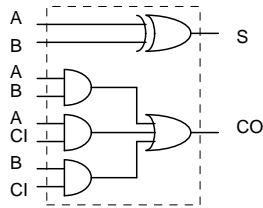


**Digit ZH Mintamegoldás**  
**2009. április 10.**

Megi.: A megoldásokat sok esetben DigitalWorks-ben is elkészítettük, ilyenkor megadjuk a .dwm fájl nevét is.

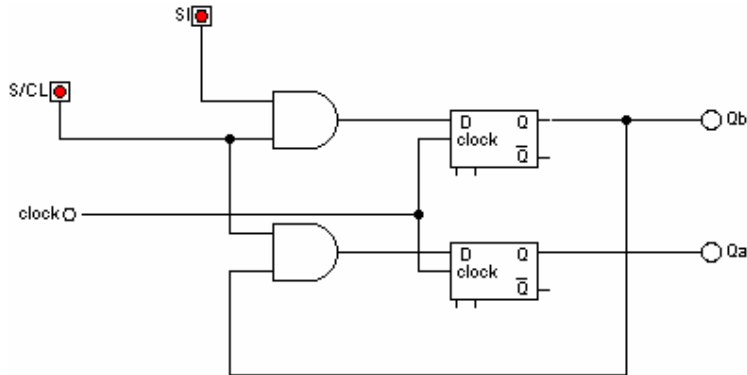
**E1.** Milyen funkcionális elem belsejét mutatja az alábbi ábra és mi az A,B,CI,S,CO-val jelölt jelek szerepe? (2p)



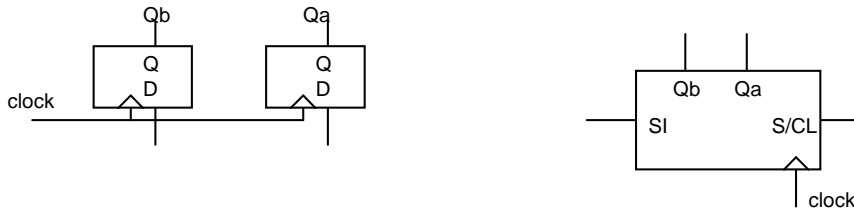
Ha az S függvényébe mod2-vel be vesszük a CI bemenetet is, akkor ez egy „**egybités teljes összeadó**”, amivel már a múlt félévben is megismerkedtünk.

**E2. a.** Egészítse ki az alábbi baloldali kapcsolást egy **szinkron** módon törölhető shiftregiszterré! A shiftregiszter jobbra shiftel, ha az S/CL=1, töröl, ha S/CL=0. A megoldáshoz tetszőleges kapukat használhat. (2p)

ZH2009\_E2a.dwm:



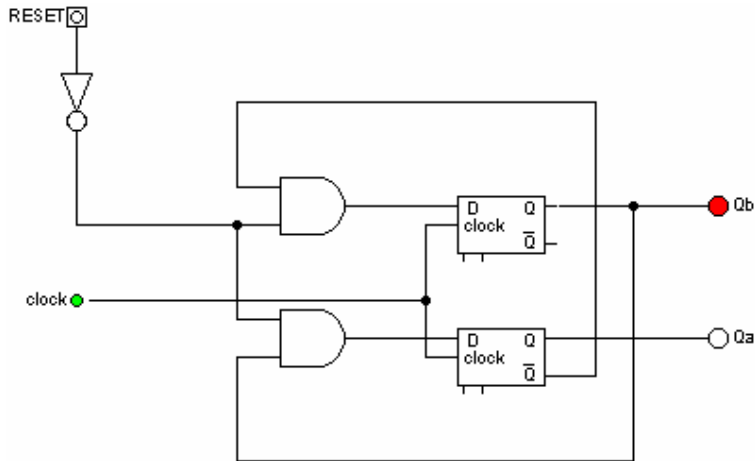
**b.** A jobboldali ábra az előző kapcsolás blokkvázlatát mutatja. Egészítse ki az áramkört 2 bites **Johnson** számlálóvá! (Egy H aktív bekapcsolási RESET jel is rendelkezésre áll, mely bekapcsoláskor 2 órajelig logikai 1-et ad.) (2p)



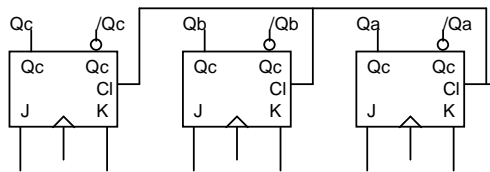
Az alábbi kapcsoláson a /Qa visszacsatolásához a soros bemenetre még csak inverter sem kell!

A RESET negált pedig kiválóan vezérli a kezdeti törlést.

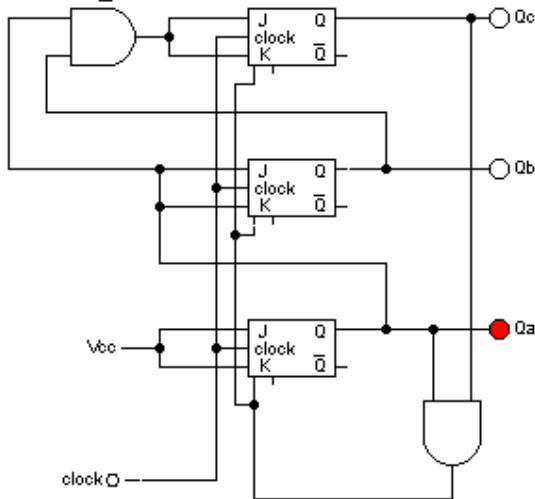
ZH2009\_E2b.dwm:



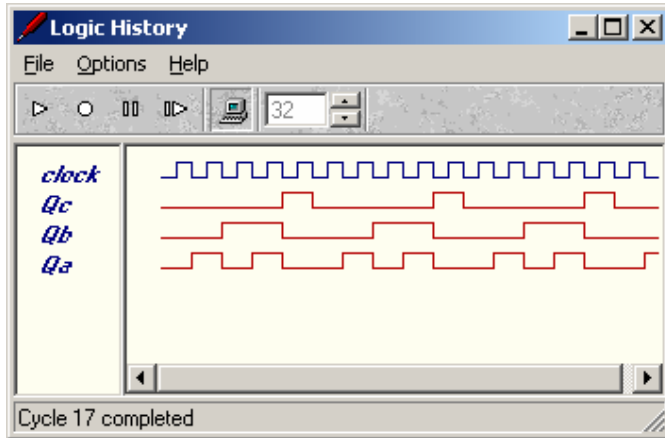
**E3.** Készítsen *szinkron* 5-ös modulusú (000,001,010,011,100,000,001...) *felfele* számlálót az alábbi 3 db JK flip-flop-os kapcsolásból! A kiegészítéshez tetszőleges kapukat és invertereket használhat fel. (A modulus csökkentéséhez a flip-flopok aszinkron Cl bemenét használja fel! Nagyon rövid időre az 101 állapot fennállhat.) (2p)



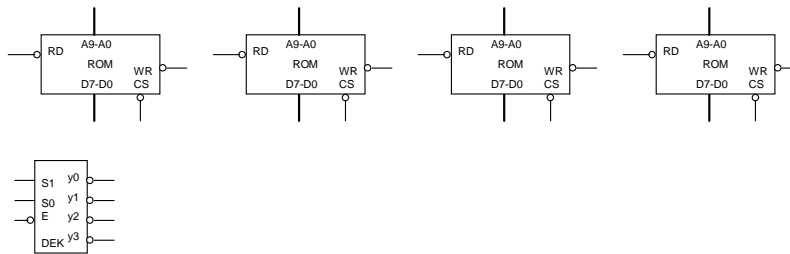
ZH2009\_E3.dwm:



- Először felépítjük a hárombités szinkron felfele számlálót (a J-K bemenetek megfelelő bekötésével),
- majd az 101 állapot dekódoltját törlőjelként visszacsatoljuk. Így „aszinkron” törlést valósítunk meg: az 101 állapot csak egy „pillanatig” áll fenn, lenullázza a számlálót és innen folytatódik a ciklus. Ez a törlési „pillanat” az iDigitlaWorks idődiagramján nem is látszik.



**E4. a.** Készítsen az alábbi 4db RAM-ból egy négyszer akkora kapacitású egységet, mint amekkora a megadott RAM chip kapacitása! A kialakítandó egység jelei: A11-A0, D7-D0, /CS, /RD, /WR (2p)



Itt a kapacitásbővítés cím irányában történik, az eredeti (10 bittel címezhető) 1K szavas, szavanként 8 bites csipekből 4 K szavas (12 bittel címezhető) szavanként 8 bites memóriatömböt állítunk elő.

Módszer: kiválasztó CS lábakat kivéve minden jelet párhuzamosan kötünk, a négy csip szelekt jeleit rendre a dekóder kimeneteire kötjük. A dekóder kiválasztó bemenete vezérli a teljes tömb kiválasztását, míg S1-S0 bemeneteire a két bővítő címbitet kell kötni (A11-A10). F2.a.

**E5.** Mely állítások igazak és melyek hamisak? Jelölje +-al az igaz, --al a hamis állításokat! (4p)

1.	Ha egy <i>hazárdmentesen megvalósított</i> 2 bites összeadó (A1A0, B1B0, Ci, Co, S1, S0) A1, A0 bemeneteire egy 2 bites <i>Johnson számláló</i> kimeneteit kapcsoljuk, a többi bemenetére pedig 0-át, akkor nem jelenik meg hazárd az összeadó kimenetein.	+
2.	Az n bites gyűrűs számláló modulusa n+1.	-
3.	A PLA-kban az <i>ÉS</i> hálózat programozható, a <i>VAGY</i> hálózat pedig nem.	-
4.	Ugyanazon feladatot megvalósító számláló típusú vezérlő és mikroprogramozott vezérlő feltétel multiplexerét összehasonlítva a számláló típusú vezérlőé kevesebb vagy ugyanannyi bemenetű mint a mikroprogramozott vezérlőé.	-

Ad1. A Johnson számláló biztosítja a bemenet „egy Hammig távolságú” vezérlését. Ezért a statikus és dinamikus hazárd ellen hazárdmentesített összeadó kimenetén nem fordulhat elő hazárd.

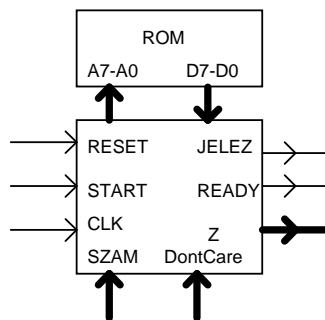
Ad2. A modulus n.

Ad3. PLA-kban mindkét hálózat programozható.

Ad4.

- A mikroprogramozott vezérlőnél annyi MPX bemenet kell, ahány FELTÉTEL-JEL van
- A számláló típusúnál annyi, ahány ÁLLAPOT van
- A FELTÉTEL JELEK száma nem lehet több az ÁLLAPOTOK számánál.

**F1.** Tervezze meg egy olyan egység részletes funkcionális blokkvázlatát, amely megkeresi egy 256 byte-os ROM-ban levő, az alább megadott feltételeket kielégítő *első* adatot és kiadja ennek címét a kimenetén. Ha talált ilyen számot, azt a JELEZ kimeneten adott 1-gyel mutatja. Ha végzett a feladattal, azt a READY kimeneten adott 1-el jelzi és leáll a működés. A megtalált szám címe az Z kimeneten jelenik meg. A feltételek: *a megkeresendő szám az egység SZAM bemenetére adott 8 bites adat. Az egység a keresésnél csak a szám azon bitjeit veszi figyelembe, amely bitekhez tartozó azonos sorszámú bit a DontCare bemenetre adott adatban 1 értékű. Pl. SZAM=10110100 és DontCare=01111110 esetén a SZAM 7. és 0. bitjét figyelmen kívül hagyja az áramkör, tehát a figyelt adat: x011010x.* A feltétel logikát a blokkvázlaton egy blokk jelölje, s külön rajzolja le a blokk részletes belső felépítését. A folyamat a START bemeneten érkező, legalább egy órajel periódus ideig tartó impulzus hatására indul. Az áramkör összes bemenete és kimenete magas aktív. Az Z kimenet értéke a feltétel teljesüléséig tetszőleges. Az áramkört egy a bekapcsoláskor aktivizálódó RESET jel hozza alaphelyzetbe.



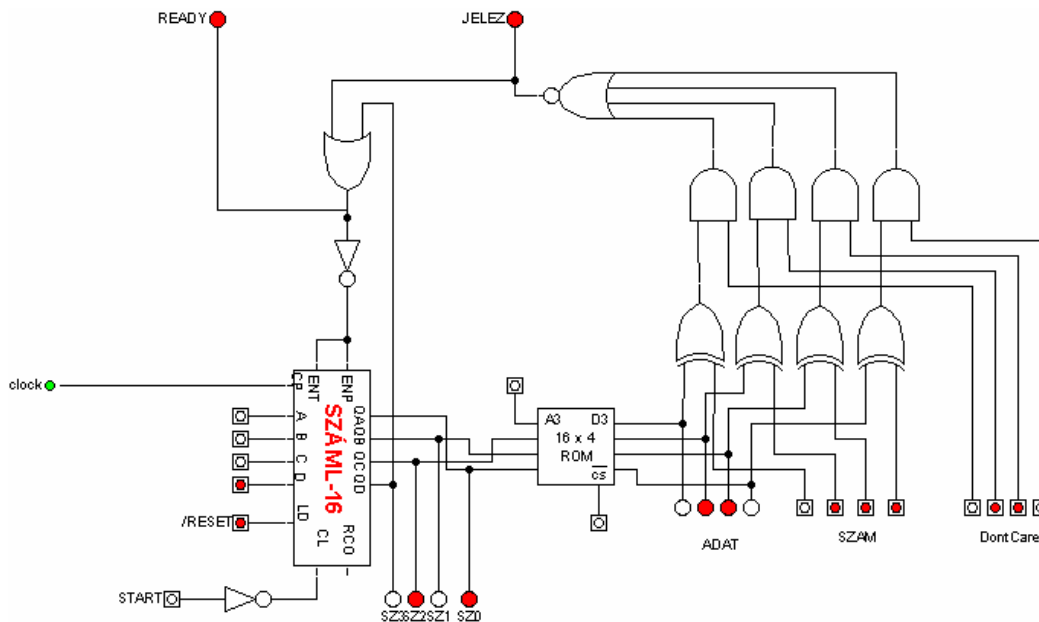
a. Külön lapon rajzolja le a megoldását! A feladatot kétféleképpen is megoldhatja:

- Egyetlen számlálóra alapozott random logikával (ekkor a teljes kapcs. rajzot kérjük)
- vagy adatstruktúra vezérlő szemlélettel (ekkor a részletes adatstruktúrát és a vezérlés folyamatábráját kérjük). (13p)

Mi itt az első változatot dolgozzuk ki, a második ebből könnyen transzformálható.

A teljes kapcsolást 8 szavas, szavanként 4 bites ROM-mal az alábbi ábra mutatja:

ZH2009\_F1.dwm:



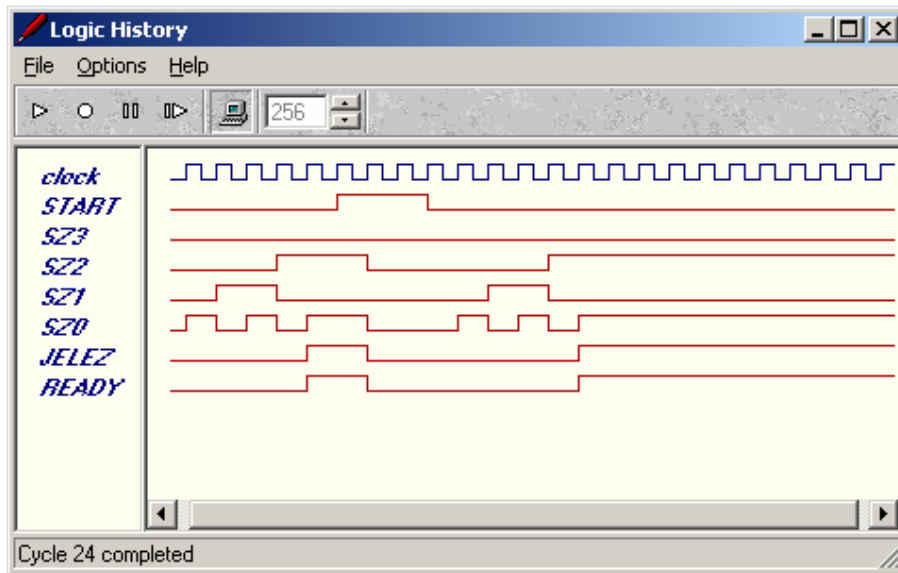
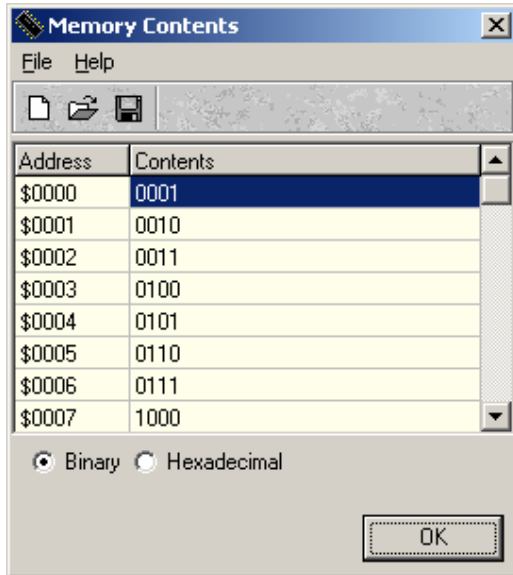
Az alapelv az előadáson elhangzottaknak megfelelően:

- Egy bittel hosszabb számlálót választunk (jelen esetben  $3+1=4$  biteset), a 4. bit bebillenése jelzi a ROM-on történő végighaladást. Ebbe az alapállapotba visszük a számlálót a bekapcsolási RESET-tel. Ilyenkor a számláló a  $READY=1$  miatt nincs engedélyezve.
- A START törli és ez egyben engedélyezi is a számlálót, ami végigpásztázza a ROM-ot vagy az első megtalált kedvező adatig ( $JELEZ=1$ ), vagy a címtartomány végéig.

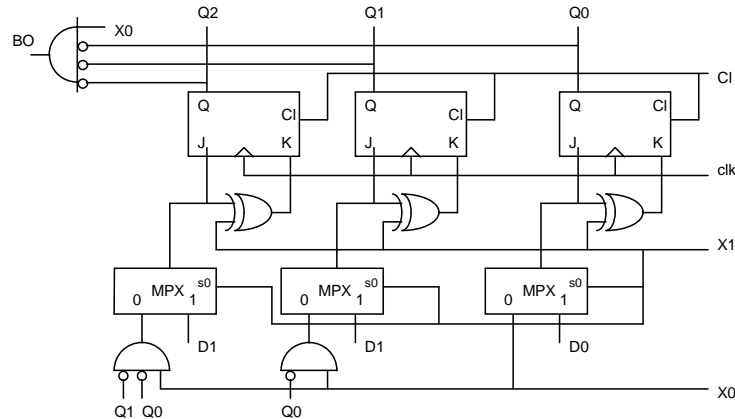
Az általános séma mellett egyedi feladat a keresett adat felderítése. Ennek – sok más lehetséges megoldás mellett – egy egyszerű változata az ábrán látható:

- Az ADAT-ot és a SZAM-ot bitenként mod2-zük (az egyezést a 0 mod2 kimenet jelenti,
- A DontCare maszkkal bitenként És kapcsolatba hozva ezeket, az És kapu kimenetek 0-val jelzik, ha az adott sorszámú biteket nem kell figyelembe venni, VAGY ha egyformák.
- Az összes ilyen kimenet VAGY kapcsolata akkor 0, ha az ADAT a feltételeknek megfelelő, ezt invertálva kapjuk a JELEZ jelet.

A fenti kapcsolást az alábbi ROM tartalommal teszteltük és az áramkör nagyon helyesen az 5-ös ROM címen találta meg az első keresett adatot. Itt aztán meg is állt, amint a mellékelt idődiagram is mutatja.



F2. Adott az alábbi kapcsolás, amely egy számlálót valósít meg. (10p)



a. Adja meg, hogy X1X0-tól függően mit csinál a kapcsolás! (4p)

X1 X0	működés
0 0	marad
0 1	lefele számlál
1 0	betölt
1 1	betölt

b. Mekkora a modulusa az egység számlálójának? (1p) .....8.....

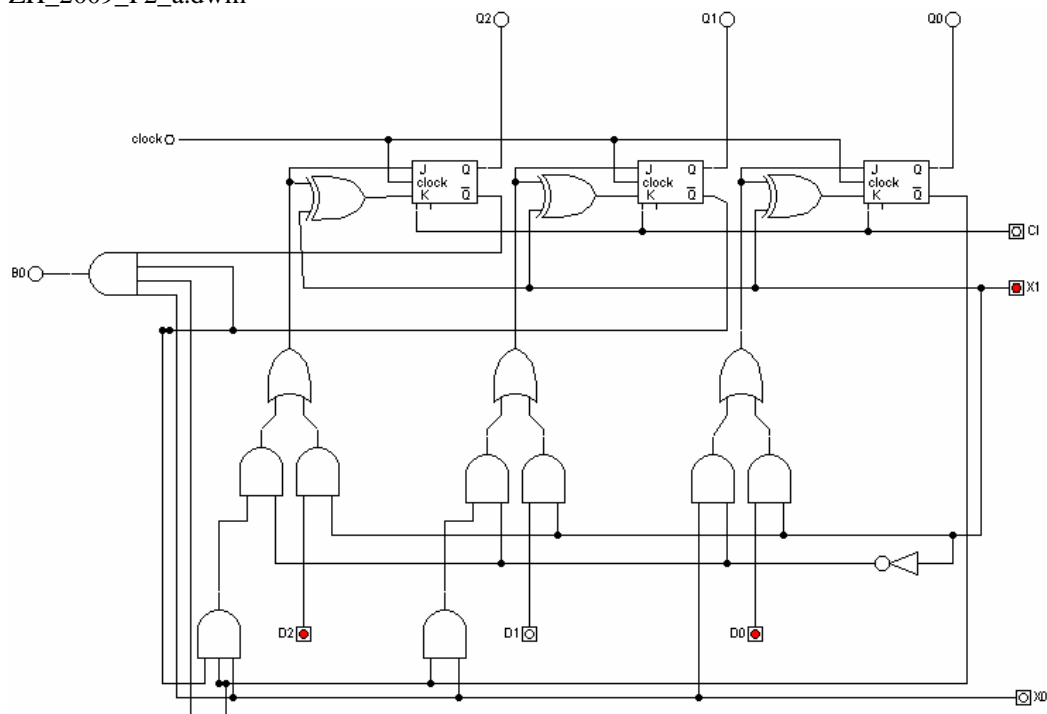
c. Fel vagy le számlál a számláló üzemmódban? (1p) ...lefele...

A kapcsolás megfejtésének lépései:

- X1 szerepe kettős
  - MPX címző üzemmód választás:
    - X1=0: számlálás ág kiválasztása
    - X1=1: D2-D0 ág kiválasztása
  - FF üzemmód választás:
    - X1=0: a JK FF-ből T FF lesz
    - X1=1: a JK FF-ből D FF lesz
- X0 szerepe az „engedélyezés”,
  - ha X0=0, akkor a számlálás ág 0-kat ad a T FF-oknak, így azok helyben maradnak
  - ha X0=1, akkor mehet a számlálás
- mivel az ÉS kapukba a Q negáltak vannak bekötve, ezért ha az egység számlál, akkor csak lefele számlálhat
- a BO kimenet szerepe az engedélyezés kaszkádosítása

Fentiek ellenőrizhetők az alábbi kapcsoláson:

ZH\_2009\_F2\_a.dwm

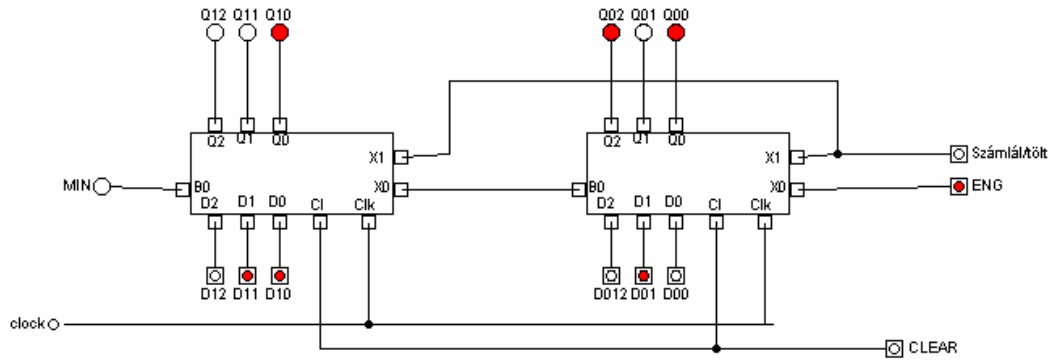


Ezt a kapcsolást ZH2009\_F2\_macro.dwm formában mentettük el, az alábbi kapcsolásban már ezeket használjuk.

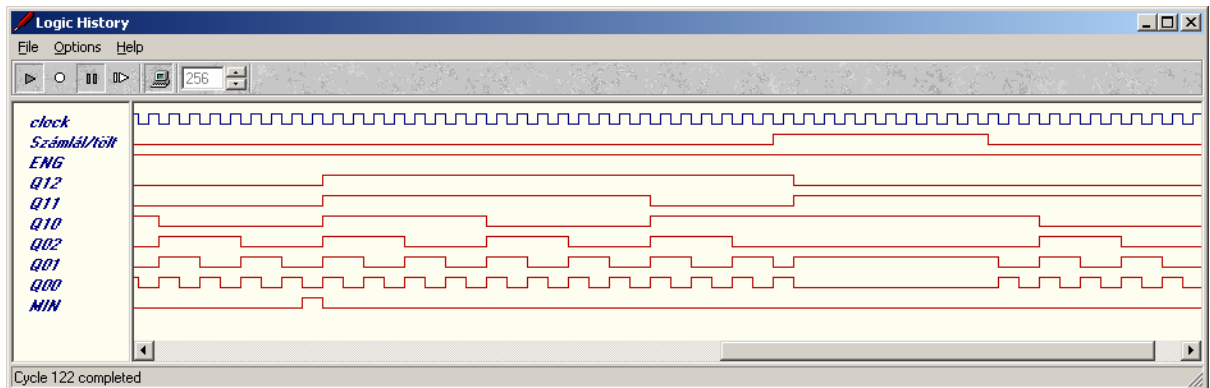
**d.** Rajzolja le, hogyan kaszkádósítana két fenti elemet, hogy azok együtt egy ugyanolyan funkciójú, 6 bites egységet alkossanak? (4p)

Az üzemmódváltást, a törlést, és az órát párhuzamosan kell kötni, az engedélyezést pedig sorosan kell kaszkádósítani (l. a következő ábrát!).

ZH\_2009\_F2\_d.dwm

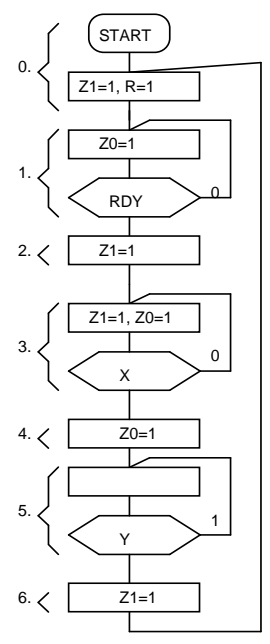
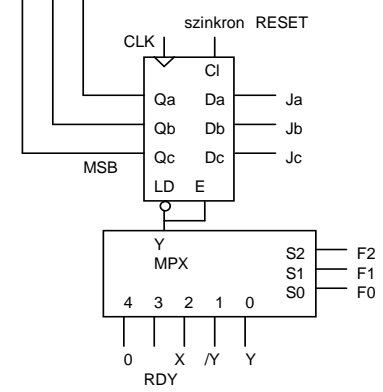
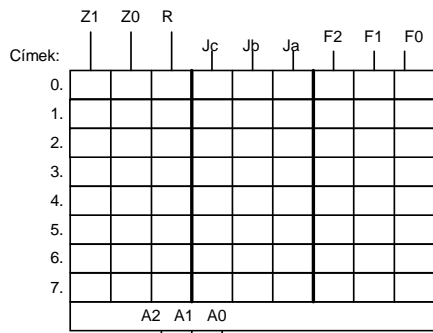


Az idődiagram egy számlál – tölt – számlál sorozatot mutat.



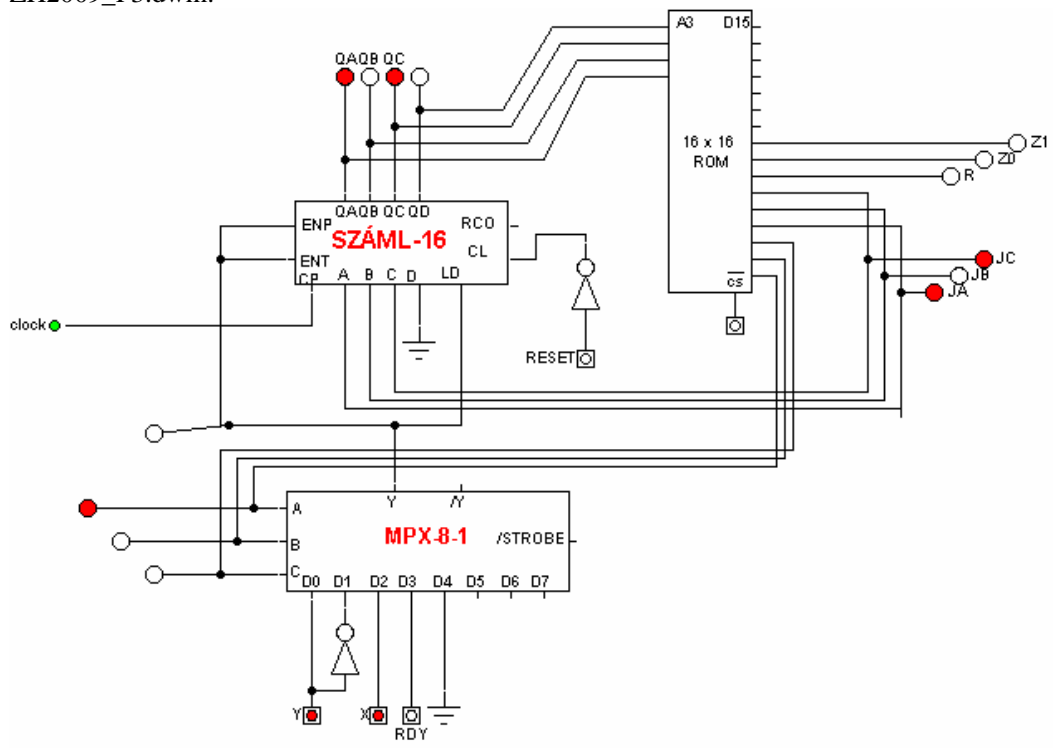
**F3. a.** Adott egy számlálós címképzésű mikroprogramozott vezérlő. A folyamatábra alapján töltsse ki a mikroprogram ROM tartalmát! (14p)



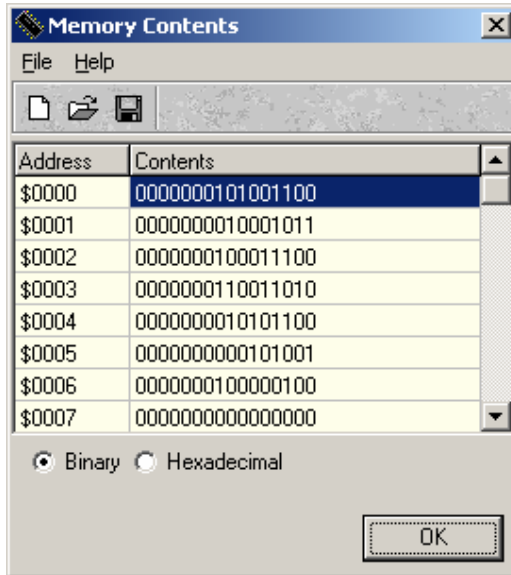


A vezérlő kapcsolását felépítettük DigitalWorks-ben:

ZH2009\_F3.dwm:



Ehhez a mikroprogram ROM tartalma



Fenti ROM képen a szavak jobboldali kilenc bitje a példa mikroutasítása, pontosan a példa szerinti sorrendben (elől a három vezérlő bit, majd a Jc-Ja címbitek, végül az F2-F0 feltétel kiválasztó bitek.

A következő idődiagramon látható, hogy RESET-tel elindított vezérlő az 1-es ütemben a RDY jelre vár, ha ez megjön, akkor (mivel X=1) továbbhalad az 5-ös ütemig, ahol egyelőre reménytelenül várakozik az Y=0 jelre.

F3\_idodiagram

