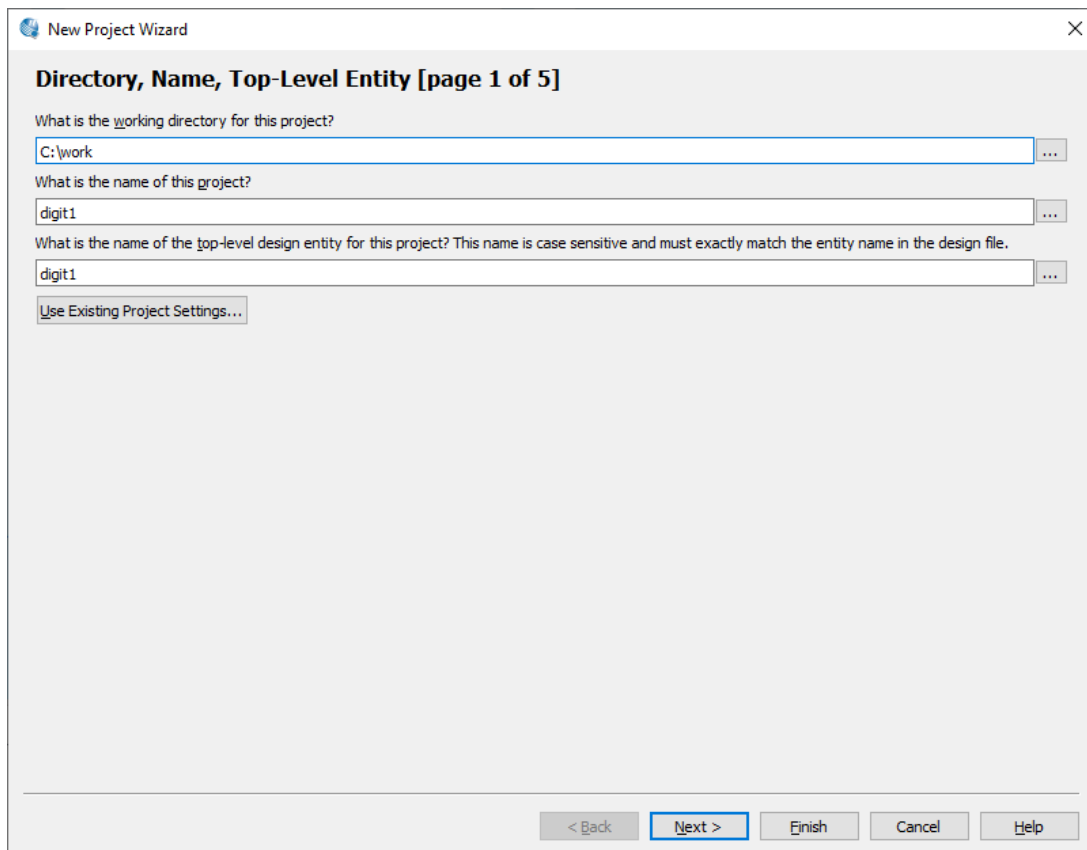
- A félév során CAD-szoftverek egész sorát fogjuk használni, amelyek eltérő futtatási környezeteket igényelnek. A laborban található számítógépeken Windows 10 operációs rendszer fut. A bejelentkezés a **Student** felhasználóval, jelszó nélkül lehetséges.
- Bejelentkezés után a munkával kapcsolatos fájlokat (pl. az EDU-rendszerből letöltött segédanyagokat, az adott laborhoz kapcsolódó projektfájlokat) érdemes az asztalon/home könyvtárban (Desktop vagy Home) tárolni. Ehhez minden felhasználónak garantáltan van írási/olvasási joga, továbbá a fájlokat itt könnyű megtalálni.
- A számítógép kikapcsolásakor/újraindításakor az felhasználók által végzett minden művelet eredménye - beleértve az újonnan létrehozott fájlokat - elvész! A kritikus fájlokról (saját forrásfájlok, jegyzőkönyvek) érdemes valamilyen formában biztonsági másolatot készíteni (e-mail, dropbox, drive, stb.). Pendrive használata nem javasolt!
- A laborban a számítógépes munkaállomások hardver elemeit elmozdítani (monitor, UTP patch-kábel kihúzása, egér, billentyűzet kihúzása) szigorúan tilos!
- A belső hálózatra saját számítógépet csatlakoztatni szigorúan tilos!
- A labor végeztével a munkaállomáson rendet kell tenni, biztosítva a következő csoportok számára a kulturált körülményeket.
- **A laborvezető engedélye nélküli hardveres változtatás (lásd szigorúan tilos rendelkezések), illetve operációs rendszeren történő módosítás (esetleges biztonsági rések kihasználása, rendszer törlése, stb.) rongálásnak minősül és fegyelmi eljárást von maga után!**

2. Szintézis FPGA technológiára

A laboratóriumi gyakorlat során az *Altera Quartus II* fejlesztői környezetét és a *DE0* fejlesztőkártyát használjuk. A gyakorlat során megvalósítandó feladatokat az *Altera DE0* fejlesztőkártyán rendelkezésre álló hardverkörnyezethez illesztettük.

Az *Altera Quartus II* egy integrált fejlesztőkörnyezet, amely hatékony felületet nyújt a szintézis lépéseinek végrehajtásához és a fejlesztés során minden lépés elvégezhető a grafikus felhasználói felületen keresztül.

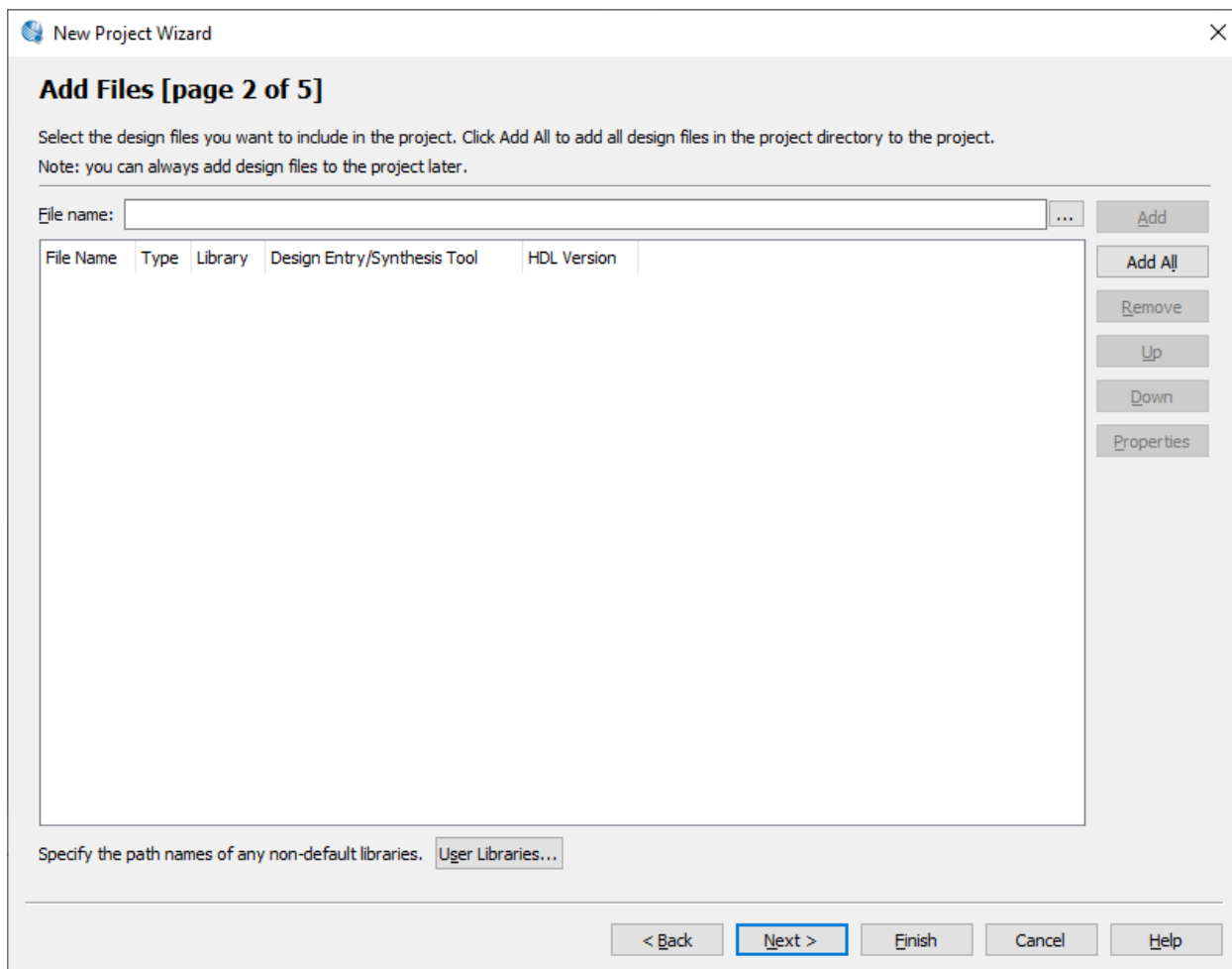
A szintézist egy új projekt létrehozásával kezdjük. Új projektet a **File** menü **New Project Wizard...** parancsával hozhatunk létre. Ekkor egy párbeszédablak-sorozat segítségével beállíthatjuk a projekt legfontosabb részleteit. Az első ablak egy bevezetés (*Introduction*), amely bemutatja, hogy a varázsló milyen beállításokat tesz lehetővé. A **Next** gomb segítségével lépünk át a következő ablakra, ahol a projekt helyét és nevét adhatjuk meg, valamint azt, hogy a HDL forrás hierarchiájában melyik a legfelső szintű (*toplevel*) modul, amelyiket szintetizálni szeretnénk (2-1. ábra).



2-1. ábra Az új projekt helye, neve és a toplevel modul neve

Az új projektet a **C:\work** könyvtárban kell létrehozni a jogosultsági problémák elkerülése céljából.

A következő ablakban lehetőségünk van forrásfájlok hozzáadására. Forrásfájlokat természetesen a projekt létrehozása után is tudunk hozzáadni.



2-2. ábra Forrásfájlok hozzáadása

Forrásfájlok hozzáadása nélkül a **Next** gombra kattintva az FPGA eszközcsalád és azon belül a konkrét eszköz kiválasztására van lehetőség. A *DE0* fejlesztőkártyán egy **Cyclone III FPGA** kapott helyet, amelynek típusazonosítója a tokról leolvasható: **EP3C16F484C6**. A grafikus felület az eszköz gyorsabb kikeresése céljából szűrők beállítását teszi lehetővé. Az eszközcsalád (*Family*) mezőben válasszuk a **Cyclone III**-at, a tokozásnál (*Package*) **FBGA**-t (*Flip-chip Ball Grid Array*), a kivezetések számánál (*Pin count*) **484**-et és a sebességnél (*Speed grade*) - amely a gyártástechnológiára utal - **6**-ot. Ezeknek a szűrőbeállításoknak négy különböző eszköz felel meg, ezek közül válasszuk az elsőt (2-3. ábra)!

New Project Wizard

Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

Device family

Family: Cyclone III

Devices: All

Target device

☐ Auto device selected by the Fitter

☒ Specific device selected in 'Available devices' list

☐ Other: n/a

Show in 'Available devices' list

Package: FBGA

Pin count: 484

Speed grade: 6

Name filter:

☒ Show advanced devices ☐ HardCopy compatible only

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit elements	PLL	bal Clo
EP3C16F484C6	1.2V	15408	347	516096	112	4	20
EP3C40F484C6	1.2V	39600	332	1161216	252	4	20
EP3C55F484C6	1.2V	55856	328	2396160	312	4	20
EP3C80F484C6	1.2V	81264	296	2810880	488	4	20

Companion device

HardCopy:

☐ Limit DSP & RAM to HardCopy device resources

< Back

Next >

Finish

Cancel

Help

2-3. ábra Eszközcsalád és eszköz kiválasztása

Az eszköz kiválasztása után már csak egy beállítás van hátra. Bár az áramkörünk működését a fejlesztőkártyán fogjuk ellenőrizni, a teljesség kedvéért bemutatjuk, hogy miként lehet előállítani azokat a fájlokat, amelyek a standard cellás szintézis esetén az időzítési szimulációhoz szükségesek, nevezetesen a post-layout HDL modellt (amelyet FPGA technológia esetén post-place&route modellnek hívnak) és az időzítési viszonyokat leíró SDF fájlt. Ahhoz, hogy ezek a fájlok a nekünk megfelelő formátumban generálódjanak, ki kell választanunk a verifikációs környezetet és a generált kapusintű modell nyelvét. Ehhez az **EDA Tool Settings** ablak **Simulation** sorának **Tool Name** mezőjében állítsuk be a **ModelSim**-et, a **Format(s)** mezőben pedig válasszuk a **Verilog**-ot (2-4. ábra).

New Project Wizard

EDA Tool Settings [page 4 of 5]

Specify the other EDA tools used with the Quartus II software to develop your project.

EDA tools:

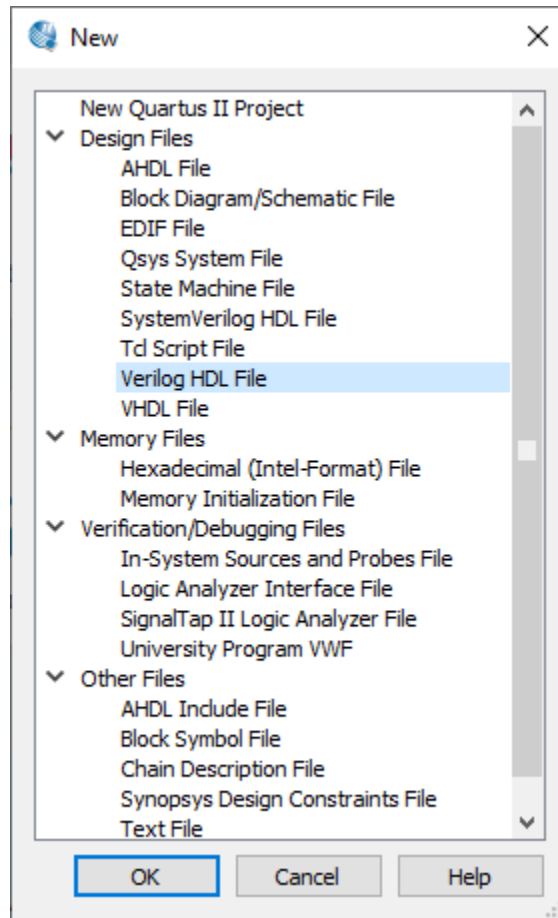
Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synthesis	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	ModelSim-Altera	Verilog HDL	<input type="checkbox"/> Run gate-level simulation automatically after compilation
Formal Verification	<None>		
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

< Back **Next >** Finish Cancel Help

2-4. ábra A kimenetimodell-generáláshoz szükséges beállítások

Ezek után a **Next** gombra kattintva egy összegzést kapunk, amelyet a **Finish** gombbal hagyhatunk jóvá.

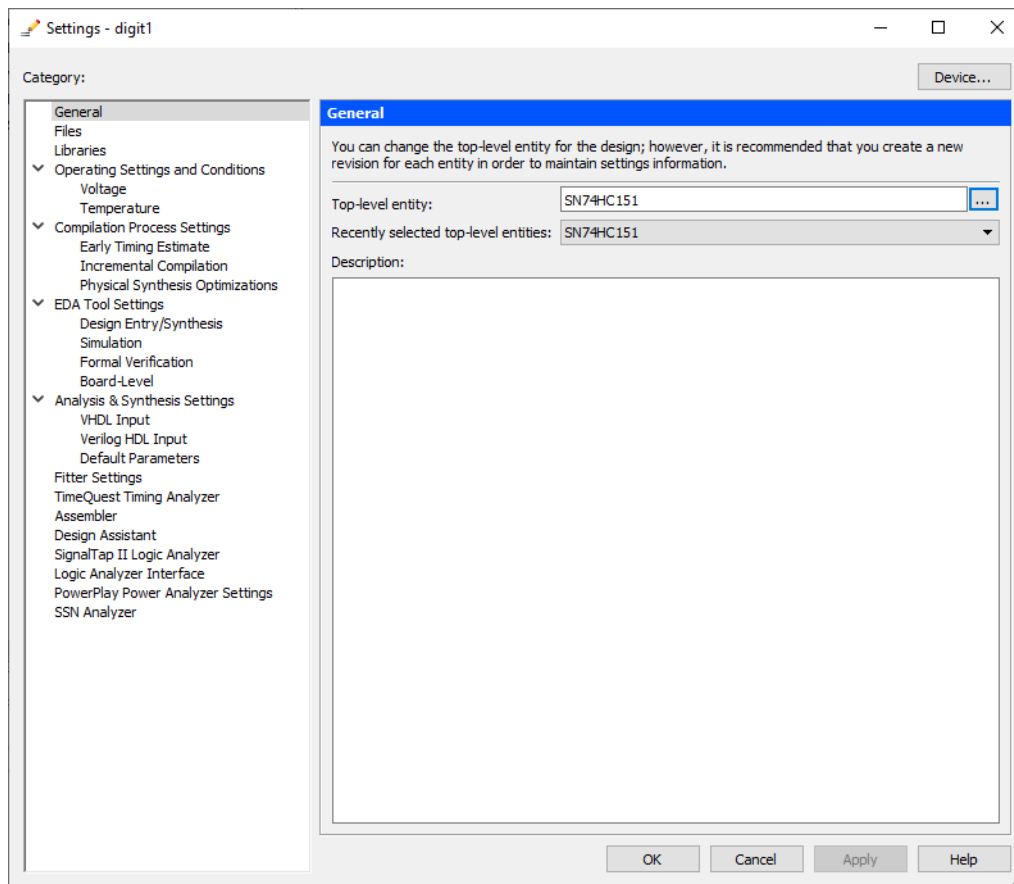
Az így elkészített projekt jelenleg nem tartalmaz hardver leíró nyelven (HDL) készült forrásfájl. Adjunk hozzá a projekthez egy új Verilog HDL nyelvű fájlt (2-5. ábra). Ebbe a fájlba kell leírni a digitális áramkör HDL leírását a labor sillabusz alapján. Az egyszerűség kedvéért másoljuk be a sillabuszban kiadott Verilog HDL nyelvű leírást az új fájlba és kattintsunk a **Mentés** gombra.



2-5. ábra Verilog forrásfájl hozzáadása a projekthez

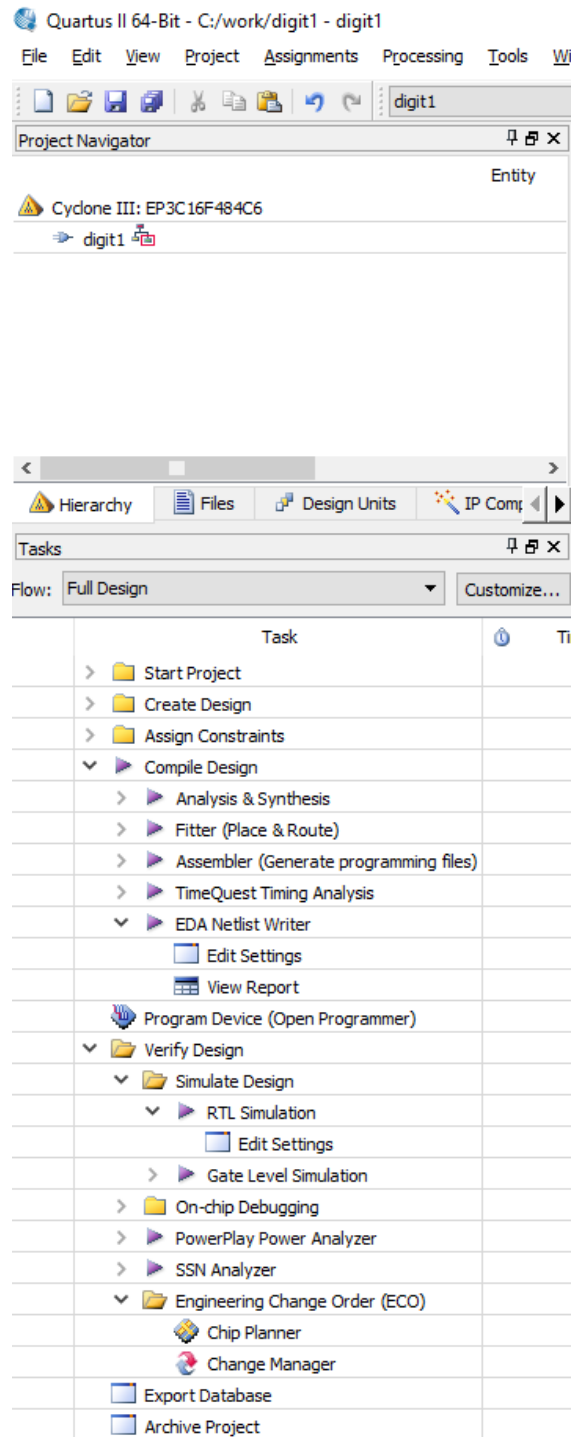
A sillabuszban szereplő Verilog HDL modulok neve minden feladat esetén más: pl. **LogikaiErtek**, **Vezetek**, **SN74HC151**, stb. a megvalósított funkciónak megfelelően.

A Quartus II fejlesztő környezet számára meg kell határozni az ún. **top-level** modul nevét. Egy bonyolultabb áramkört tervezés esetén a különböző logikai/sorrendi funkciókat külön modulokba érdemes kiszervezni. Ezek a modulok egy hierarchiába tartoznak, ahol egyes modulok más modulokba épülhetnek be. A top-level modul az a modul, ami a modul-hierarchia legfelsőbb szintjén van. A szimuláció és szintézis lépéseknek tudniuk kell, hogy az áramkör melyik modulja a legfelsőbb elem. Ennek nevét a **Settings** menüben kell beállítanunk a Verilog HDL leírás **modul nevével megegyező** értékre (2-6. ábra).



2-6. ábra Top-level modul kiválasztása

A grafikus felület bal oldalán található **Project Navigator** ablak tetején a HDL modell szerkezetét, alján pedig a szintézis elvégzendő részfeladatainak felsorolását láthatjuk. A **Flow** nevű legördülő menüben válasszuk a **Full Design** lehetőséget (2-7. ábra)!



2-7. ábra A Project Navigator ablak

A következő lépés a HDL modell szintaktikai ellenőrzése és szerkezetének feltérképezése. A **Project Navigator** ablakban a **Compile Design** menüpont alatt található **Analysis & Synthesis** almenüben kattintsunk duplán az **Analysis & Elaboration** parancsra!

A HDL modell beolvasása és elemzése után már ismertek a modell logikai kivezetései, amelyeket az FPGA eszköz általános célú kivezetéseéhez kell hozzárendelnünk. Ez a lépés az ún. *kivezetés-hozzárendelés (pin assignment)*. Ha egy már elkészült PCB-n (*Printed Circuit Board*) helyet foglaló FPGA-ról van szó (ahogy esetünkben is), akkor a logikai kivezetések fizikai megfelelőinek kiválasztása nem teljesen önkényes. Egy logikai kimenetet értelemszerűen egy meghajtást igénylő, míg egy logikai bemenetet egy meghajtó jellegű erőforrásra kell kapcsolnunk. A felhasznált fejlesztőkártya felhasználói dokumentációja tartalmazza azt az információt, hogy a PCB egyes erőforrásai (LED-ek, kapcsolók stb.) az FPGA melyik kivezetéséhez vannak hozzákapcsolva.

A kivezetés-hozzárendelés a Quartus II környezetben az **Assignments** menü **Pin Planner** nevű alprogramjával végezhető el (2-8. ábra).

Az eszköz kivezetései (felülnézet)

A különböző kivezetéstípusok (GPIO, órajel, reset, tápellátás) más-más szimbólummal vannak jelölve.

Node Name	Direction	Location	I/O Bank	WREF Group	Filter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair
clk	Input	PNL_G21	1	B1_N0	PNL_G21	2.5 V (default)		8mA (default)	2 (default)	
cout[7]	Output	PNL_C2	1	B1_N0	PNL_C2	2.5 V (default)		8mA (default)	2 (default)	
cout[6]	Output	PNL_C1	1	B1_N0	PNL_C1	2.5 V (default)		8mA (default)	2 (default)	
cout[5]	Output	PNL_E1	1	B1_N0	PNL_E1	2.5 V (default)		8mA (default)	2 (default)	
cout[4]	Output	PNL_F2	1	B1_N0	PNL_F2	2.5 V (default)		8mA (default)	2 (default)	
cout[3]	Output	PNL_H1	1	B1_N1	PNL_H1	2.5 V (default)		8mA (default)	2 (default)	
cout[2]	Output	PNL_J1	1	B1_N1	PNL_J1	2.5 V (default)		8mA (default)	2 (default)	
cout[1]	Output	PNL_J2	1	B1_N1	PNL_J2	2.5 V (default)		8mA (default)	2 (default)	
cout[0]	Output	PNL_J1	1	B1_N1	PNL_J1	2.5 V (default)		8mA (default)	2 (default)	
direction	Input	PNL_D2	1	B1_N0	PNL_D2	2.5 V (default)		8mA (default)		
enable	Input	PNL_E4	1	B1_N0	PNL_E4	2.5 V (default)		8mA (default)		
load_n	Input	PNL_F1	1	B1_N0	PNL_F1	2.5 V (default)		8mA (default)		
parallel_un[7]	Input	PNL_E3	1	B1_N0	PNL_E3	2.5 V (default)		8mA (default)		
parallel_un[6]	Input	PNL_F7	1	B1_N0	PNL_F7	2.5 V (default)		8mA (default)		
parallel_un[5]	Input	PNL_J7	1	B1_N1	PNL_J7	2.5 V (default)		8mA (default)		
parallel_un[4]	Input	PNL_G5	1	B1_N0	PNL_G5	2.5 V (default)		8mA (default)		
parallel_un[3]	Input	PNL_G4	1	B1_N0	PNL_G4	2.5 V (default)		8mA (default)		
parallel_un[2]	Input	PNL_H6	1	B1_N0	PNL_H6	2.5 V (default)		8mA (default)		
parallel_un[1]	Input	PNL_H5	1	B1_N0	PNL_H5	2.5 V (default)		8mA (default)		
parallel_un[0]	Input	PNL_H6	1	B1_N0	PNL_H6	2.5 V (default)		8mA (default)		
reset_n	Input	PNL_H2	1	B1_N1	PNL_H2	2.5 V (default)		8mA (default)		

A modell logikai be- és kimeneteinek fizikai tulajdonságai (meghajtóképesség, I/O szabvány, irány, hozzárendelt fizikai kivezetés stb.).

2-8. ábra Pin Planner

A kivezetésekre vonatkozó táblázatban elegendő a **Location** oszlopot kitölteni a „DE0_User_manual.pdf” 4.2-es fejezetében meghatározott PIN elnevezéseknek megfelelően.

A szillabusz **SN74HC151 8:1 multiplexer** áramköre esetén a lábkiosztást tartalmazó leíró fájlt (SN74HC151.qsf) az EDU rendszerbe is feltöltöttük. Az Import Assignments... menüpont

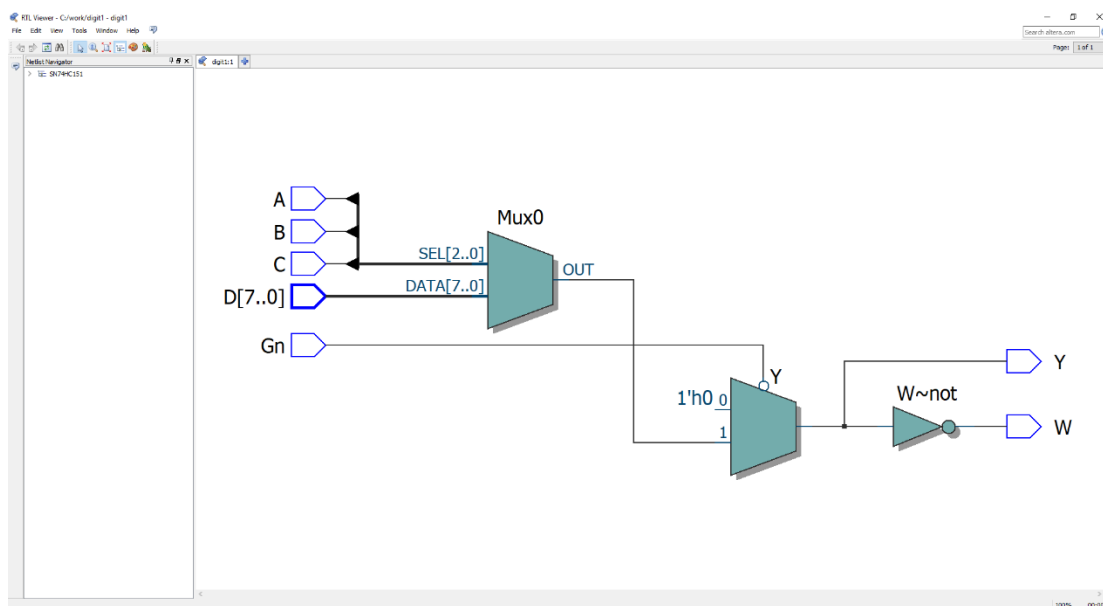
segítségével ez a lábkiosztás betölthető a Quartus II tervező rendszerbe. A gyakorlati ismeretek elmélyítése céljából azonban javasoljuk a lábak hozzárendelését kézzel, a DE0 User Manual segítségével elvégezni.

A **Pin Planner** a **File** menü **Close** parancsával zárható be.

Ezután a **Project Navigator** ablak **Compile Design** parancsára duplán kattintva a program a következő lépéseket automatikusan elvégzi:

- **Analysis and Synthesis:** Generikus szintézis és mapping.
- **Fitter (Place & Route):** Csoportképzés, elhelyezés és huzalozás.
- **Assembler (Generate programming files):** Bitfolyam előállítás.
- **TimeQuest Timing Analysis:** Statikus időzítés-analízis.
- **EDA Netlist Writer:** Post-Place&Route modell és SDF fájl generálása.

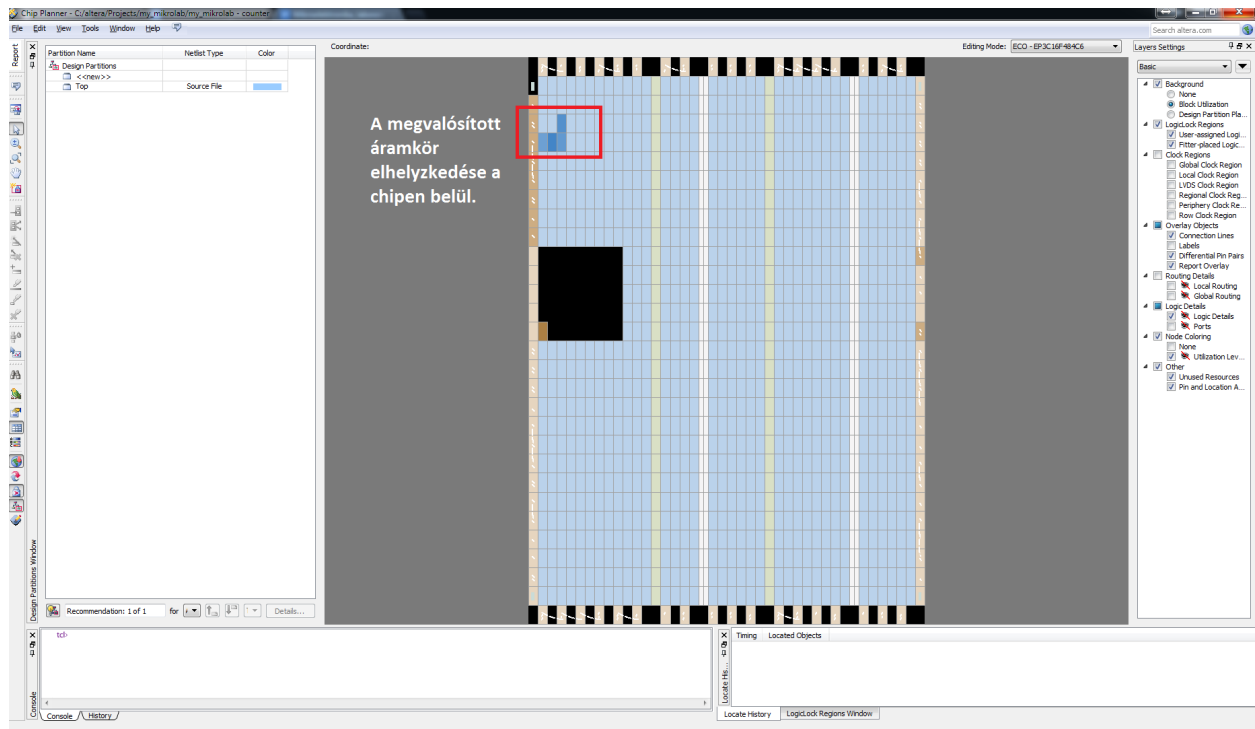
Nyissuk le a **Project Navigator** ablak **Analysis and Synthesis** pontját, azon belül pedig keressük a **Netlist Viewers** mappát, amelyben három különböző alkalmazás áll rendelkezésre a szintézis ellenőrzésére. Az **RTL viewer** - ahogy neve is utal rá - az RTL modell alapján elkészített, RTL és kapusintű erőforrásokat tartalmazó netlista (2-9. ábra).



2-9. ábra A multiplexer áramkör kapcsolási sémája

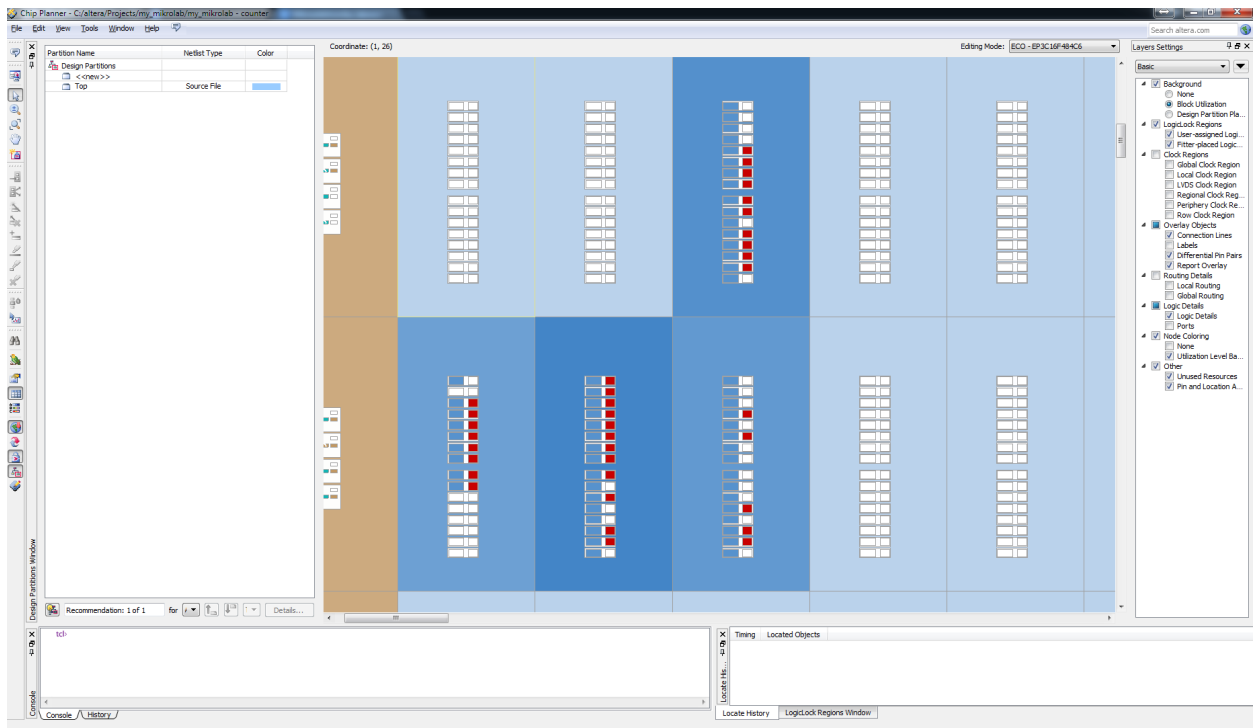
Bár összetett rendszerek esetén egy ilyen kapcsolási rajz nehezen átláthatónak tűnhet, arra kiválóan alkalmas, hogy az RTL tervező eldöntse, hogy a szintézis szoftver valóban azt az áramkört valósította meg, amelyet ő elképzelt. A **Technology Map Viewer** segítségével pedig ellenőrizhető a post-mapping modell kapcsolási rajza.

Az automatizált elhelyezés és huzalozás eredményét is megtekinthetjük a *Chip Planner* alkalmazás segítségével, amely a **Tools** menü **Chip Planner** parancsával indítható el (2-10. ábra).



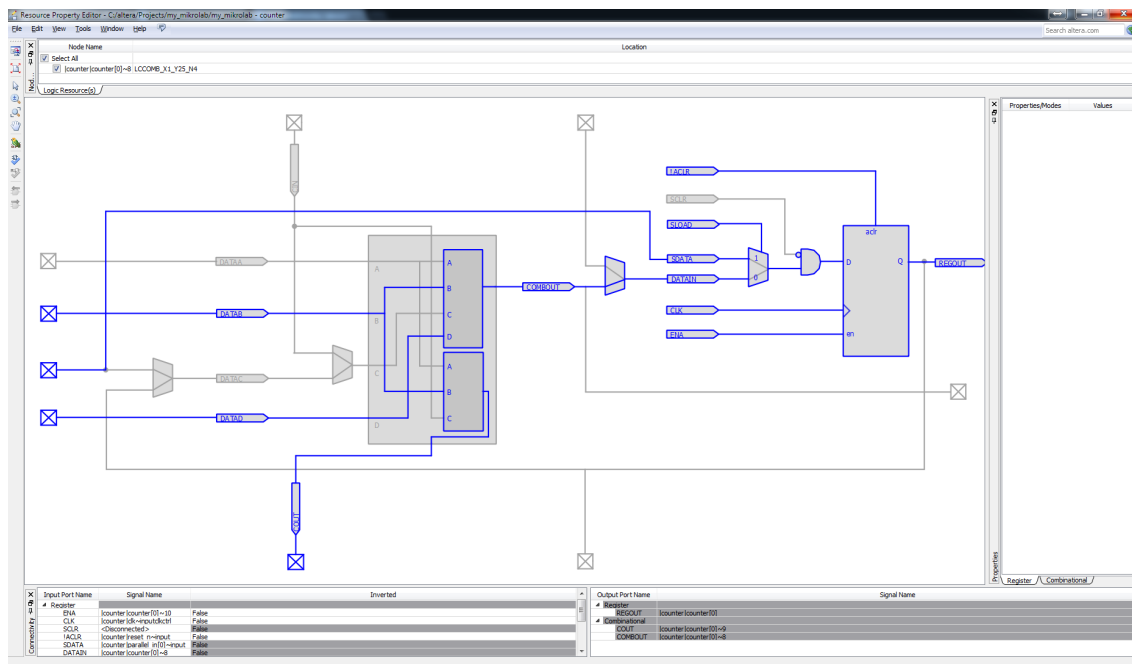
2-10. ábra *Chip Planner*

A *Chip Planner* az eszköz felülnézeti képét mutatja, kiemelve az egyes logikai blokkokat és heterogén architektúrális elemeket. Az ábrán világoskék színnel jelölt LAB-ek (Logic Array Block) árnyalata kihasználtságuktól függően változik; minél több BLE (Basic Logic Element) foglalt az adott LAB-en belül, annál sötétebb a hozzá tartozó téglalap. Látható, hogy az áramkör nem igényel sok erőforrást, egészen kis részét foglalja csak el a rendelkezésre álló LAB-eknek. Ha ráközelítünk a foglalt területekre, az áramkör szerkezete a legapróbb részletekig megfigyelhető (2-11. ábra).



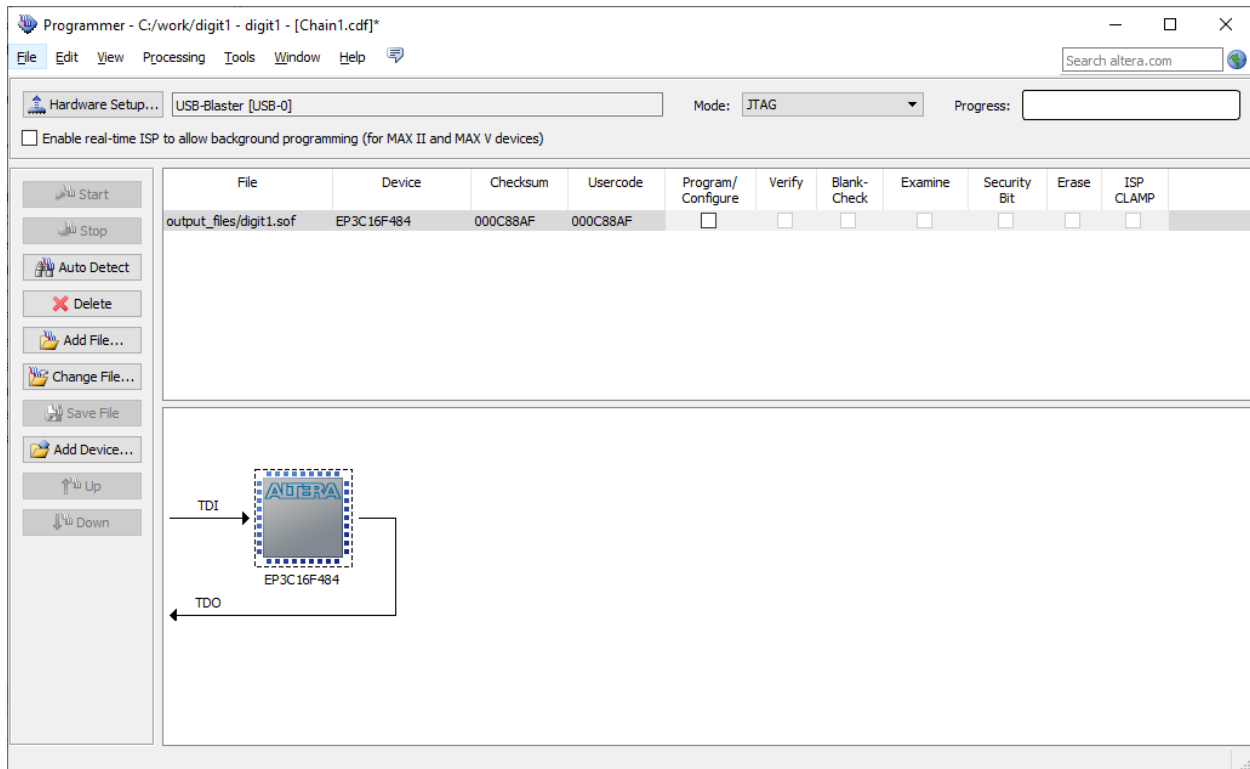
2-11. ábra Az áramkör által felhasznált LAB-ek

A 2-11. ábrán megfigyelhető a LAB-ekben helyet foglaló 16 BLE. A BLE-k egy-egy kék és piros téglalappal vannak jelölve. A piros téglalapok az adott BLE regiszterének, a kék pedig LUT-jának a kihasználtságát jelzik. Egy BLE-re duplán kattintva azt is megfigyelhetjük, hogy az adott BLE konfigurációja milyen áramkört valósít meg (2-12. ábra).



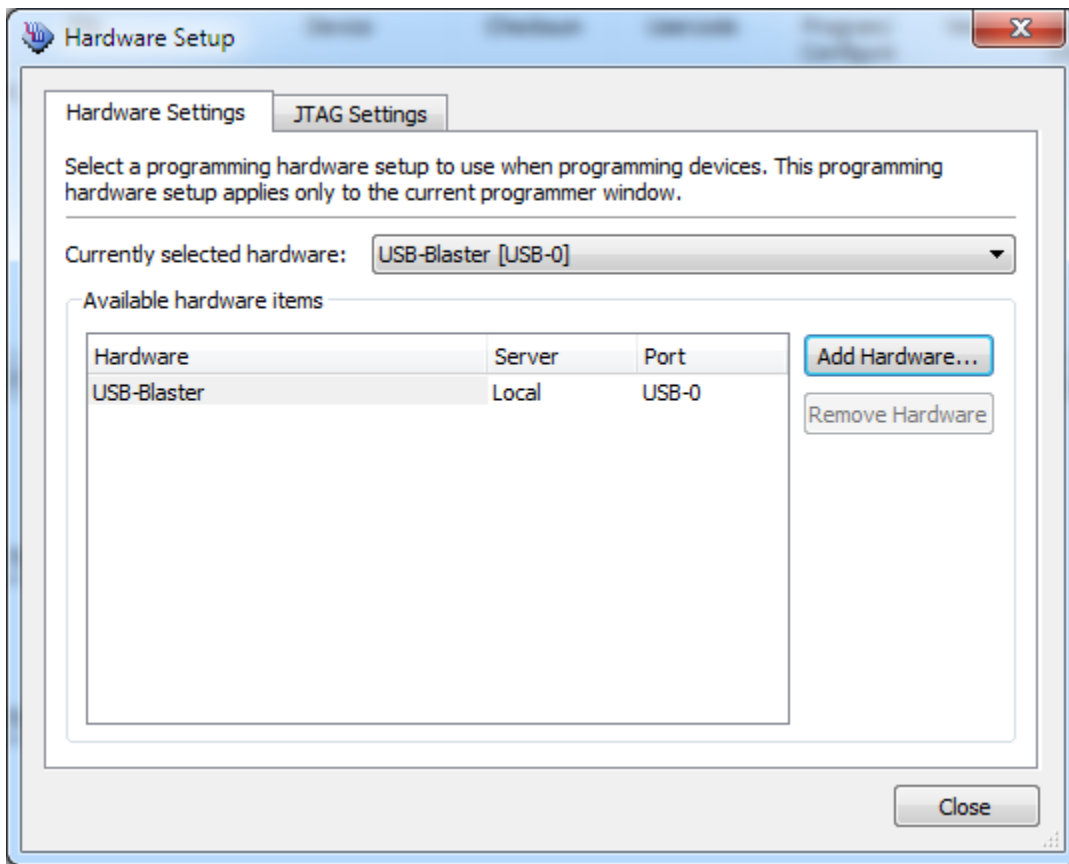
2-12. ábra BLE konfigurációja

A szintézis minden lépését elvégeztük, így nincs más hátra, mint kipróbálni az áramkört élesben. A konfigurációs memória tartalmának feltöltése az eszközre a **Tools** menü **Programmer** parancsával indítható alkalmazással lehetséges (2-13. ábra).



2-13. ábra Programmer

Ha a bal felső sarokban lévő **Hardware Setup** mezőben **No Hardware** feliratot látunk, akkor a **Hardware Setup** gombra kattintva a lehetséges hardverelemek közül válasszuk az **USB Blaster**-t (2-14. ábra).



2-14. ábra USB Blaster kiválasztása

Ezután a **Programmer** ablak felső mezőjében megjelenik az aktuális projektünkben már rendelkezésre álló programozófájl (*output_files/counter.sof*), alsó mezőjében pedig látni fogjuk azt a JTAG láncot, amelyet a program az adott programozófájjal kompatibilisnek ismert fel. Ha a programozófájl nem jelent meg automatikusan, akkor bal oldali gombsor **Add File...** gombjával kézzel is kijelölhetjük azt. A bal oldali gombsor start gombjára kattintva a letöltés elkezdődik. A sikeres letöltést a jobb felső sarokban elhelyezkedő **Progress** mező **"100% (Successful)"** felirata jelzi.