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

Digitális technika (VIMIAA02)
Laboratórium 1

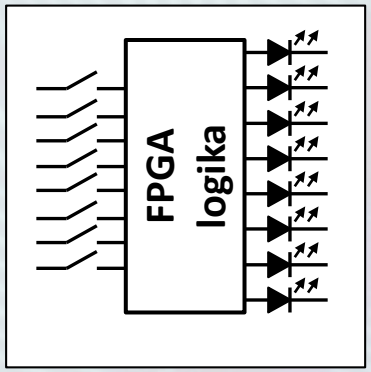
Fehér Béla
Raikovich Tamás, Fejér Attila

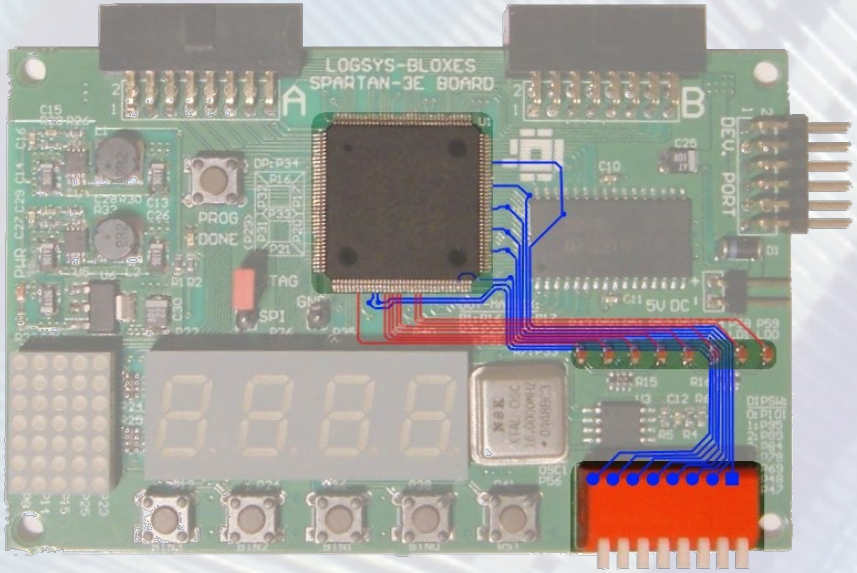
BME MIT

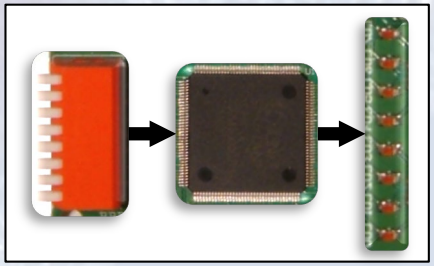
Lab1_1 feladat: HW „Hello World!”

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

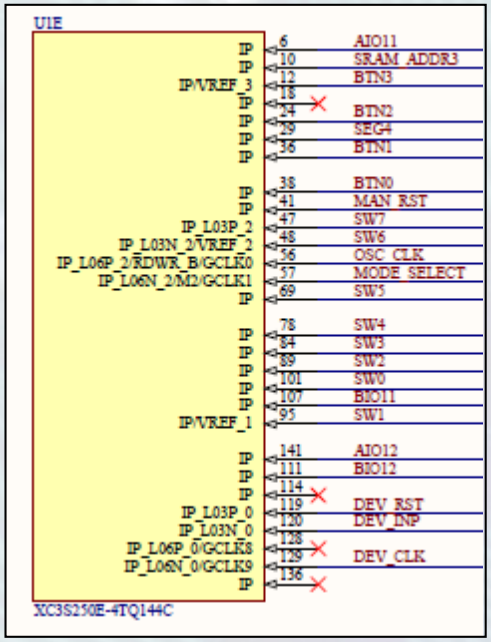


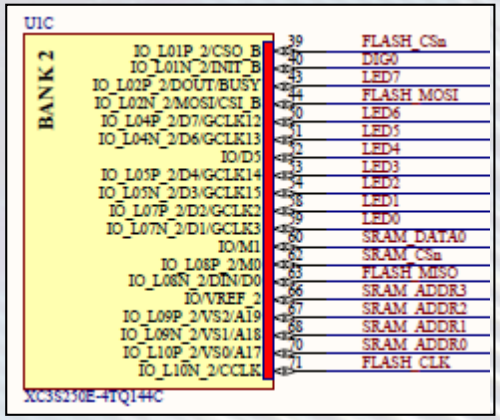


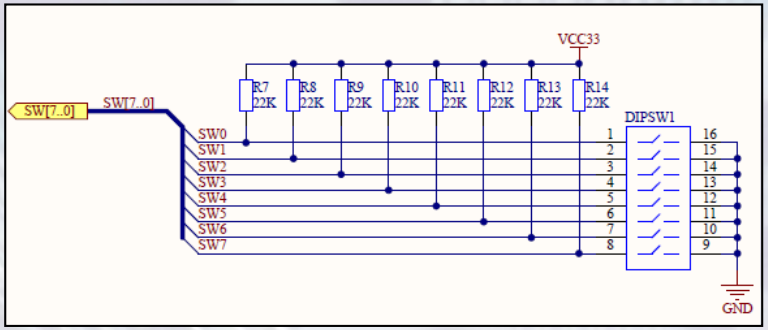


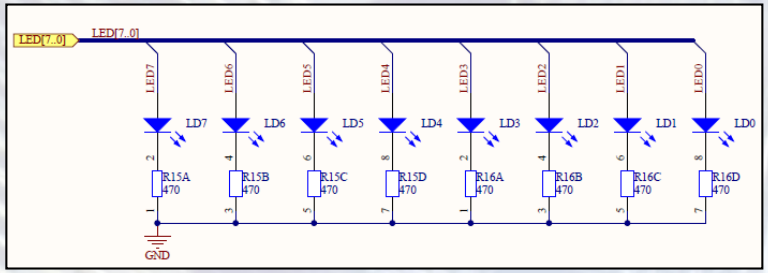
Lab1_1 feladat: HW „Hello World!”

- **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

Xilinx ISE GUI - Lab1_1 feladat

A Xilinx ISE GUI használata diasorozat bemutatása ITT

Ennek keretében a Lab1_1 feladat elkészül

- ISE elindítása, projekt létrehozása
- Lab1_1.v forrásfájl mintakeret specifikálása
- LOGSYS_SP3E.UCF fájl hozzáadása és adaptálása a kívánt interfészekhez (szükséges lábak kiválasztása)
- A Lab1_1a feladat specifikálása a funkcionális (az adott műveletet specifikáló) kódrészlettel
- A funkcionális kód ellenőrzése szimulációval
- Konfigurációs fájl generálása, letöltés, működés tesztelése a kártyán

Lab1_1_1 feladat - KÉSZ

Egyszerű jelátvezetés a kapcsolókról a LED-ekre

- Kezelhető a 8 bit együtt, vektorosan vagy egyedi bitenként, de akkor 8 sorban kell megadni a kijelölést
- LD[i] = SW[i] esetén a vektoros egyszerűbb

```
module Lab1_1(
    input   [7:0] sw,
    output  [7:0] ld
    );


//////////////////////////////////////////////////////////////////////////////////
// 1_1_1 A kimeneti LED-ek vezérlése közvetlenül a kapcsolókkal
// LD[i] = SW[i] , ez az áramkörben egy direkt vezérlés, a Verilog HDL nyelvben
// folytonos értékadásnak nevezzük
//////////////////////////////////////////////////////////////////////////////////

//?????????????

endmodule
```

Lab1_1_1 feladat

Egyszerű jelátvezetés a kapcsolókról a LED-ekre

- Ez egy folytonos vezérlés SW → LED bitjei között
- Egy egyszerűsített modellje a HW működésének

```
module Lab1_1(
    input   [7:0] sw,
    output  [7:0] ld
    );

//////////////////////////////////////////////////////////////////////////////////
// 1_1_1 A kimeneti LED-ek vezérlése közvetlenül a kapcsolókkal
// LD[i] = SW[i] , ez az áramkörben egy direkt vezérlés, a Verilog HDL nyelvben
// folytonos értékadásnak nevezzük
//////////////////////////////////////////////////////////////////////////////////


assign ld = sw;                 // Vektoros folytonos értékadás


endmodule
```

Lab1_1_2 feladat

Kettes komplemens képzés SW → LED között

- A Lab1_1_1 feladat egyetlen aktív kódsorát hatástalanítva (//komment) a Lab1_1 modulban elkészítjük a Lab1_1_2 feladatot is
- Aritmetikai és bitművelet az SW vektorváltozón

```
//assign ld = sw;                    // Vektoros folytonos értékadás

////////////////////////////////////////////////////////////////////////////////
// 1_1_2 Kettes komplemens képzés
// A LED-eken jelenítsük meg a 8 bites kapcsolón beállított bitkombináció
// kettes komplemensét
////////////////////////////////////////////////////////////////////////////////

//?????????????
```

Lab1_1_2 feladat

Kettes komplemens képzés SW → LED között

- Az SW kapcsoló összes bitjének egyedi invertálása
- Az „1” hozzáadása 8 bites értékként (0000_0001)
- Aritmetikai és bitművelet az SW vektorváltozón

```
//assign ld = sw;               // Vektoros folytonos értékadás

//////////////////////////////////////////////////////////////////////////////////
// 1_1_2 Kettes komplemens képzés
// A LED-eken jelenítsük meg a 8 bites kapcsolón beállított bitkombináció
// kettes komplemensét
//////////////////////////////////////////////////////////////////////////////////

assign ld = ~sw + 8'b1;      // Bitenkénti invertálás és 1 hozzáadása

endmodule
```

Lab1_1_2 feladat

- **Ellenőrzés szimulációval egy-két értékre**
  - Hogyan érdemes beállítani a Waveform kijelzését?
  - Bináris, decimális?
    - Hullámforma ablakban jelnév kiválasztása
    - Jobb egérgomb -> Radix -> Signed Decimal
- **Generáljuk a specifikációhoz tartozó konfigurációs adatfájlt**
- **Letöltés után ellenőrizzük a működést**

Lab1_1_3 feladat

Aritmetikai műveletek vizsgálata

- Továbbra is a Lab1_1 modult szerkesztjük
- Definiáljunk 2 db 4 bites belső változót (a[3:0] és b[3:0]) előjel nélküli vagy előjeles formátummal
- A 8 bites SW jelből résztartomány kijelöléssel származtatjuk a két 4 bites forrásoperandust

```
//////////////////////////////////////////////////////////////////////////////////
// 1_1_3 Aritmetikai műveletek vizsgálata (+, -, *) 4 bites operandusokon
// Az operandusok lehetnek előjel nélküli vagy előjeles értékek
// A 8 bites SW kapcsoló bemenetet 2 db 4 bites változóra osztjuk
// Az alsó digit az "a", a felső digit a "b".
// Ezek között végezzük el az aritmetikai műveleteket,
// az eredményt a LED-eken jelenítjük meg, 8 bites formátumban.
//////////////////////////////////////////////////////////////////////////////////
```

Lab1_1_3 feladat

Aritmetikai műveletek vizsgálata

- Egyszerre egy típus és egy művelet lehet aktív

```
//assign ld = ~sw + 8'b1;     // Bitenkénti invertálás és 1 hozzáadása

////////////////////////////////////////////////////////////////////////////////
// 1_1_3 Aritmetikai műveletek vizsgálata (+, -, *) 4 bites operandusokon
// Az operandusok lehetnek előjel nélküli vagy előjeles értékek
// A 8 bites SW kapcsoló bemenetet 2 db 4 bites változóra osztjuk
// Az alsó digit az "a", a felső digit a "b".
// Ezek között végezzük el az aritmetikai műveleteket,
// az eredményt a LED-eken jelenítjük meg, 8 bites formátumban.
////////////////////////////////////////////////////////////////////////////////

//wire  [3:0] a;              // Előjel nélküli változók deklarálása
//wire  [3:0] b;              // Értéktartományuk 0 ... +15

//wire signed [3:0] a;        // Előjeles változók deklarálása
//wire signed [3:0] b;        // Értéktartomány -8 ... +7

//assign a = sw[3:0];         // Kapcsoló bitek hozzárendelése
//assign b = sw[7:4];         // a négybites változókhoz

//assign ld = a + b;          // 1. művelet ÖSSZEADÁS
//assign ld = a - b;          // 2. művelet KIVONÁS
//assign ld = a * b;          // 3. művelet SZORZÁS
```

Lab1_1_3 feladat

A feladat jellemzői

- A bemeneti értéktartományok előjeles és előjel nélküli esetben természetesen eltérőek
- Az eredmény minden esetben 8 biten áll elő és általában megőrzi a forrásoperandusok típusát
  - Mikor nem ábrázolható az eredeti formátumban?
  - Hogyan állapítható meg a túlcsordulás?
- A kivonásnál az eredmény kettes komplemens lesz
  - Előjel nélküli pozitív operandusok esetén is

Lab1_1_3 feladat

- Ellenőrzés szimulációval egy választott bemeneti típusra és egy kiválasztott műveletre, néhány értékkel
- Generáljuk a specifikációhoz tartozó konfigurációs adatfájlt
- Letöltés után ellenőrizzük a működést