Digit kódom: 7235416

1. A ciklusok felderítése

A digit kódom alapján keletkező visszacsatolás:
SI = Q6 mod2 Q2 mod2 (Q3*Q5) mod2 (Q5*/Q4) mod2 (Q4+/Q1)

A ciklusokat egy Google Táblázat[1] segítségével találtam meg. Az összes függvényem másolható, így csak egy-egy prototípust mutatok be, amik a jelölt cellákban helyezkednek el.

A Táblázat első oszlopában a lehetséges kódszavak vannak felsorolva, a második oszlopba pedig a visszacsatolással kiszámolt következő kódszó található. A visszacsatolást egy segédtáblázattal számoltam ki a következő függvénnyel:

```
=MOD( C4
        +INDEX(C4:H4;1;7-'DIGIT2 HF'!F$1)
        +AND(
                INDEX(C4:H4;1;7-'DIGIT2 HF'!G$1);
                INDEX(C4:H4;1;7-'DIGIT2 HF'!H$1))
        +AND(
                INDEX(C4:H4;1;7-'DIGIT2 HF'!H$1);
                NOT(INDEX(C4:H4;1;7-'DIGIT2 HF'!I$1)))
        +OR(
                INDEX(C4:H4;1;7-'DIGIT2 HF'!I$1);
                NOT(INDEX(C4:H4;1;7-'DIGIT2 HF'!J$1)))
        ;2)
```

Ahol: a C4:H4 tartomány az előző kód binárisan,
a ‘Digit2 HF’ munkalap megfelelő cellái pedig a digit kód számjegyei

Ezután az első ciklust nullával indítottam, majd az első ciklus maradékát a következő függvény állítottal elő:

```
=IF(    OR(F4="";F$4=VLOOKUP(F4;$A$4:$B$67;2;true));
        "";
        VLOOKUP(F4;$A$4:$B$67;2;true))
```

Ahol a VLOOKUP a kijelölt tömb első oszlopában keres egy feltétel szerint, majd visszatér a sor adott sorszámú mezőjével.
Azaz: Ha a felette lévő üres vagy meg kéne egyeznie a legelső elemmel, akkor nem ír ki semmit, ellenkező esetben pedig a második oszlopból másolja le, hogy mi következik.

---

[1] http://docs.google.com oldalon elérhető online táblázatkezelő alkalmazás. Az esetlegesen nem részletezett függvények leírása a következő oldalon található: https://support.google.com/docs/bin/static.py?hl=en&topic=25273&page=table.cs

A következő ciklus első eleméhez újabb segédtáblázatot vettem fel, ami kigyűjti azokat az elemeket, amik még nem szerepeltek az eddigi ciklusokban.

```
=IF(
        ISERROR(
                MATCH('DIGIT2 HF'!A4;'DIGIT2 HF'!F$4:F$67;0));
        'DIGIT2 HF'!A4;
        "")
```

Azaz: ha hibát talál a kereső függvény (azaz, nincs benne az első ciklusban a szám), akkor kiírja a számot, egyébként pedig nem ír ki semmit.
Természetesen a későbbiekben csak azt vizsgáltam, ami eddig nem szerepelt, egy hasonló felépítésű függvénnyel.

Ebből az oszlopból már könnyen meg lehetett találni, hogy mi a legkisebb érték a következő függvénnyel:

```
=IF(    MIN(CIKLUS!B2:B65)=0;
        "";
        MIN(CIKLUS!B2:B65))
```

Azaz: ha a legkisebb találat nulla (tehát üres az oszlop), akkor nem kell új ciklust indítani, így nem írunk oda semmit, ha pedig találtunk számot, akkor azt másoljuk oda.
Ennek a függvénynek egy módosulata írja ki a “Mind meglett” szöveget, azaz jelzi, hogy nem maradt felhasználatlan állapot.

A jobb oldali ciklushossz számlálót egy egyszerű függvénnyel valósítottam meg:
`=COUNT(F4:F67)`

A Karnaugh-táblába a következő függvénnyel beírtam, hogy az adott állapot melyik ciklusban van, majd feltételes formázással kiszíneztem a mezőket.

```
=IF(ISERROR(MATCH(X$20+$W21;$F$4:$F$67;0));
        IF(ISERROR(MATCH(X$20+$W21;$J$4:$J$67;0));
                IF(ISERROR(MATCH(X$20+$W21;$N$4:$N$67;0));
                        IF(ISERROR(MATCH(X$20+$W21;$R$4:$R$67;0));
                                IF(ISERROR(MATCH(X$20+$W21;$V$4:$V$67;0));
                                        "X";
                                        5);
                                4);
                        3);
                2);
         1)
```

Azaz: végignézi, hogy sikerült-e megtalálni a megfelelő oszlopban, ha igen beírja a számot, ha nem a következőben keresi tovább

2. Az önkorrigáló függvény meghatározása

Az alábbi Karnaugh táblába bejelölve a ciklusokat olyan primimplikánst keresünk, ami legalább egyszer tartalmazza a “mellékciklusok” 1-1 tagját és nem tartalmazza a fő ciklusunk egyik tagját sem.

1	1	2	1	2	2	3	1
2	1	3	3	2	2	1	2
3	2	1	1	2	2	2	2
1	2	1	3	3	2	4	3
2	3	2	2	1	2	2	1
2	2	2	2	2	2	2	2
1	3	2	**4**	2	2	1	3
1	2	2	**3**	1	2	3	2

A kiemelt primimplikáns (/Q1./Q3./Q5.Q6) eleget tesz a feltételeknek. Vizsgáljuk meg, hogy lehetséges-e vele a soros önkorrigálás.

1. ciklus:
32-nél 0 helyett 1 következik, ha invertájluk a soros inputot, azaz csak kihagy egy állapotot
40-nél 16 helyett 17 következik, ha invertáljuk a soros inputot, azaz átjut a 2. ciklusba

2. ciklus:
Ez a fő ciklus, nincs olyan állapot, ami rendellenesen befolyásolná a soros inputot

3. ciklus:
34-nél 5 helyett 4 következik, ha invertáljuk a soros inputot, azaz átjut az 1. ciklusba

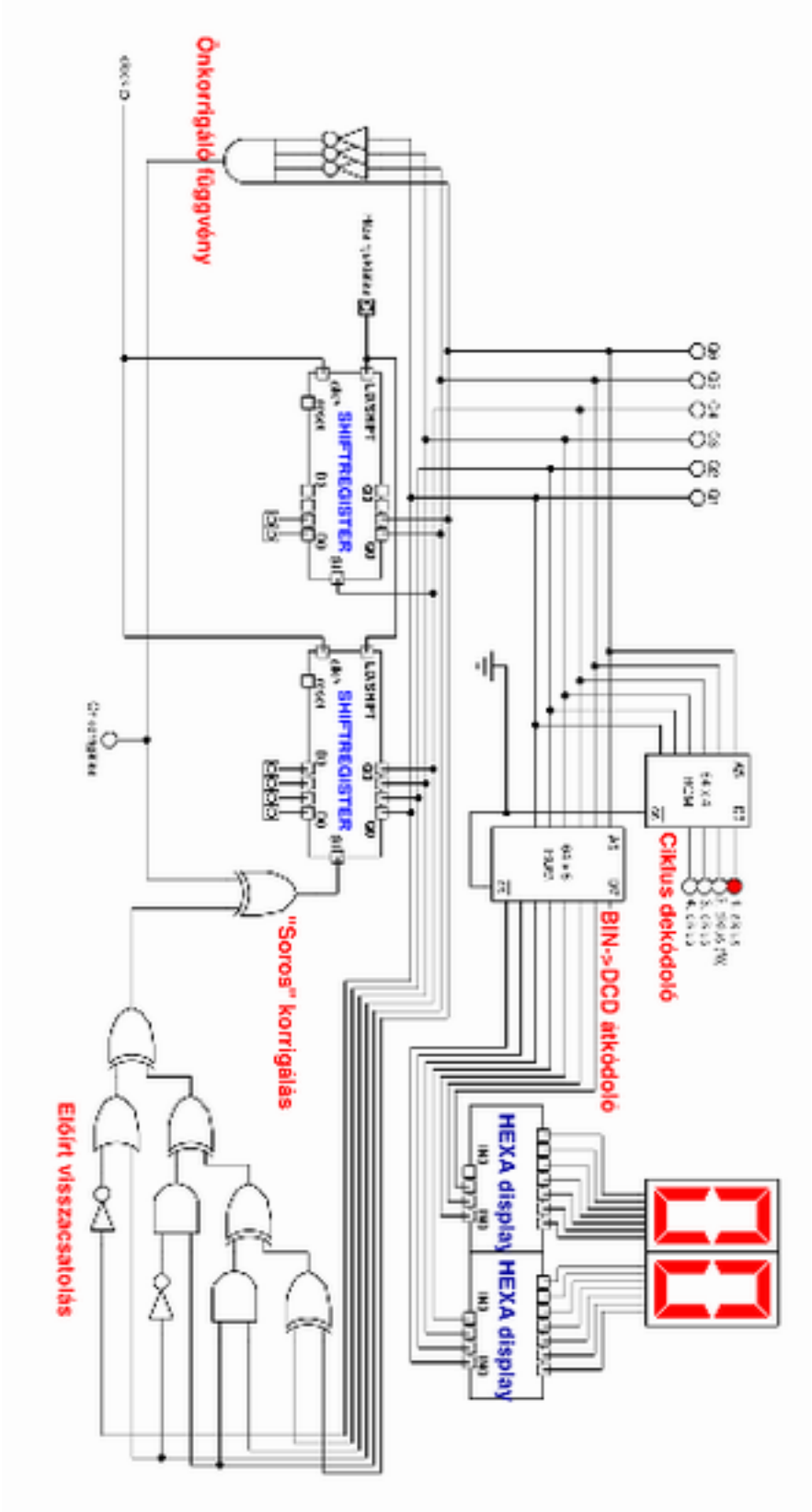
4. ciklus:
42-ből 21 helyett 20 következik, ha invertáljuk a soros inputot, azaz átjutunk a 3. ciklusba

Tehát a talált korrigáló függvény használható soros korrigálásra, mert minden esetben visszajutunk a fő ciklusba.

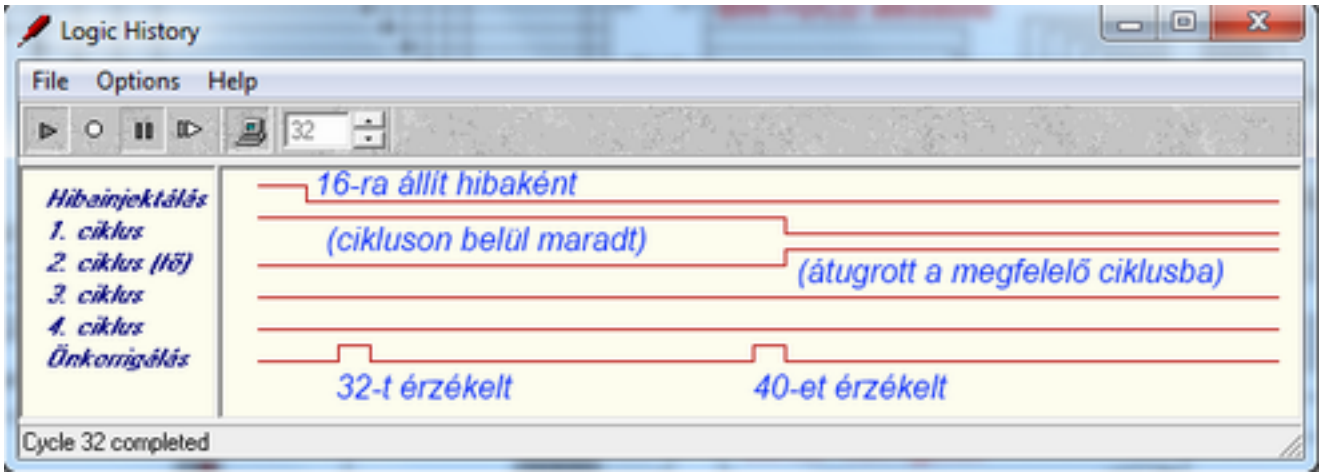
4. Kapcsolási rajz

A megvalósításhoz jól használható a függvénykönyvtárban megtalálható Shiftregiszter.

5.



Idődiagramok



Állapotok: 16 > 32 > 1 > 2 > 4 > 9 > 19 > 38 > 13 > 27 > 54 > 45 > 26 > 52 > 40 > 17 > 35 > 6
(1. ciklus) (2. ciklus)

Állapotok: 21 > 42 > 20 > 41 > 18 > 37 > 11 > 22 > 44 > 24 > 49 > 34 > 4 > 9 > 19 > 38 > 13 >
(4. ciklus) (3. ciklus) (1. ciklus)

27 > 54 > 45 > 26 > 52 > 40 > 17 > 35 > 6
(2. ciklus)

Az idődiagramokon látszik, hogy mind a négy megfigyelt állapotnál az előre megjósolt módon működik, így helyes a soros visszacsatolás