



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

Digitális technika (VIMIAA02)

6. gyakorlat és laboratórium

Raikovich Tamás
BME MIT

Shiftregiszterek, számlálók

- **1. feladat: shiftregiszter megvalósítása**
 - a) 8 bites törölhető, engedélyezhető, kétirányú SHR
 - b) Knight Rider futófény
- **2. feladat: számlálók megvalósítása**
 - a) 4 bites törölhető, tölthető, engedélyezhető felfele számláló végállapot jelzéssel
 - b) Másodperc számláló (2 digités BCD, 0 – 59 s)
- **A feladatokhoz szükséges új Verilog ismeretek:**
 - Shiftelés operátorok
 - Konkaténálás operátor

Verilog HDL ismeretek

Shiftelés (léptetés) operátorok

Logikai shift operátorok: << (balra), >> (jobbra) léptetés

- Operandusok száma: 2
- ***A belépő bit mindig 0*** (ha egy jel értékét kell beléptetni, akkor ehhez a shiftelés operátorok nem használhatók!)
- Példák:
 - `8'b0011_1100 >> 2 = 8'b0000_1111`
 - `8'b0011_1100 << 2 = 8'b1111_0000`

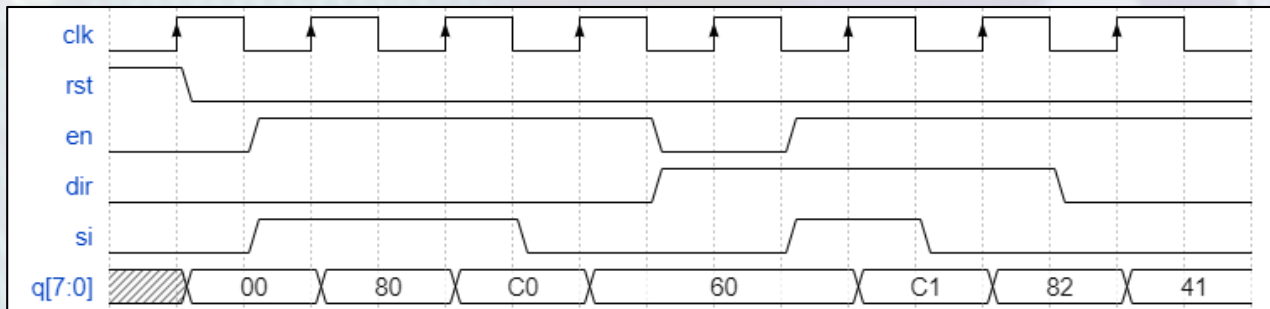
Verilog HDL ismeretek

Konkatenálás operátor

- **Konkatenálás (összefűzés) operátor: { }**
 - Több, vesszőkkel elválasztott operandus összefűzése: {op1, op2, op3, ...}
`{5'b10110, 2'b10, 1'b0, 1'b1} = 9'b1_0110_1001`
 - Ugyanazon operandus N-szeres összefűzése: {N{operandus}}
`{4{3'b101}} = 12'b101_101_101_101`
- **Fontos felhasználási esetek**
 - Előjel kiterjesztés: az előjel bitet a felső bitekbe kell másolni
`wire [3:0] s_4bit; //4 bites előjeles`
`wire [7:0] s_8bit = {{4{s_4bit[3]}}, s_4bit};`
 - Vektor maszkolása egyetlen bittel: a többszörözés hiányában az 1 bites kisebb operandus nullákkal lenne kiterjesztve a nagyobb operandus méretére
`wire [3:0] data; //4 bites adat`
`wire enable; //Engedélyezés (maszk)`
`wire [3:0] mdata1 = data & enable; //Rossz!!!`
`wire [3:0] mdata2 = data & {4{enable}}; //Helyes`
 - Tetszőleges érték (jel) beléptetése
`wire [7:0] data; //8 bites adat`
`wire si; //Beléptetendő érték`
`wire [7:0] shift_l = {data[6:0], si}; //Léptetés balra`
`wire [7:0] shift_r = {si, data[7:1]}; //Léptetés jobbra`

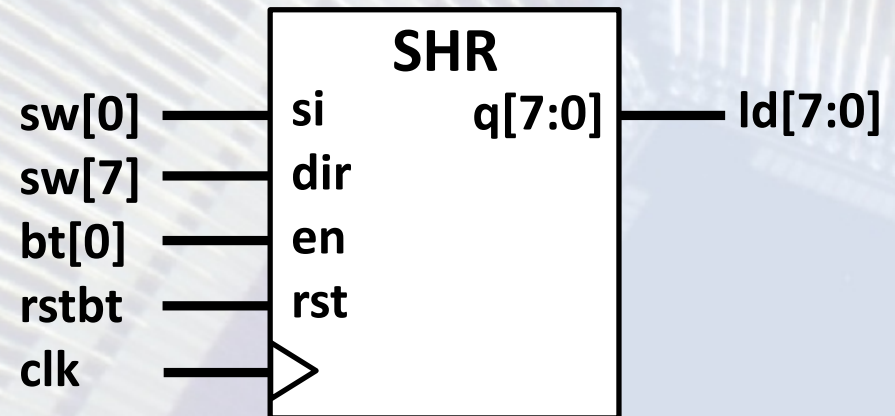
1.a feladat: 8 bites shiftregiszter

- 8 bites törölhető, engedélyezhető, kétirányú shiftregiszter egy soros bemenettel



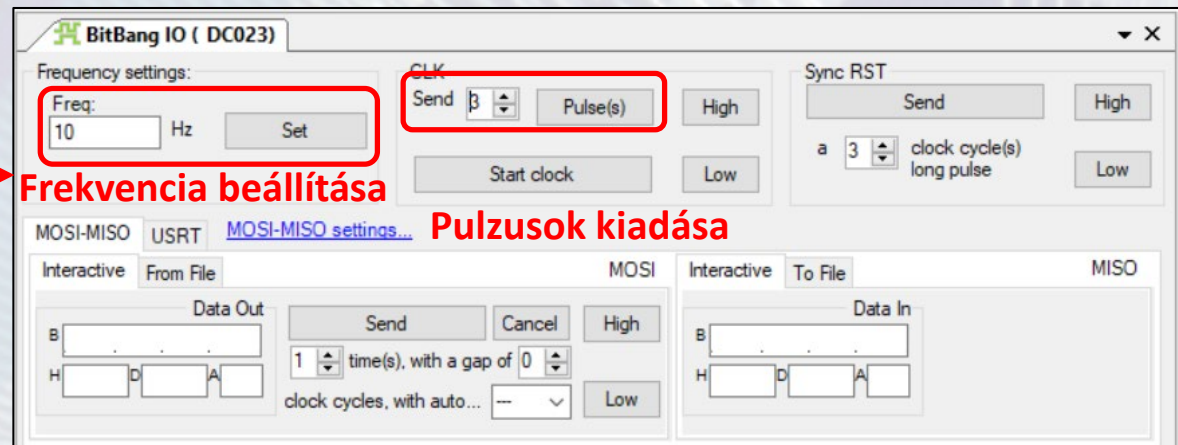
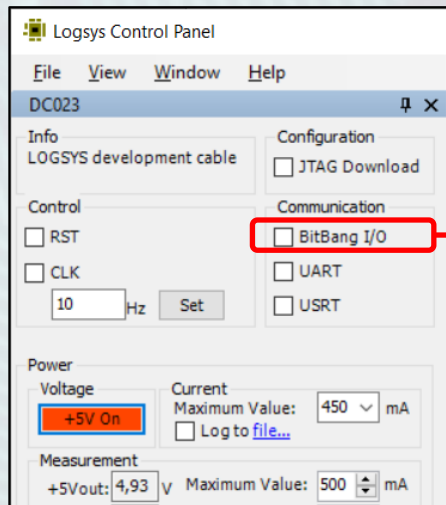
- A bemenetek és kimenetek bekötése

- clk: a letöltőkábeltől (*bitbang I/O*)
- rst: rstbt
- en: bt[0]
- dir: sw[7] (0: \rightarrow , 1: \leftarrow)
- si: sw[0]
- q: ld[7:0]



1.a feladat: 8 bites shiftregiszter

- Ellenőrizzük szimulátorban a működést (opcionális)
 - Vegyük alapul az előző idődiagramot (1 osztás: 50 ns)
- Próbáljuk ki az FPGA kártyán az SHR működését
 - Az órajelet a **Bitbang I/O** funkcióval állítsuk elő



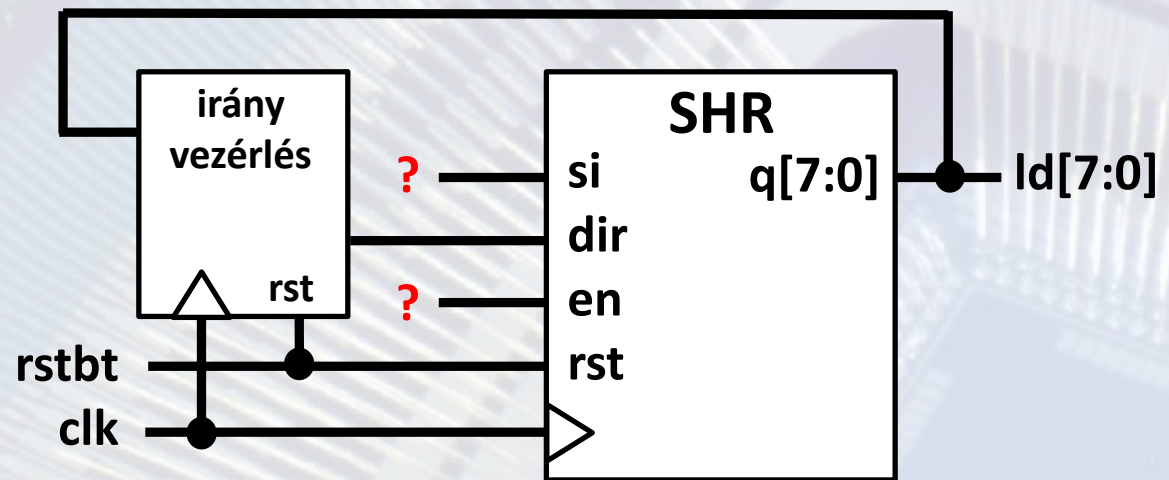
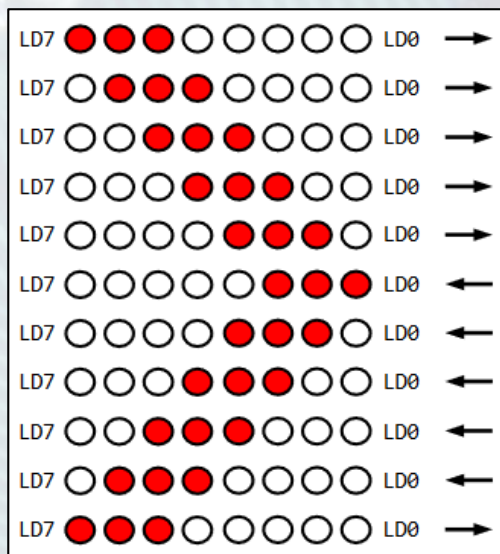
Frekvencia beállítása

Pulzusok kiadása

1.b feladat: Knight Rider futófény

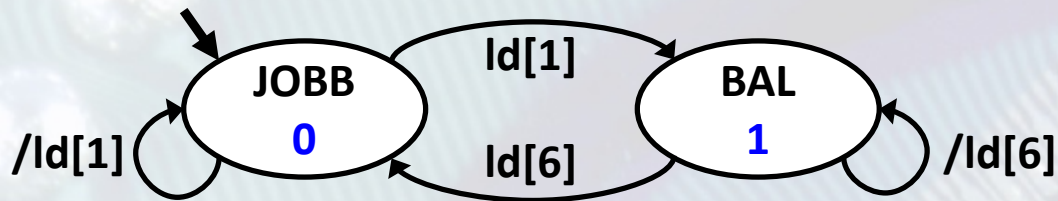
Egészítsük ki a shiftregisztert Knight Rider futófénnyé

- Moore jellegű irány vezérlést valósítsunk meg
- Melyik állapotokban kell irányt váltani?
- Mi legyen a shiftregiszter kezdőállapota *rst* hatására?
- Mi legyen a nem használt bemenetek értéke?



1.b feladat: Knight Rider futófény

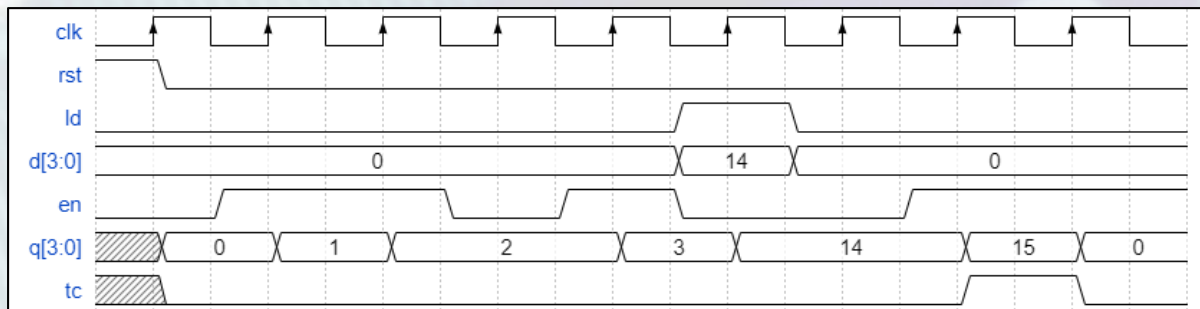
- Az irány vezérlés állapotgráfja



- Új top-level modul kijelölése: lab05_1b
 - Jobb klikk a modulon → Set as Top Module
- Ha az UCF fájl esetleg nem kerülne át az új top-level modul alá, akkor távolítsuk el a projektből az UCF fájlt és adjuk hozzá újból
- Próbáljuk ki az FPGA kártyán a futófény működését

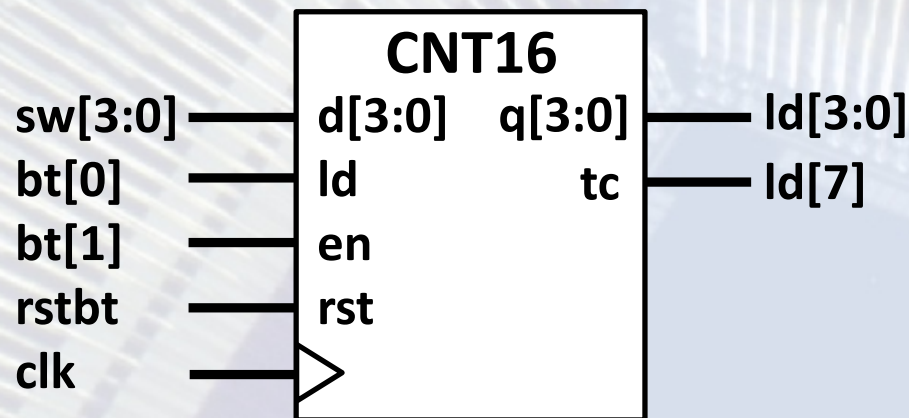
2.a feladat: 4 bites számláló

- 4 bites törölhető, tölthető, engedélyezhető felfele számláló végállapot jelzéssel



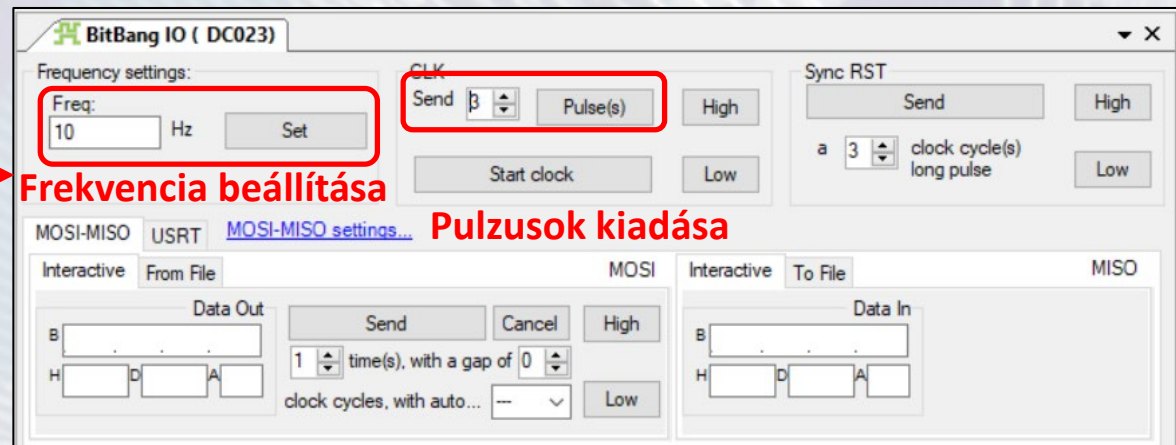
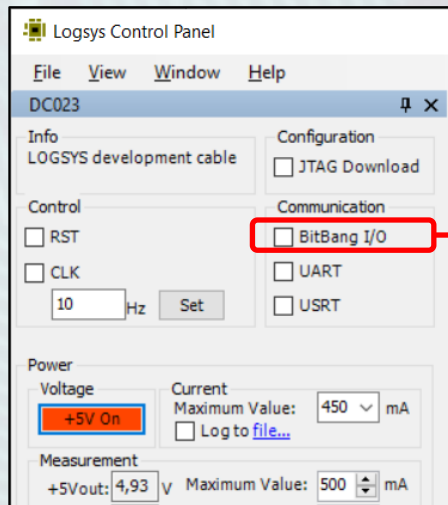
- A bemenetek és kimenetek bekötése

- clk: a letöltőkábeltől (*bitbang I/O*)
- rst: rstbt
- ld: bt[0]
- d: sw[3:0]
- en: bt[1]
- q: ld[3:0]
- tc: ld[7]



2.a feladat: 4 bites számláló

- Ellenőrizzük szimulátorban a működést (opcionális)
 - Vegyük alapul az előző idődiagramot (1 osztás: 50 ns)
- Próbáljuk ki az FPGA kártyán a számláló működését
 - Az órajelet a **Bitbang I/O** funkcióval állítsuk elő



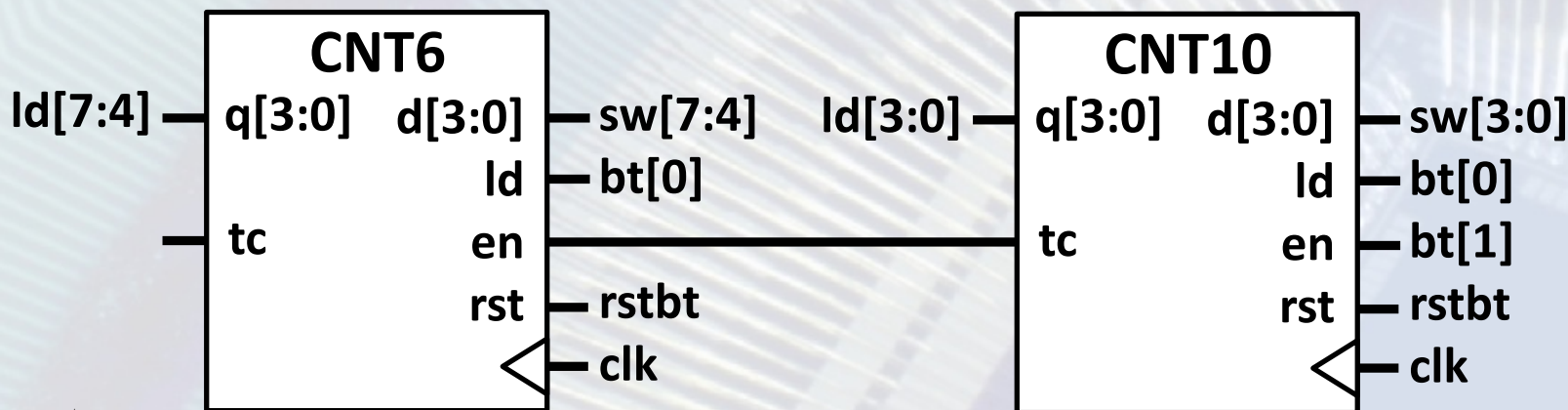
2.b feladat: másodperc számláló

Készítsünk másodperc számlálót, amely 2 digités BCD formátumban megjeleníti a 0–59 értékeket a LED-eken

- Egyes helyiérték \rightarrow $ld[3:0]$
- Tízes helyiérték \rightarrow $ld[7:4]$
- Hogy valósítsuk meg ezt a lehető legegyszerűbben?
 - 60 állapotú számláló és bináris \rightarrow BCD átalakítás?
 - Esetleg más ötlet az eddig tanultak alapján?

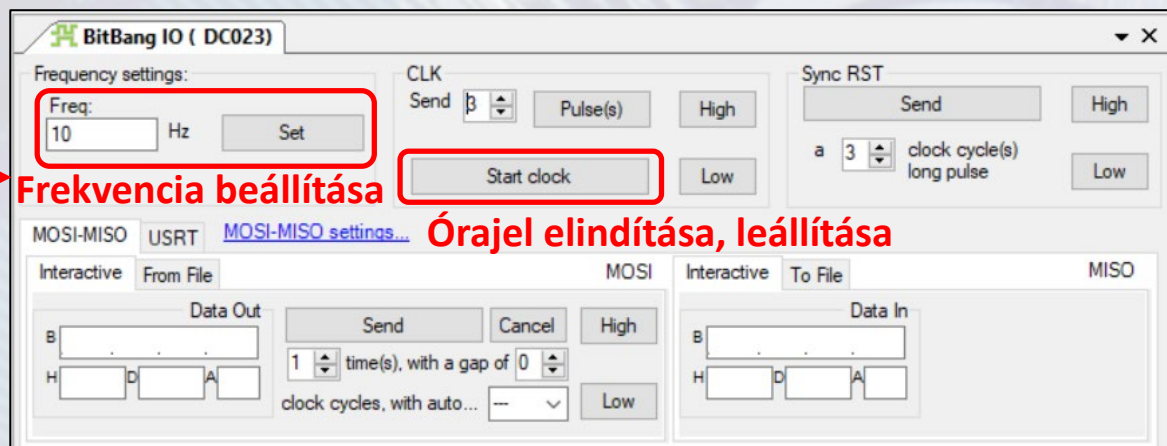
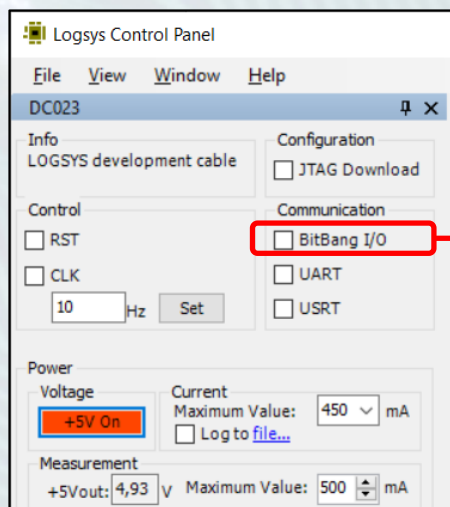
2.b feladat: másodperc számláló

- Elkerülhető a bináris→BCD átalakítás, ha mindegyik helyiértékhez külön-külön számlálót használunk
 - Egyes helyiérték → 10 állapotú számláló (0-9)
 - Tízes helyiérték → 6 állapotú számláló (0-5)
 - Minimális módosítás kell az előző számlálóban
- Az adott helyiérték végállapot jelzése engedélyezi a következő helyiérték számlálóját



2.b feladat: másodperc számláló

- Új top-level modul kijelölése: lab05_2b
- Próbáljuk ki az FPGA kártyán a számláló működését
- Az órajelet a *Bitbang I/O* funkcióval állítsuk elő



Frekvencia beállítása

Órajelel elindítása, leállítása