



**BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM**  
**VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR**  
**MÉRÉSTECHNIKA ÉS INFORMÁCIÓS RENDSZEREK TANSZÉK**

# **Digitális technika (VIMIAA02)**

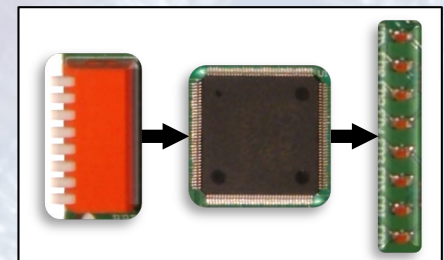
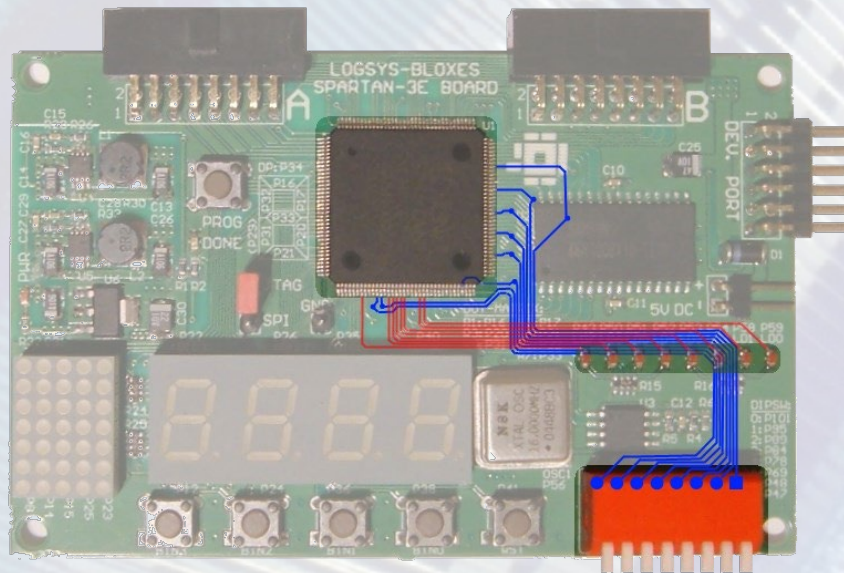
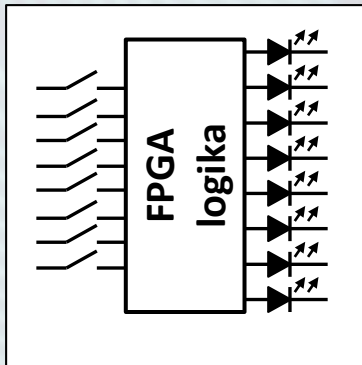
## **1. laboratórium**

**Raikovich Tamás**  
**BME MIT**

# 1. feladat: HW „Hello World!”

8 db LED vezérlése a 8 bites DIP kapcsolóval a LOGSYS Spartan-3E FPGA kártyán

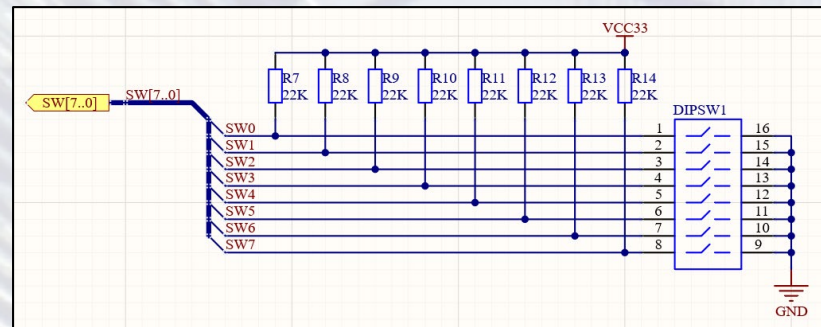
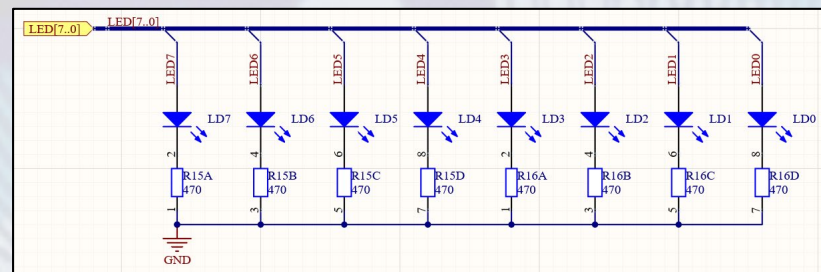
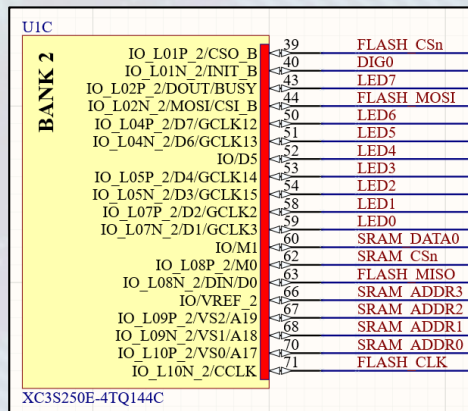
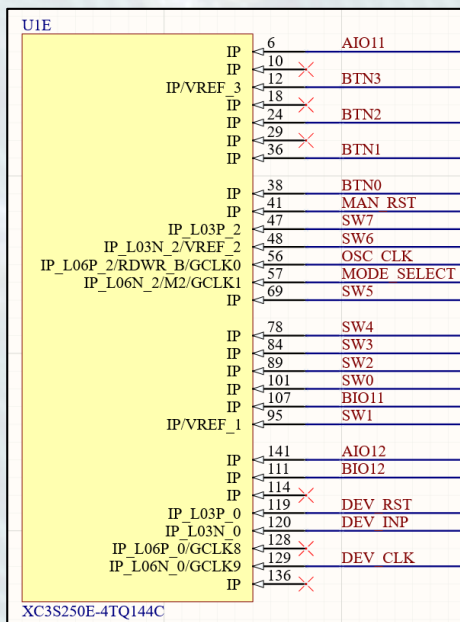
- DIP kapcsoló → FPGA bemenet: **kék huzalozás**
- FPGA kimenet → LED: **piros huzalozás**
- Mi legyen megvalósítva az FPGA logikában?





# 1. feladat: HW „Hello World!”

Az elvi kapcsolási rajz a szükséges paraméterekkel  
(nem tananyag, csak érdeklődőknek !)



LED	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
FPGA láb	P43	P50	P51	P52	P53	P54	P58	P59

Kapcsoló	7	6	5	4	3	2	1	0
FPGA láb	P47	P48	P69	P78	P84	P89	P95	P101

# 1. feladat – Szükséges Verilog ismeretek

- FPGA (Field Programmable Gate Array)
  - A programozható logikai áramkörök egyik típusa
  - Digitális rendszerek megvalósítására használható
- A megvalósítandó rendszer viselkedését a legtöbb esetben hardver leíró nyelvvel (HDL) adjuk meg
  - Mi a Verilog nyelvet fogjuk használni
- Az 1. feladathoz szükséges Verilog ismeretek
  - A Verilog modul felépítése, részei
  - A modul portjainak megadása (irány, típus, méret)
  - Folytonos értékadás az ***assign*** utasítással
  - A szimulációs tesztkörnyezet létrehozása

# 1. feladat – A projekt létrehozása

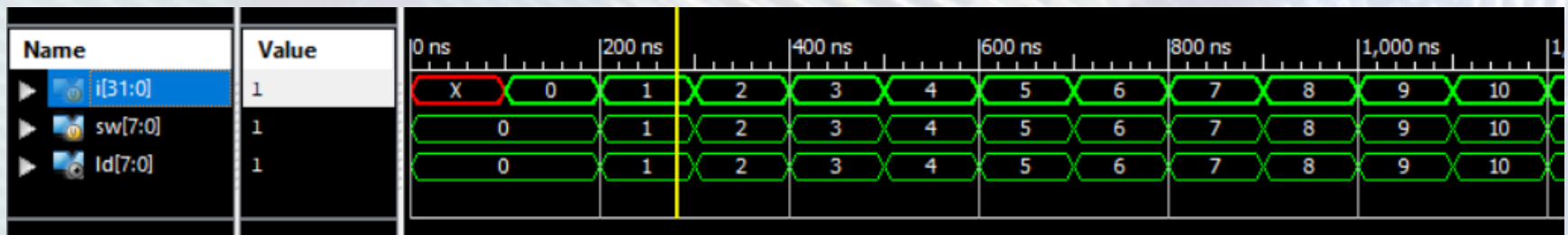
- A Xilinx ISE Design Suite 14.6 fejlesztői környezetet használjuk az FPGA fejlesztéshez
- Új projekt létrehozása
  - A projekt nevének és elérési útjának megadása
  - A használt FPGA eszköz típusának megadása  
**Family: Spartan3E**                      **Device: XC3S250E**  
**Package: TQ144**                      **Speed: -4**
- Új forrásfájl hozzáadása az üres projekthez
  - Típus: Verilog modul
  - A fájl nevének és elérési útjának megadása
  - A modul portjainak megadása (opcionális)
- Egészítsük ki a generált modul vázat
  - A LED port legyen **ld**, a kapcsoló port legyen **sw**



# 1. feladat – Tesztelés, szimuláció

A szimulátor segítségével ellenőrizhető az elkészült rendszer egészének vagy egy részének megfelelő működése a hardver nélkül is

- Bemeneti adatok: a tesztelendő modul bemenetei
  - A tesztkörnyezetben adjuk meg, hogy egymás után milyen értékek kerüljenek a bemenetekre
- Eredmény: idődiagram formájában
  - A tesztkörnyezetben (és kiegészítéssel az almodulokban) lévő belső jelek időbeli változását mutatja grafikusan



# 1. feladat – Tesztelés, szimuláció

- ***A szimuláció nagyon fontos***, de mi idő hiányában többször is el fogunk tekinteni ettől
- Váltunk át implementációs nézetről szimulációsra
- Új forrásfájl hozzáadása a projekthez
  - Típus: Verilog test fixture (tesztkörnyezet)
  - A forrásfájl nevének és elérési útjának megadása
  - A tesztelni kívánt modul kiválasztása

# 1. feladat – Tesztelés, szimuláció

- Nézzük meg a generált tesztkörnyezet vázat
- Egészítsük ki a tesztkörnyezetet a bemeneti adatokkal (tesztvektorok)
  - Megadunk néhány tesztvektort manuálisan
  - Használhatjuk a ***for*** utasítást a 256 lehetséges bemeneti kombináció előállításához
- Indítsuk el a szimulációt a tesztkörnyezetre
- Ellenőrizzük a szimulátorban a megfelelő működést
- Ha mindent rendben találunk, akkor kipróbálhatjuk a rendszert a hardveren is



# 1. feladat – Kipróbálás a hardveren

- Váltsunk vissza szimulációs nézetről implementációsra
- Vajon minden szükséges információ megtalálható a projektben a hardveren történő kipróbáláshoz?
  - Segítség: nézzük meg az első két diát
- Ha nem, akkor mi szükséges még?

# 1. feladat – Kipróbálás a hardveren

- Nincs még megadva, hogy az egyes port bitek mely FPGA lábra csatlakozzanak (külvilággal való kapcsolat)
- Új forrásfájl hozzáadása a projekthez
  - Típus: Implementation constraints file (UCF fájl)
  - A forrásfájl nevének és elérési útjának megadása
- Társítsunk minden port bithez egy FPGA lábat
  - A kapcsoló bitek az UCF fájlban: sw<0> ... sw<7>
  - A LED bitek az UCF fájlban: ld<0> ... ld<7>

***NET "port bit" LOC="FPGA láb";***

<b>LED</b>	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
<b>FPGA láb</b>	P43	P50	P51	P52	P53	P54	P58	P59

<b>Kapcsoló</b>	7	6	5	4	3	2	1	0
<b>FPGA láb</b>	P47	P48	P69	P78	P84	P89	P95	P101

# 1. feladat – Kipróbálás a hardveren

- A fejlesztői környezetben generáljuk az FPGA konfigurációs fájlt (BIT fájl)
- A Logsys GUI-val programozzuk fel az FPGA-t
  1. Az 5 V-os tápfeszültség bekapcsolása
  2. A JTAG funkció megnyitása
  3. Az eszközök felderítése a kártyán
  4. A BIT fájl letöltése az eszközre
- Próbáljuk ki a működést a hardveren is



# 1. feladat – Kipróbálás a hardveren

The screenshot shows the Logsys Control Panel software interface. The window has a menu bar (File, View, Window, Help) and a toolbar. The main area is divided into several sections:

- Info:** LOGSYS development cable.
- Control:** Includes checkboxes for RST and CLK, and a frequency input field set to 10 Hz with a 'Set' button.
- Power:** Includes a '+5V On' button (highlighted with a red arrow labeled '1'), a 'Current' section with a 'Maximum Value' of 450 mA, and a 'Log to file...' checkbox.
- Measurement:** Includes fields for '+5Vout: 4,94 V', 'I/Oref: 3,31 V', and 'JTAgref: 2,51 V', along with 'Maximum Value' and 'Samples/Second' settings.
- Configuration:** Includes a 'JTAG Download' checkbox (checked) and a 'Communication' section with checkboxes for BitBang I/O, UART, and USRT.
- Download (DC023):** A sub-window showing 'JTAG' settings. It includes a 'Query JTAG chain' button (highlighted with a red arrow labeled '3'), a 'Devices in the JTAG chain:' dropdown menu showing 'XC3S250E (Xilinx)', and a 'Configure the selected device...' button (highlighted with a red arrow labeled '4').
- Log:** A text area showing 'Found 1 device(s) in the JTAG chain.'
- Bottom Left:** A graphical representation of a current meter with a needle pointing to 0,45,00 on a scale from 0 to 500,00.

Red arrows labeled 1, 2, 3, and 4 point to the '+5V On' button, the 'JTAG Download' checkbox, the 'Query JTAG chain' button, and the 'Configure the selected device...' button, respectively.

## 2. feladat – Kettes komplementens képzés

- Módosítsuk úgy az előző feladat forrásfájlját úgy, hogy a LED-eken a kapcsolón beállított érték kettes komplementense jelenjen meg
- Milyen lehetőségek vannak a kettes komplementens képzésére?
- A 2. feladathoz szükséges új Verilog ismeretek
  - Bitenkénti negálás operátor
  - Összeadás és kivonás operátorok
  - Konstansok megadása (bitszám, számrendszer)
- Keressük meg azt a két értéket, melynek kettes komplementense önmaga!

# 3. feladat – Összeadó (opcionális)

- Készítsünk egy egyszerű összeadó kalkulátort
  - A bemeneti adatokat a kapcsoló alsó és felső 4 bitjén adjuk meg
  - Az eredményt 5 biten (4 bites összeg és a kimenő átvitel bit) jelenítsük meg a LED-eken
  - A felső 3 LED legyen kikapcsolva
- A 3. feladathoz szükséges új Verilog ismeretek
  - Belső jelek megadása (típus, méret)
  - Indexelő operátor (bit, bittartomány kiválasztása)
- Próbáljuk ki az összeadót az FPGA kártyán előjel nélküli és kettes komplementes kódolású adatokkal is