



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

# **Digitális technika (VIMIAA02)**

## **2. laboratórium**

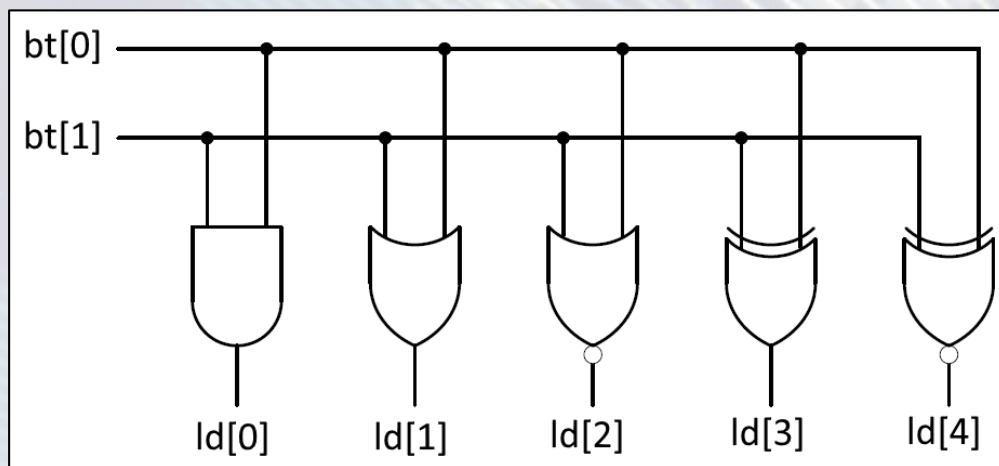
**Raikovich Tamás**  
**BME MIT**

# Kombinációs hálózatok

- **1. feladat: 2 változós elemi logikai függvények**
  - AND, OR, NOR, XOR, XNOR
- **2. feladat: 4 változós elemi logikai függvények**
  - AND, OR, NOR, XOR, XNOR
- **3. feladat: hierarchikus tervezés bemutatása**
  - 1 bites teljes összeadó realizálása → FADD modul
  - Ebből 4 bites összeadó kialakítása kaszkádosítással, azaz 1 bites egységek láncba kapcsolásával

# 1. feladat: elemi logikai függvények

- **Az 1. feladathoz szükséges új Verilog ismeretek**
  - Bitenkénti operátorok (ezeket használjuk)
  - Bitredukciós operátorok (inkább a 2. feladathoz)
- **Nyissuk meg a kiindulási projektet**
- **A megvalósítandó rendszer kapcsolási rajza (a nem használt LED-ek legyenek kikapcsolva)**

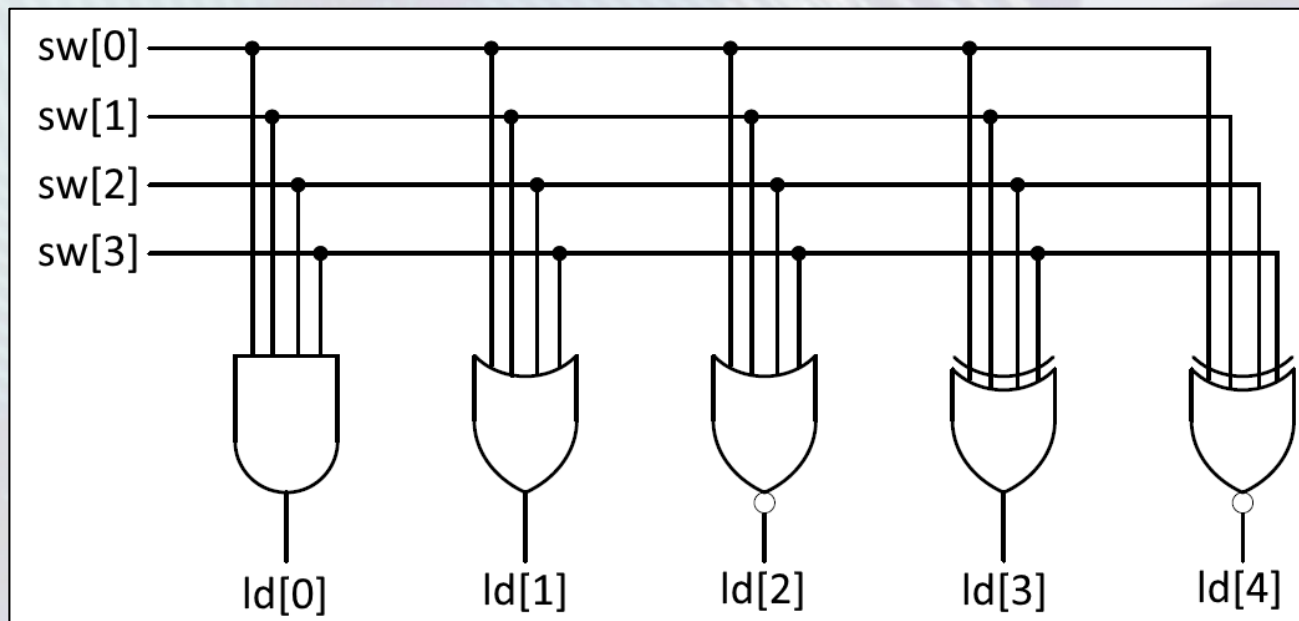


# 1. feladat: elemi logikai függvények

- Egészítsük ki a tesztkörnyezet vázat és szimuláljuk a rendszer működését
- Ha a szimuláció során mindent rendben találtunk, akkor próbáljuk ki a működést az FPGA kártyán is

## 2. feladat: elemi logikai függvények

- Ugyanaz, mint az 1. feladat, de a logikai kapuk most 4 bemenetűek lesznek
- A megvalósítandó rendszer kapcsolási rajza (a nem használt LED-ek legyenek kikapcsolva)



## 2. feladat: elemi logikai függvények

Az alábbi speciális értékek, illetve tulajdonságok jelzésére mely logikai függvények használhatók?

0 érték

$2^N - 1$  érték

Két bit egyenlősége

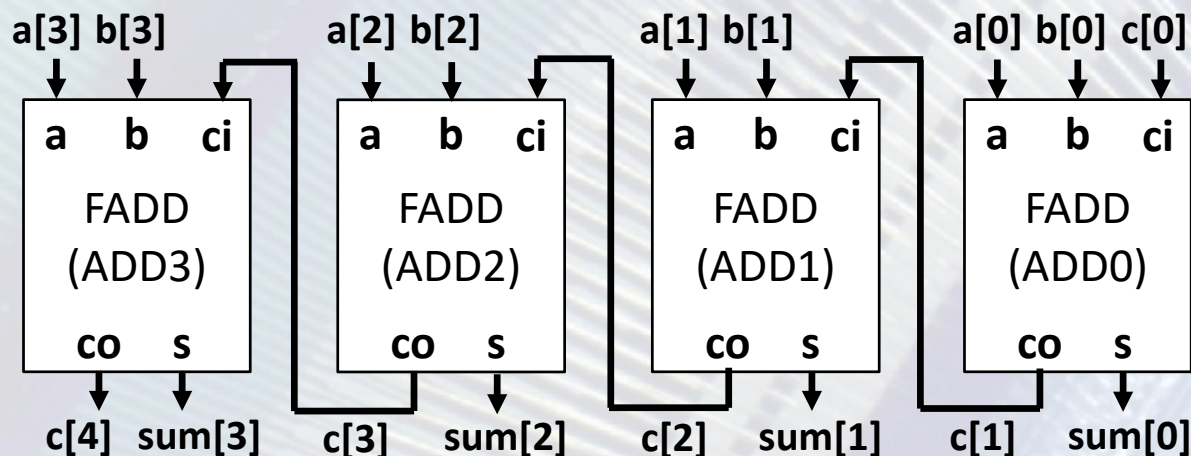
Két bit nemegyenlősége

Páratlan paritás

Páros paritás

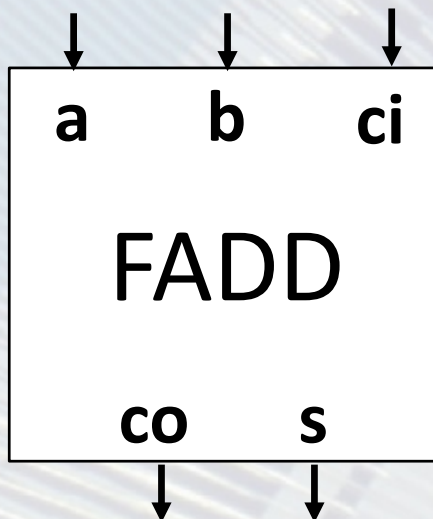
# 3. feladat: hierarchikus tervezés

- Teljes összeadókból készítünk egy 4 bites összeadót
- A 3. feladathoz szükséges új Verilog ismeretek
  - Modulok példányosítása
- A rendszer blokkvázlatát az alábbi ábra mutatja
  - $sw[3:0] \rightarrow a[3:0]$        $sw[7:4] \rightarrow b[3:0]$
  - $sum[3:0] \rightarrow ld[3:0]$



# 3. feladat: hierarchikus tervezés

- Nyissuk meg a kiindulási projektet
- Először a teljes összeadót (FADD modul) készítjük el
  - Mik a kimenetek logikai függvényei?
  - Használjuk az ábrán látható port neveket





# 3. feladat: hierarchikus tervezés

- A top-level modulban hozzunk létre 4 FADD modul példányt és kössük be a portjaikat
  - A kaszkádosítást az átvitel bitekkel végezzük el  
 $c[0] (=?) \rightarrow ADD0 \rightarrow c[1] \rightarrow ADD1 \rightarrow c[2] \rightarrow ADD2 \rightarrow c[3] \rightarrow ADD3 \rightarrow c[4]$
  - Operandusok:  $sw[3:0]$  és  $sw[7:4]$
  - Eredmény 4 biten:  $ld[3:0]$
- Modul példányosítása  
***modul\_név példány\_név(port-jel társítás lista);***
- Port-jel társítási lista  
***.port\_név1(jel\_név1), .port\_név2(jel\_név2), ...***

# 3. feladat: hierarchikus tervezés

**Szimulációval ellenőrizzük a működést**

- **Tegyük ki a 4 bites összeadandókat az idődiagramra**
- **A 4 bites összeadandók és a 4 bites kimenet számrendszere (radix) legyen előjeles decimális**
- **Mit tapasztalunk egyes bemeneti értékeknél?**
  - Mi lehet ennek az oka?
  - Segítség: mi a számaábrázolási tartomány 4 bites kettes komplementes kódolás esetén?
  - Hogyan lehetne ezt „kijavítani”?

# 3. feladat: hierarchikus tervezés

- A LED port méretét növeljük meg 5 bitre:  $c[4] \rightarrow Id[4]$
- Szimulációval ellenőrizzük ismét a működést
- Próbáljuk ki az összeadót az FPGA kártyán előjel nélküli és kettes komplementes kódolású adatokkal
- Megjegyzés
  - *Az aritmetikai műveleteket Verilog nyelven sohasem így, hanem a megfelelő aritmetikai operátorokkal (+, -, \*) valósítjuk meg*
  - A feladat célja csak a hierarchikus tervezés bemutatása volt