

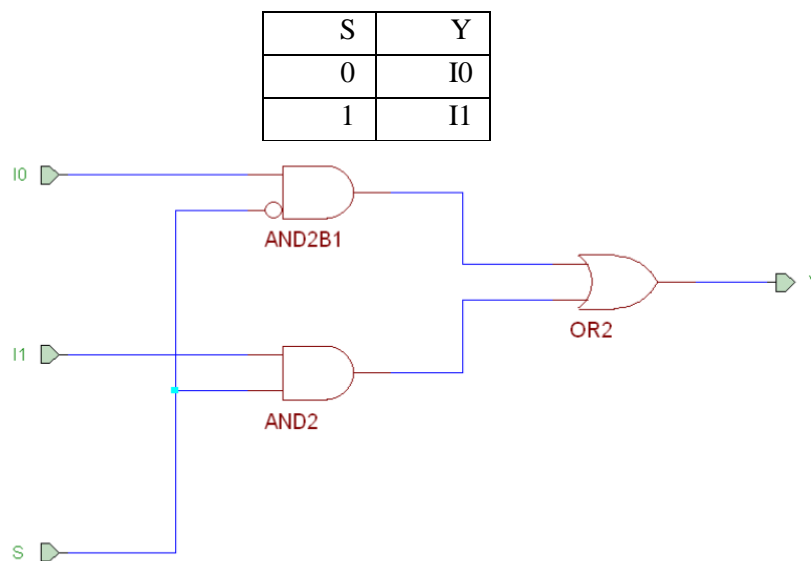


## Laborgyakorlat

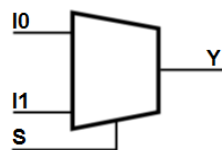
### Logikai áramkörök számítógéppel segített tervezése (CAD)

#### Multiplexer (MPX)

A multiplexer egy olyan áramkör, amely több bemeneti adat közül a megcímzett bemeneti adatot továbbítja a kimenetére. Ez az egyik leggyakrabban használt kombinációs hálózat. Legegyszerűbb esetben a cím az egy bites „select” jel (S), ami két bemenet közül (I0, I1) választ ki egyet (1. ábra, 2. ábra). A két bemenetű multiplexer igazságtáblája a következő.



1. ábra - Két bemenetű multiplexer kapcsolási rajza



2. ábra - Két bemenetű multiplexer logikai rajza

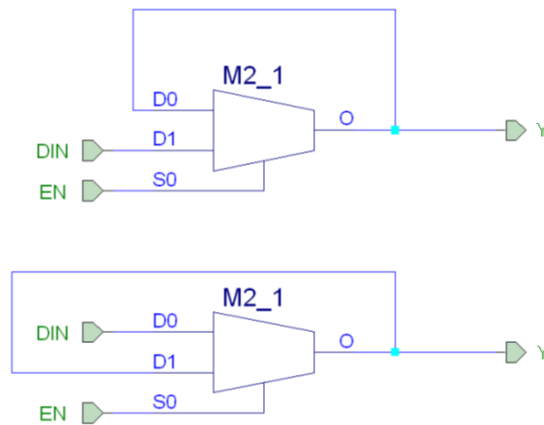
A multiplexált adat nem csak 1-bites lehet, azaz a címmel kiválasztott és a kimenetre továbbított adat több bites is lehet.

A multiplexert legtöbbször bonyolult adat utak kialakítására használják pl. a processzorok ALU-jában. Vannak egyéb alkalmazási, ezek közül először kettőt mutatunk meg.



## Latch

A latch egy olyan tároló elem, amely nem órajelre (élre) működik, hanem egy engedélyező jel megléte esetén az áramkör transzparens, azaz átengedi a bemenetén lévő jelet, az engedélyező jel megszűnésére az utolsó értéket tárolja. Az alábbi két rajzon (3. ábra) az engedélyező jel aktív értéke magas, ill. alacsony.

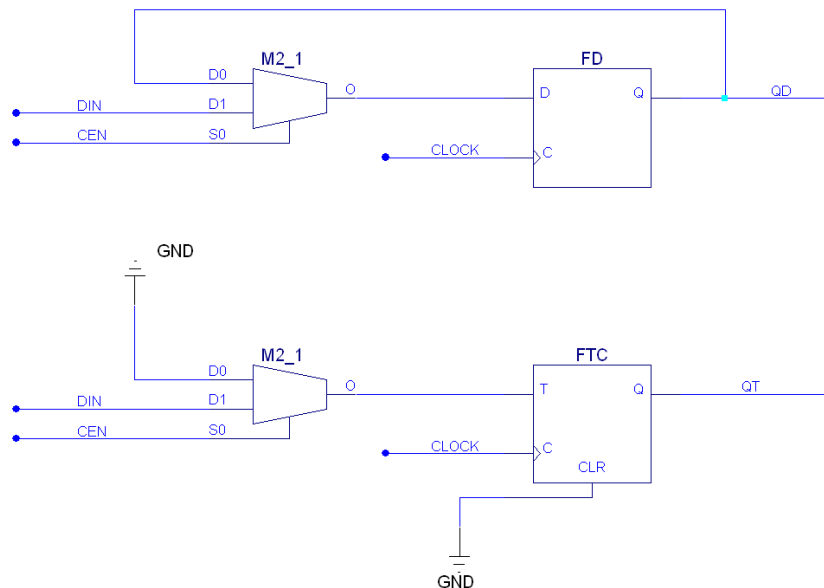


3. ábra – Latch



### Korrekt flip-flop órajel engedélyezés

Az órajelre működő szinkron hálózatok esetén fontos, hogy az összes tároló egyszerre billenjen, ellenkező esetben rövid időre hazard állapotok léphetnek fel. Ezen kívül a dinamikus bemeneten a jelnek stabilnak kell lennie bizonyos idővel az órajel felfutása előtt és után. Ez az áramkörre jellemző katalógus érték (setup time, ill. hold time). Ha ezek a feltételek nem teljesülnek a flip-flop működése bizonytalanná válik, ha például az órajelet kapuzzuk, így plusz késleltetést viszünk bele. Ezért a korrekt megoldást egy multiplexeren áthaladó enable jel biztosítja (4. ábra). Az első esetben egy D flip-flop, a másodikban egy T flip-flop esetében. Az első esetben, ha az órajel nincs engedélyezve, az órajelre saját magát írja be újra meg újra, ugyanaz a hatás, mintha nem lenne órajel. A második esetben a bemenetre 0 kerül, így a T flip-flop nem billen.

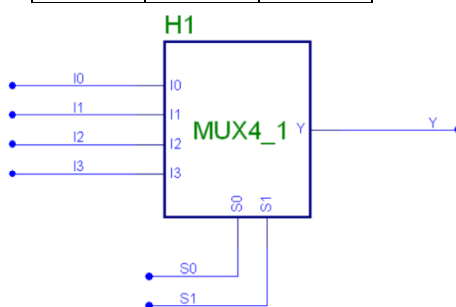


4. ábra - Korrekt flip-flop órajel engedélyezés

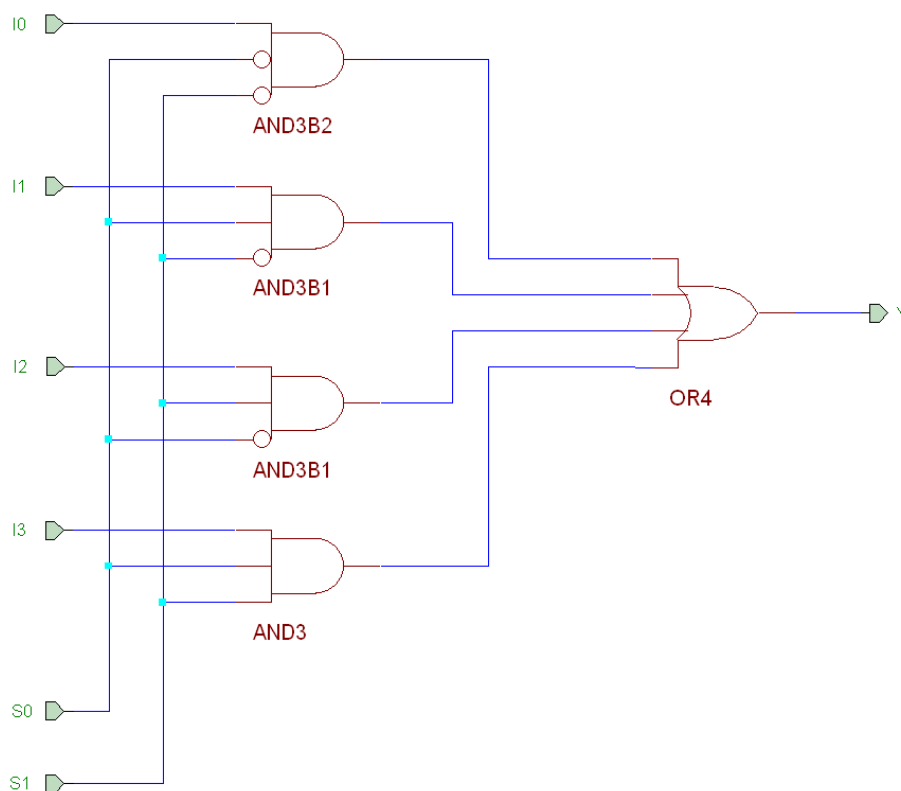


A multiplexer címbitjeinek eggyel való megnövelése a bemenetek számának a kétszeresét teszi lehetővé. Így két címbit esetén kapjuk a 4-bemenetű multiplexert (5. ábra, 6. ábra). A négy bemenetű multiplexer igazságtáblája a következő.

S	S	Y
1	0	
0	0	I0
0	1	I1
1	0	I2
1	1	I3



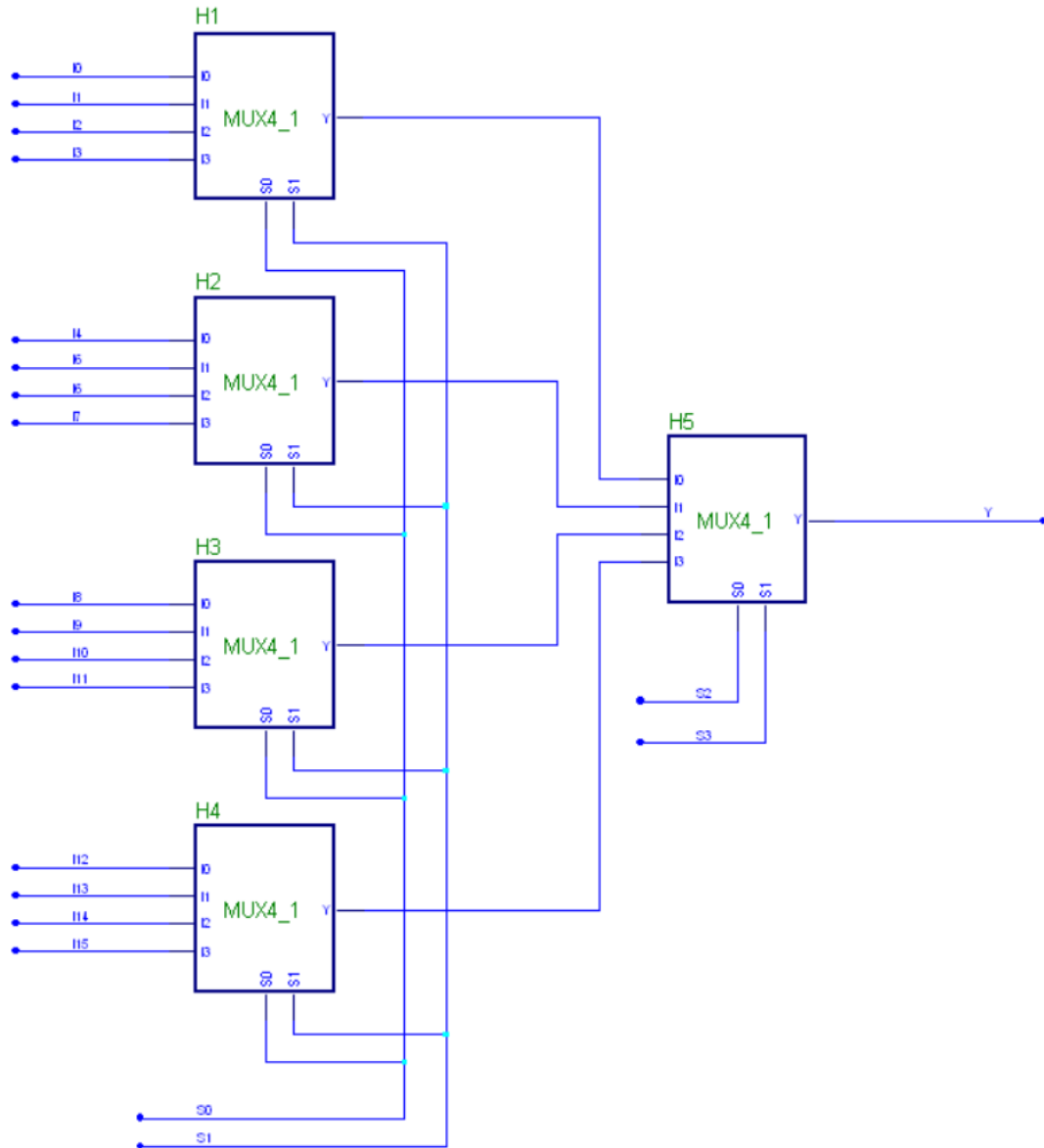
5. ábra - Négy bemenetű multiplexer logikai rajza



6. ábra - Négy bemenetű multiplexer kapcsolási rajza



A nagyobb számú bemenetekkel rendelkező multiplexer tervezése kétféleképpen lehetséges: a megfelelő számú kapuból megtervezve, vagy kisebb multiplexerek szisztematikus összekapcsolásával (kaszkádosítás). Például egy 16 bemenetű multiplexert így is megtervezhetünk (7. ábra). Ez ugyan redundáns a felhasznált kapuk számát illetően, de az alkatrész könyvtárban (szimbólum könyvtár) meglévő alkatrészekből összerakható.

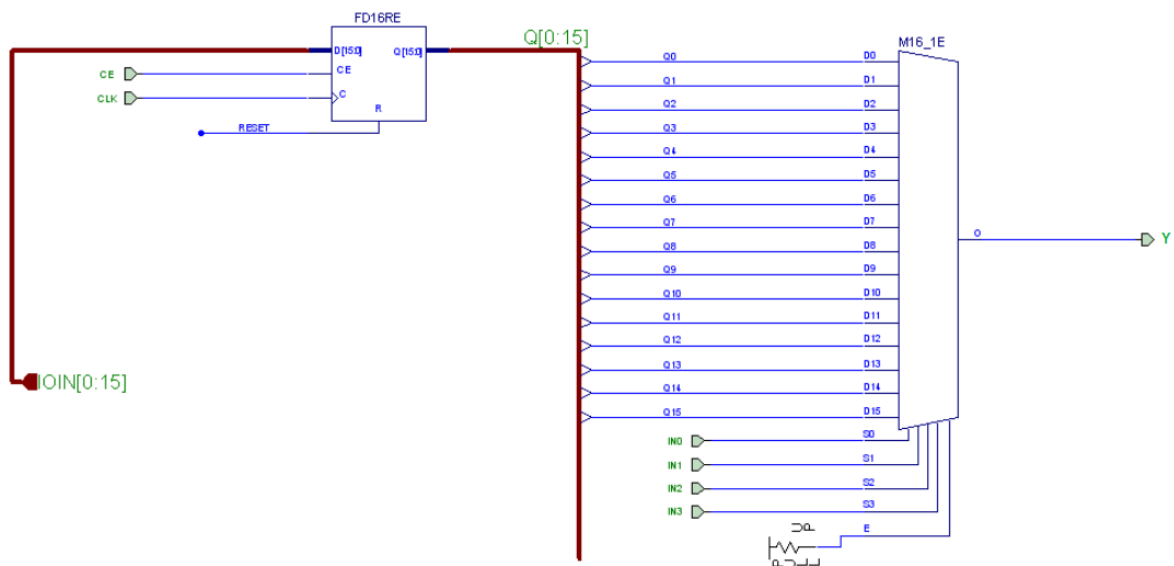


7. ábra - 16 bemenetű multiplexer tervezése kisebb multiplexerek felhasználásával

A multiplexer lehetőséget nyújt look-up table megvalósítására (lásd a bevezető gyakorlat leírását). Az általunk használt FPGA CLB-jében 4-bemenetű LUT található tetszőleges 4-változós logikai függvény előállítására. Ha ezt egy tárolóval oldanák meg, minden egyes CLB-ben egy 64K bites tárolóra lenne szükség az összes logikai függvény előállítására. Ehelyett pl. egy 16 bites regiszter és egy 16 bemenetű multiplexer elegendő (8. ábra). A multiplexer címbitjeire csatlakozik az adott logikai függvény 4 változója. A rajz alapján a



szoftver kiszámolja, hogy melyik logikai függvény kell a megvalósításhoz, és a 4 bemenet összes kombinációjának megfelelő 16 kimeneti bitet betölti egy 16 bites regiszterbe, amelynek kimenetei csatlakoznak a multiplexer megfelelő címzésű bemenetéhez. Látható, hogy egy ilyen megoldás erőforrás igénye lényegesen kisebb.

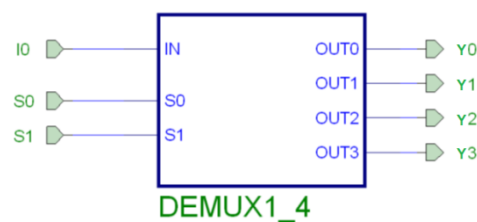


8. ábra - Look-up table megvalósítása multiplexer segítségével

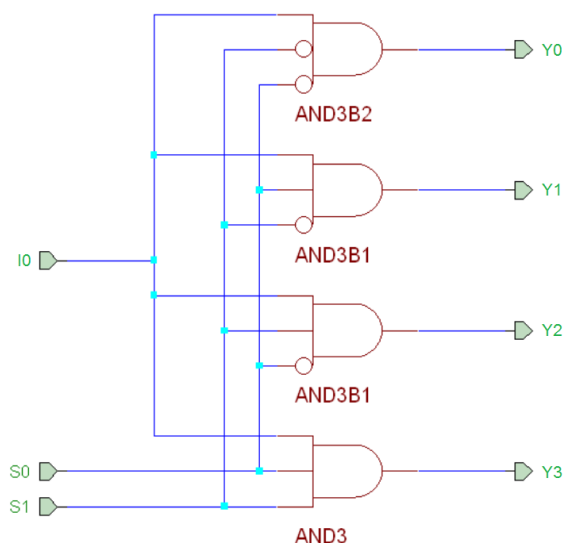
### Demultiplexer (DMPX)

A demultiplexer működése a multiplexerének a fordítottja. Egy adat bemenete (I) van, és a címző bemenetek (S) határozzák meg, hogy melyik kimenetre (OUT) továbbítja a bemeneti adatot (9. ábra, 10. ábra). A többi kimenet állapota inaktív. A felhasználási területe: adat szétosztás több feldolgozó egység felé.

Az adat bemenet (I), ill. a kimenetek (OUT) nem csak 1-bitesek lehet.



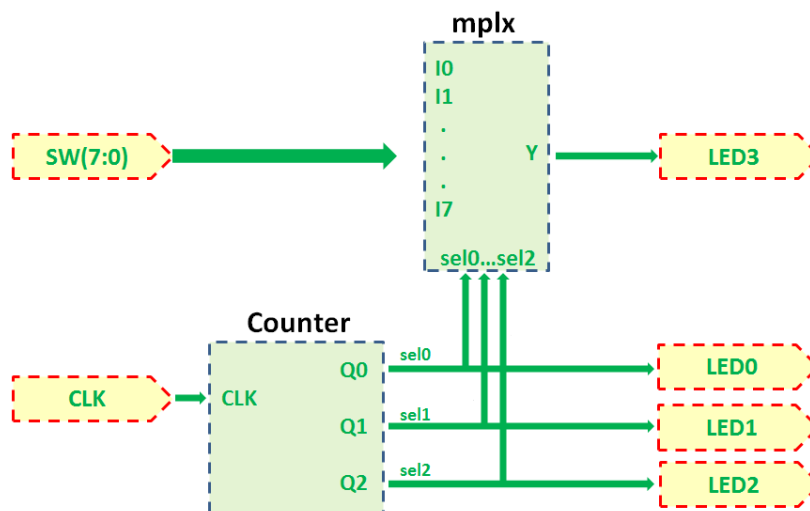
9. ábra - Négy kimenetű demultiplexer



10. ábra - Négy kimenetű demultiplexer kapcsolási rajza

### Laborfeladat

Tervezzen egy 8-bemenetű multiplexeres kapcsolást (11. ábra). **A projektben adja a top modul (főmodul) kapcsolási rajznak az mplxtop.sch nevet!** A multiplexert külön makróban (rajzolt modulban) készítse el (AND és OR kapuk felhasználásával), amihez először létre kell hozni egy új kapcsolási rajzot (mplx.sch). A multiplexer adatbemeneteit a Digilent Basys2 kártyán lévő SW0-SW7 kapcsolókon (SW(7:0)) lehessen beállítani, a címző bemeneteket a multiplexer bemenetére csatlakozó 3-bites számláló kimenete adja. A 3-bites számláléhoz használja a szimbólumok (alkatrészek) között megtalálható CB4CE 4 bites számlálót (a 4. kimeneti bitjét, azaz a Q3-at nem használjuk majd fel), aminek az órajele (CLK) a BTN0 nyomógomb legyen. A helyi értékek megfelelnek a kapcsolók és a számláló kimeneteinek az indexének. Végezze el az áramkör szimulációját a kapott szimulációs fájl segítségével.



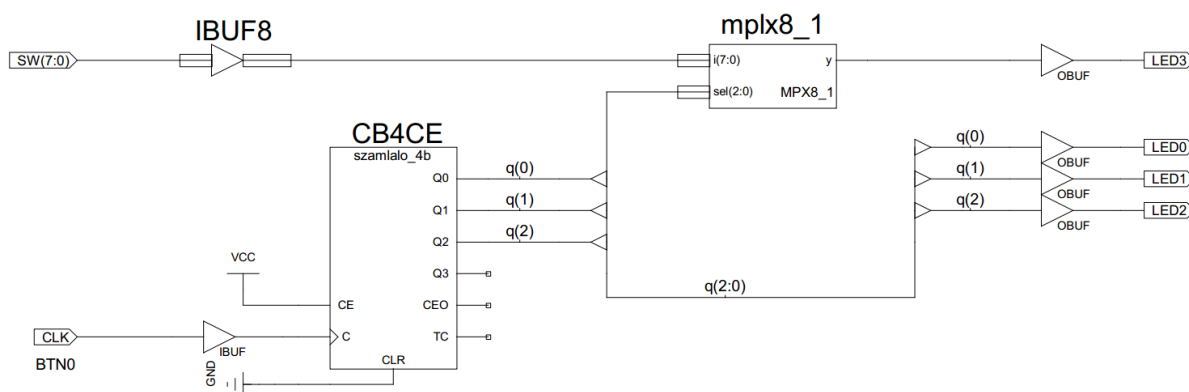


11. ábra – Multiplexeres kapcsolás blokkvázlata

Amennyiben az áramkör működése megfelelő, implementálja azt, a kapott bit kiterjesztésű fájl töltse le a Basys2 kártyába, és a kapcsolók és az órajel nyomógomb segítségével próbálja ki az áramkör működését.

A feladathoz szükséges port nevek (ucf file)			
Port név	Busz	CP132 tokozás	Leírás
CLK	-	NET „clk” LOC = „G12”;	A számláló órajel bemenete: BTN0 nyomógomb
SW(7:0)	8 bit	NET „SW<7>” LOC = „N3”; NET „SW<6>” LOC = „E2”; NET „SW<5>” LOC = „F3”; NET „SW<4>” LOC = „G3”; NET „SW<3>” LOC = „B4”; NET „SW<2>” LOC = „K3”; NET „SW<1>” LOC = „L3”; NET „SW<0>” LOC = „P11”;	Multiplexer adat bemenetei
LED0, LED1, LED2, LED3	-	NET „LED3” LOC = „P6”; NET „LED2” LOC = „P7”; NET „LED1” LOC = „M11”; NET „LED0” LOC = „M5”;	Számláló állapotát megjelenítő LED-ek: LED0, LED1, LED2  Multiplexer adat kimenete: LED3

Lab\_03: 8-1 multiplexer 4-bites számlalóval



Vígh Tamás, ÓBUDAI EGYETEM, 2015





## **Alkalmazandó műszerek és eszközök**

- PC számítógép
- Digilent Basys2 Spartan-3E FPGA mérőpanel
- Digilent Adept konfiguráló szoftver

## **Hivatkozások, felkészüléshez ajánlott irodalom**

- [1] FPGA fejlesztés a Xilinx ISE Webpack-ben, Elektronikus formában a tantárgy honlapján
- [2] Digilent Basys2 Board Reference Manual, Elektronikus formában a tantárgy honlapján
- [3] Spartan-3E Libraries Guide for Schematic Designs, Elektronikus formában a tantárgy honlapján
- [4] Kóré László: Digitális elektronika I. BMF 1121
- [5] Arató Péter: Logikai rendszerek tervezése, Tankönyvkiadó