

## Digitális technika II. (vimia111)

### 2. gyakorlat: Komparátor, Háromállapotú meghajtó, PLA, Regiszter, Shiftregiszter, Számláló

#### Elméleti anyag:

- Komparátor
- Összeadó
- PLA-k: felépítési elvük, kétszintű diszjunktív minimalizálás, hazárdmentesítés, méretkorlátok
- Regiszterek: élvezérelt és szintvezérelt (latch)
- Shiftregiszter
  - Használata S-P és P-S átalakításra
  - Használata számlálóként: gyűrűs, Johnson, LFSR
- Számlálók
  - Modulusok: 16, 10, 12
  - Órajelezés: szinkron, aszinkron (soros órajelű)
  - Vezérlési variációk: engedélyezés, egy-, kétirányú számláló
  - Alaphelyzetbe állítás: CLEAR, LOAD, szinkron, aszinkron
  - A kaszkádosítás támogatása: végállapot dekódolása, kapuzott órajel
- Számlálók kaszkádosításának módszerei
- Számlálók modulusának csökkentése
- 

#### Irodalom:

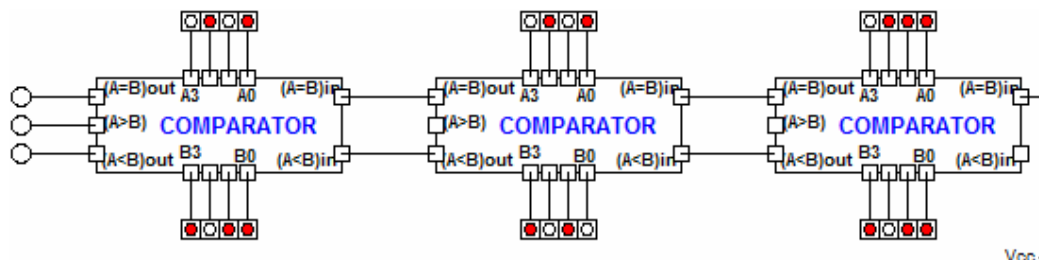
Benesóczky Zoltán: Digitális tervezés funkcionális elemekkel és mikroprocesszorokkal, egyetemi tankönyv, MK55033, 15-33.; 53. old.

#### Gyakorló példák:

**A gyakorlatokhoz kidolgozott DW példák a gyakorlathoz tartozó Segédlet könyvtárban található.**

**2.1.** Tesztelje le a DW komparátor makróját! Kaszkádosítson sorosan három komparátor IC-t.

Megoldás: Először egy egységet vizsgáljon, majd kaszkádosítson hármat az alábbiak szerint. /komparator\_12\_bits.dwm:



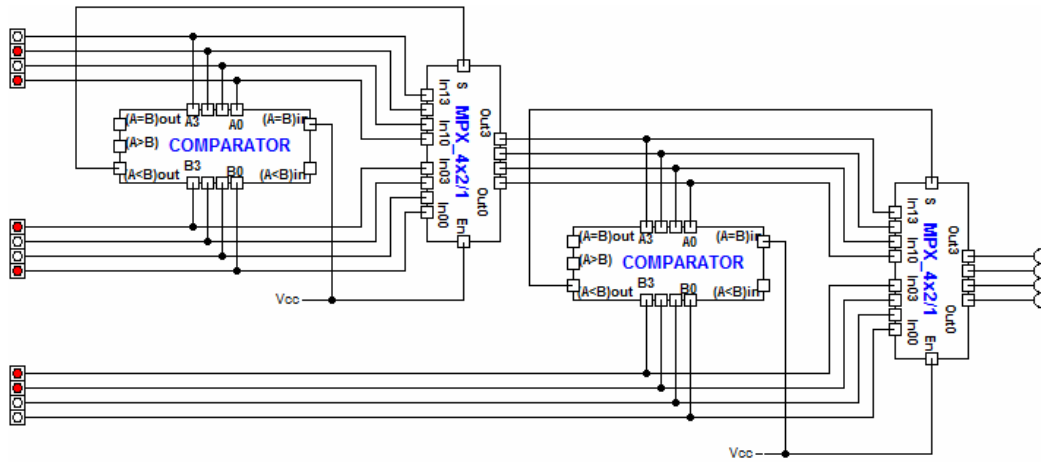
**2.2.** Az előző példa folytatásaként tervezzen 12 bites számokat összehasonlító komparátort MSI IC-kből! Az egységnek két vezérlő biten megmondható, hogy az összehasonlítandó számok 1-es, 2-es komplementes, vagy előjeles abszolút értékes kódolásúak.

Megoldás: a fenti kapcsolás közvetlenül használható off-set kódú előjeles számok összehasonlítására. Így célszerűen a többi kódolásból off-set kódra fogunk áttérni.

**2.3.** Készítsen áramkört három négy bites szám közül a legkisebb kiválasztására

Megoldás: Egy komparátorral és egy 2/1-es MPX-rel kettő közül a kisebbik kiválaszható, egy további ilyennel a „kisebb” és a harmadik közül kisebb is kiválasztható.

**/harombol\_legkisebb.dwm:**



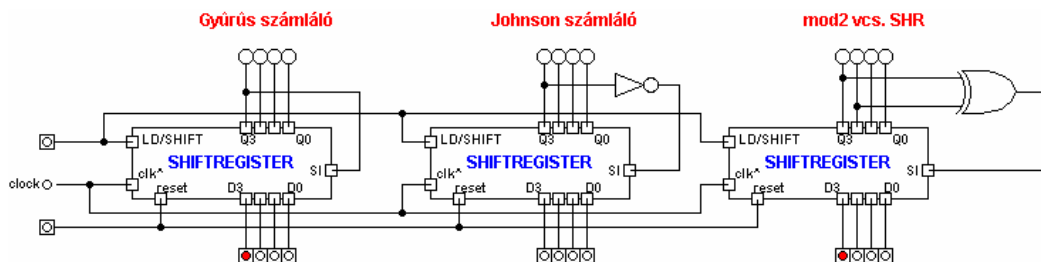
**2.4.** Elemezze az új DW könyvtár shiftregisztereinek működését!

Készítsen belőlük:

- Johnson számlálót
- Gyűrűs számlálót
- Max hosszúságú számlálót
- Kaszkádosítson három db. kétirányú SHR-t!

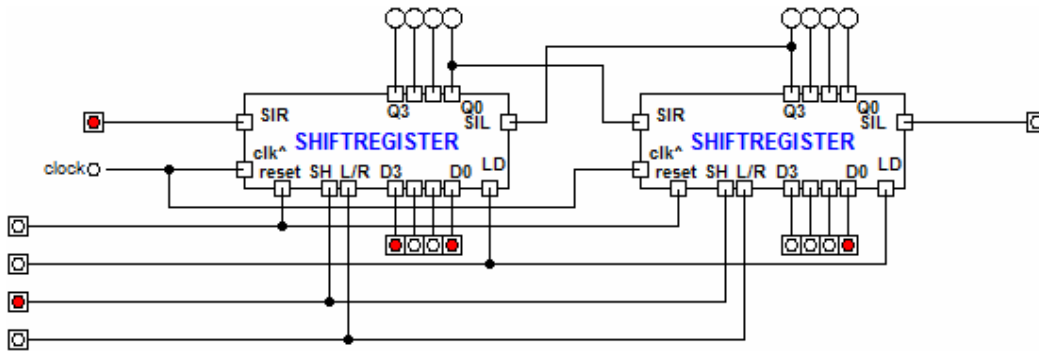
Megoldás a kétfunkciós shiftregiszterrel:

**/SHR\_szamlalok.dwm:**



És két négyfunkciós SHR kaszkádosítása:

**/SHR\_8\_bites.dwm:**



2.5 Ötbités shiftregiszternél csatolja vissza a 3. és 5. bit mod2 kapcsolatát! Milyen ciklusokat talál?

Megoldás: Két ciklus van!

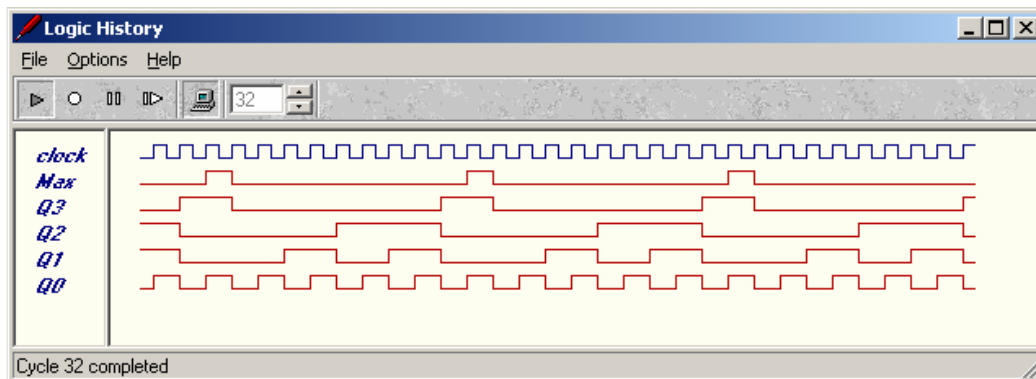
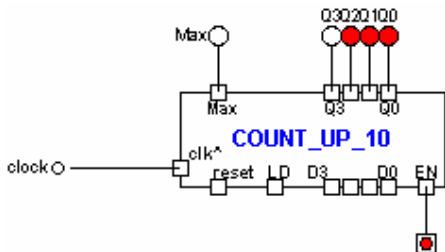
- 00000-ból indulva 1 hosszúságú a ciklus,
- 00001-ből indulva 31 hosszúságú a ciklus,

ezzel megvan a 32 állapot

2.6. Rajzolja fel egy 10-es számláló kimeneteinek idődiagramját!

Megoldás:

A COUNT\_UP\_10-et kipróbáló kapcsolás és az általa kiadott idődiagram:



2.7. Készítsen a COUNT\_U/D\_10-es dekadikus számláló egységekből 3 dekadós, engedélyezhető kétirányú számlálót!

Megoldás:

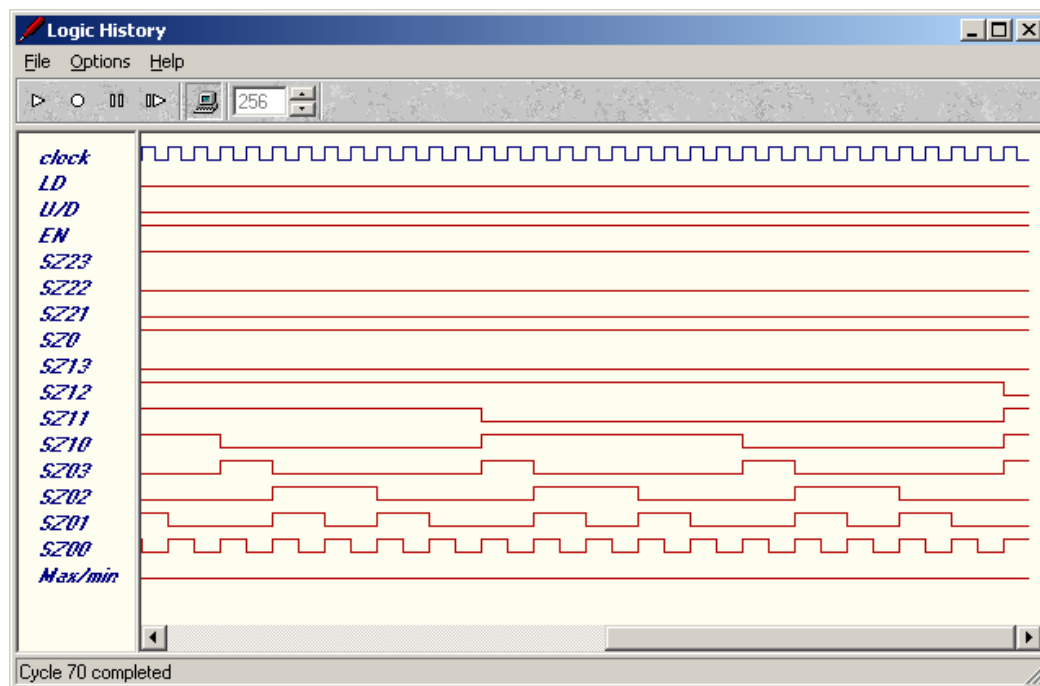
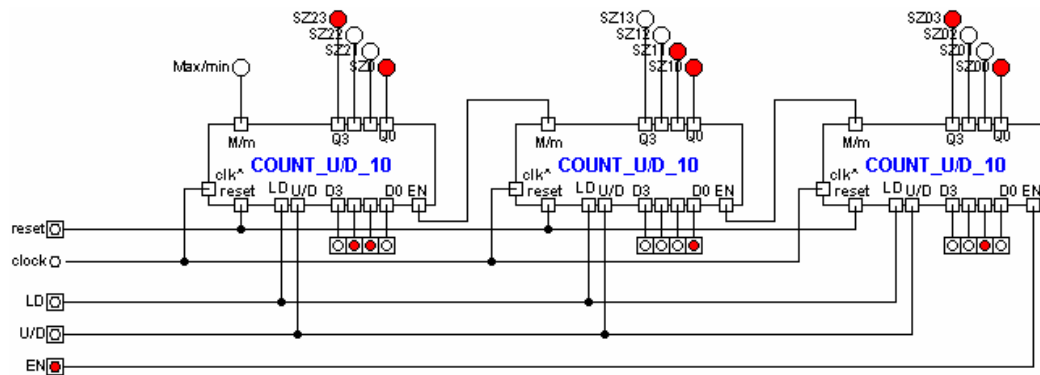
Ez egy kaszkádosítási rutinfeladat. Megoldásához az adott számláló tulajdonságait kell kigyűjteni:

- Engedélyezhető, kétirányú 10-es számláló
- Szinkron load (betöltés)
- A kaszkádosítást állapotdekódolt Max/min jellel (bele van kapuzva az Engedélyezés!) támogatja

Mindez az alábbi kapcsolási rajzon megfigyelhető.

A kaszkádosítás kapcsolása:

LOAD, D/U, CLK párhuzamosan, az engedélyezést pedig fokozatról fokozatra fel kell építeni, mivel a Max/min kimenetben és kapcsolattal benne van az EN, ezért egyszerű lánckapcsolás a kaszkádosítás.



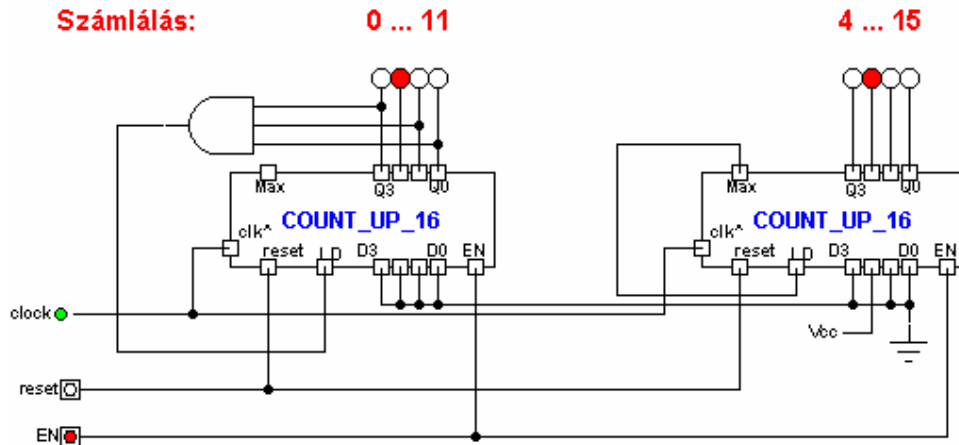
A fenti kapcsolás megtalálható:  
...\3\_dekados\_szamlalo.dwm

**2.8.** Csökkentse le a modulusát a COUNT\_UP\_16-os számlálónak 12-re, több megoldást is készítsen!

Megoldás:

Több lehetőség van:

- A 11-es állapotban 0 betöltése (a ciklus 0-tól 11-ig tart)
- A végállapotban 4 betöltése (a ciklus 4-től 15-ig tart)
- Az n. állapotban m betöltése, itt  $\text{mod } 16(n-m)=11$



A fenti kapcsolások közül az első kettő megtalálható:

[...\modulus\\_csokkentés\\_12-re.dwm](#)

**2.9.**

**a.** Készítsen programozható impulzusszélesség modulátort! A modulátor 100 óraütemenként periodikusan ismétlődő jelet állít elő. Egy perióduson belül először 100-N ütemig 0, utána N ütemig 1-es érték van.

**b.** Hogyan kell módosítani a megoldást, ha a periódus a fix 100 helyett beállítható M értékű?

Megoldás:

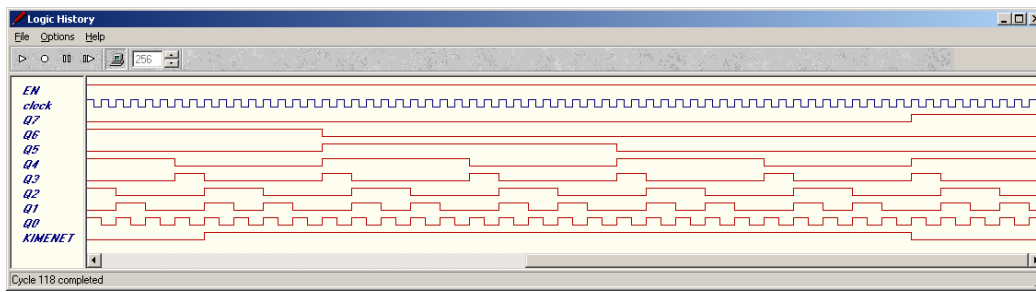
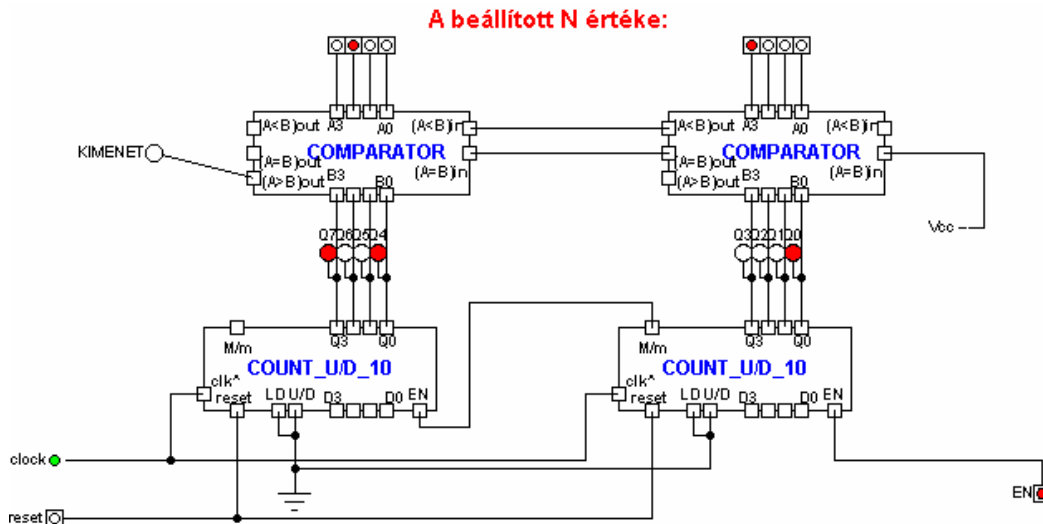
**a.** A több lehetséges megoldás közül most a következőt választjuk:

- Készítünk egy 100-as lefelé számlálót, kimenete SZ,
- Ezt összehasonlítjuk a kívülről beállított N-nel
- És ha  $N > SZ$ , akkor kijelzünk.

A számláló kiválasztása: kétirányú decimális számláló kell, más különleges igény nincs, ehhez készletünkben megfelelő a COUN\_U/D\_10 kétirányú decimális számláló. Két ilyen decimális számlálót kaszkádosítani kell.

Komparátor kiválasztása: készletünkben az általános COMPARATOR kiválóan megfelel, ebből kettőt kell kaszkádosítani.

A kapcsolási rajz itt látható.



^47. ütem

^0. ütem

Az előző idődiagram mutatja, hogy a KIMENET a lefele számláló 47-es ütemétől a 00-ás ütemig lesz 1 értékű, tehát valóban a beállított 48 ütem hosszú impulzust adunk ki.

Megjegyzések: Ebben a megoldásban N értéke 00...99 között változhat. Ha az  $N = 100$ -at (a KIMENET állandóan 1-es) is meg akarjuk engedni, akkor N kódolásához egy további bitet (N8) is fel kell venni és a komparátor kimenetét VAGY kapcsolatba kell hozni ezzel az új bittel. Ennek részletes kidolgozását az olvasóra bízunk.

A fenti kapcsolás megtalálható:

...[\impszelessegmodulator.dwm](#):

**b.** A funkcionális blokkvázlat bővül: dekódolni kell egy komparátorral a számláló megfelelő (M-1) állapotát és szinkron módon be kell tölteni 0-át.

Kérdés: mi történik, ha  $N > M$ ?

### Nehéz példák az érdeklődőknek:

**2.n1.** Négy szám közül a második legnagyobbat akarjuk kiválasztani komparátorok és multiplexerek segítségével. A MPX és a komparátor arányától függően mikor hogyan csináljuk?

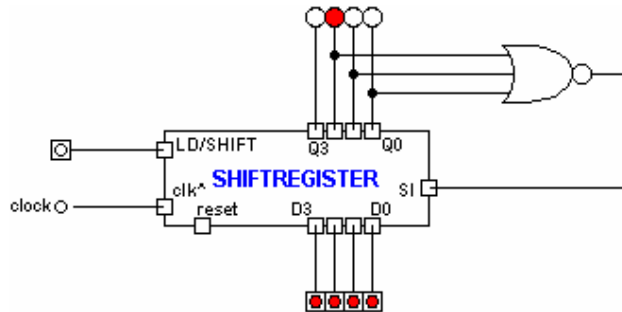
**2.n2.** Négybites shiftregiszter visszacsatolásával tud-e 16-os számlálót csinálni?

Igen, a negyedik és a harmadik bitet mod2 visszacsatolva 15-ös számlálót (max. hosszú LFSR-t) kapunk. Ennek 0001 és 1000 állapotai közé kell befűzni a 0000 állapotot. Tehát a 0001 és 0000 állapotokban invertálni kell az általános mod2 visszacsatolást!

**2.n3.** Készítsen n-bites gyűrűs számlálóhoz önkorrigáló áramkört csupán a soros bemenet felhasználásával!

Megoldás 4 bitre:

**/onkorr\_gyurus\_szamlalo.dwm:**



Ezen kipróbálható a LDMSHIFT vezérlésével bármilyen állapotból történő indítás.