Megoldás

Digitális technika II. (vimia111)
3. gyakorlat: ROM, RAM. Tervezés adatstruktúra-vezérlés szétválasztással, vezérlőegység generációk

Elméleti anyag:

- ROM, RAM és használatuk egyszerű feladatokban
- Adatstruktúra – vezérlő szétválasztás, vezérlő jelek (engedélyező és működtető) és feltétel jelek
- Egyfázisú és kétfázisú órajelezés
- Random logika
- Fázisregiszteres vezérlő felépítése (lép-ugrik típusú)
    - Számláló (szinkron tölthető, engedélyezhető)
    - Feltétel MPX, fázisdekódoló
    - Ugrási címek előállítása
    - Engedélyező és működtető vezérlő jelek előállítása
- A fázisregiszeres vezérlő módosításai: lép-vár, lép-vár-ugrik típusú
- Mikroprogramozott vezérlő általános felépítése, a mikroutasítás kialakítása, a különböző mezők (műveleti-rész, feltétel-rész, cím-rész) szerepe
- A műveleti rész kódolásának módszerei: horizontális, vertikális, kevert
- A következő címképzés módszerei: lép-ugrik, két cím a mikroutasításban, feltétel bemásolása a címbe
- Adott folyamatábrából a mikroprogram készítésének alapjai
- Vezérlők használata egyszerű feladatokban

Irodalom:

Benesóczky Zoltán: Digitális tervezés funkcionális elemekkel és mikroprocesszorokkal, egyetemi tankönyv, MK55033,.34-45., 90-102. old.

Gyakorló példák:

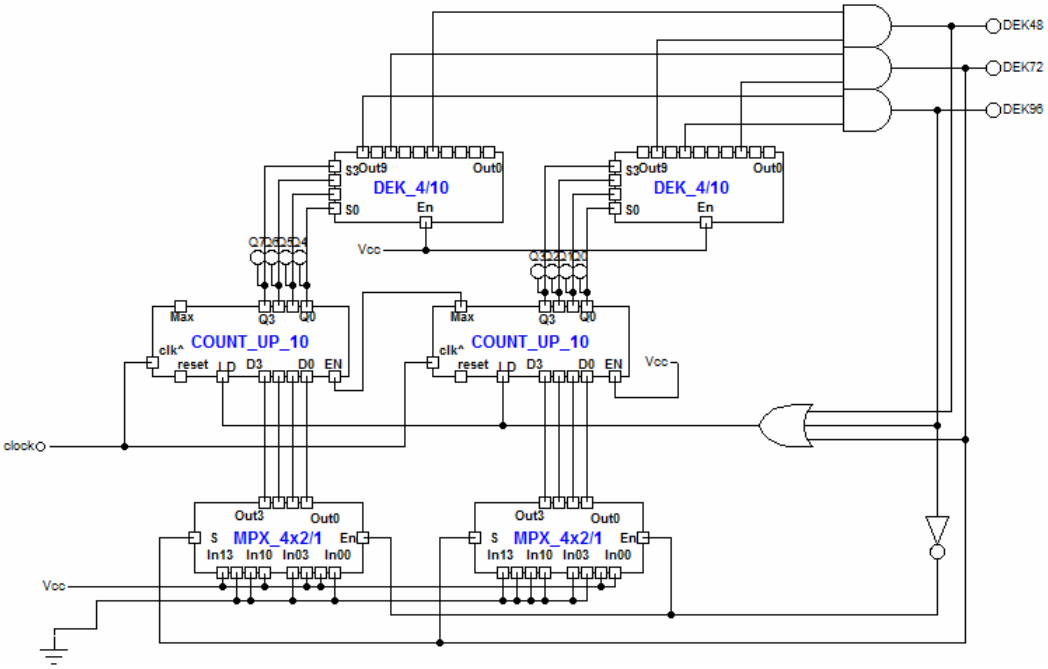
3.1. Készítsen kétdekádos decimális számlálót, amely a következő ciklusban számol: 0,1,2,3,…48,63,64…72,90,91…96, és innen kezdődik elölről.

A feladatban a számláló általában felfelé számlál, kivéve

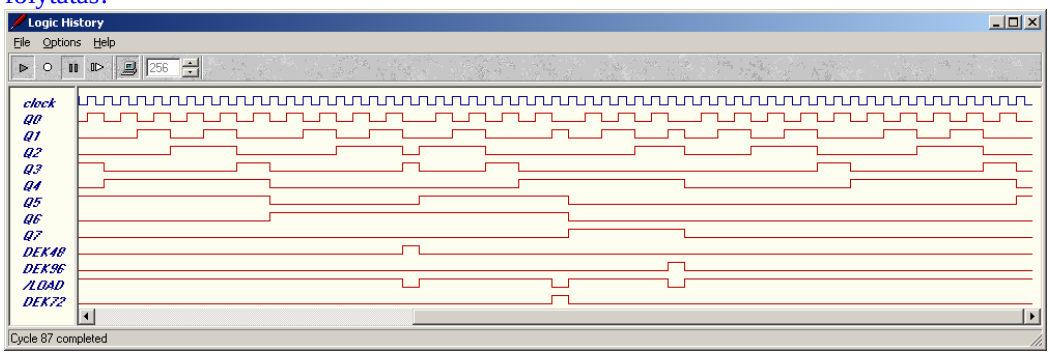
- A 48-as után a 63-mas,
- A 72-es után a 90-es és
- A 96 után a 0-s következik.

Ehhez egyirányú, párhuzamosan betölthető 100-as számláló kell, amit a DW készletből két db. COUNT_UP_100-es egység kaszkádosításával építünk fel.
Dekódolni kell a 48, 72, 96 állapotokat és gondoskodni kell a soron következő állapot párhuzamos beírásáról. Ezt valósítja meg az alábbi kapcsolás.

A számláló három keresett állapotát két 4-ből 10-es dekóder kimeneteinek ÉS kapcsolata állítja elő.
A betöltésnél „szerencsénk” van, a három betöltendő érték közül az egyik 0, ezért elegendő 2/1-es multiplexert használni, a 0át úgy töltünk be, hogy a multiplexert nem engedélyezzük.



Az alábbi idődiagramon jól látszanak az állapot-dekódolások és az utána való helyes folytatás!

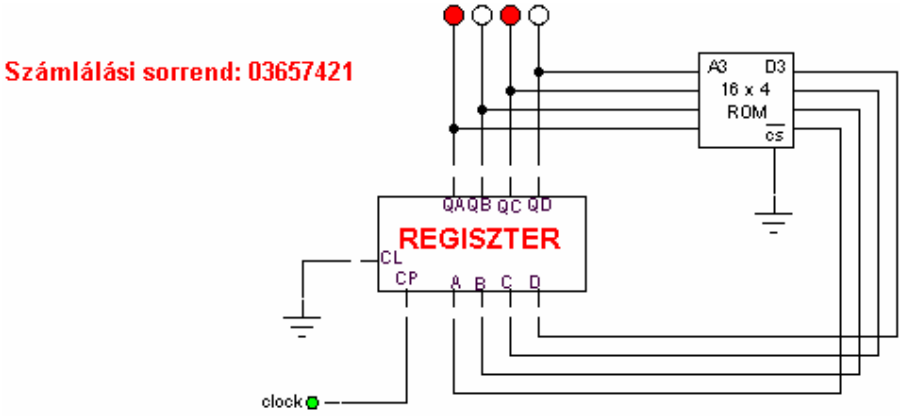


A fenti kapcsolás megtalálható: **cseles_szamlalo.dwm:**

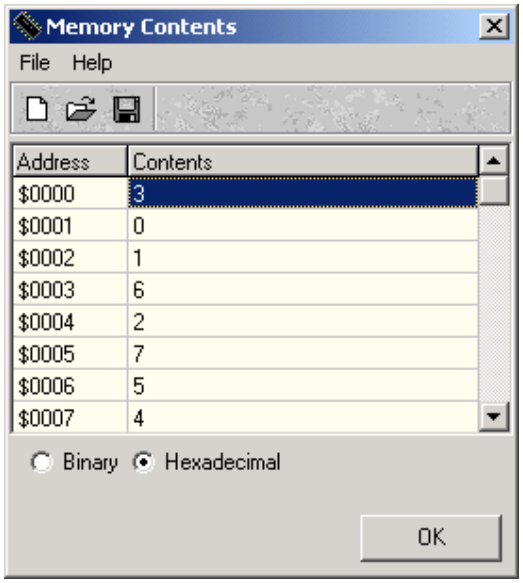
3.2 Készítse el számlálóból és ROM-ból a múlt félévi HF-et (a digit kód sorrendjében számláló számláló)

a: A számlálási sorrend közvetlenül az állapotkódban
tavalyi_szamlalo1.dwm:

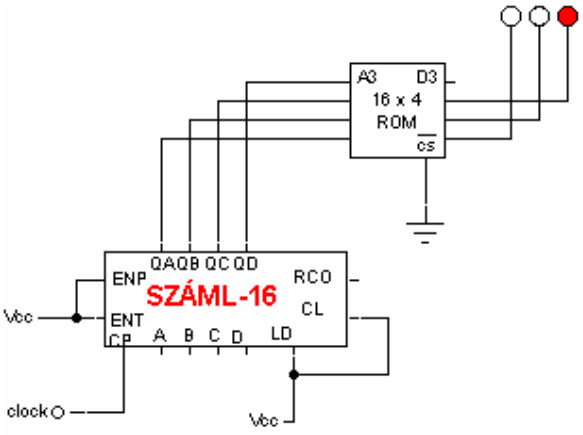
Számlálási sorrend: 03657421

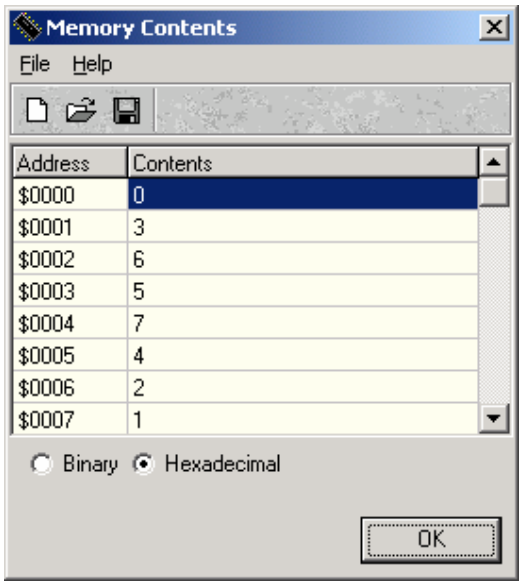


A kívánt sorrendhez a ROM tartalma



b. Bináris számláló kimeneti átkódolóval
tavalyi_szamlalo2.dwm:





3.3. Készítse el a fenti számlálót fázisregiszteres vezérlővel! Egy kapcsoló állásától függően vagy a normál számlálási sorrend, vagy a digit kód szerinti sorrend legyen!

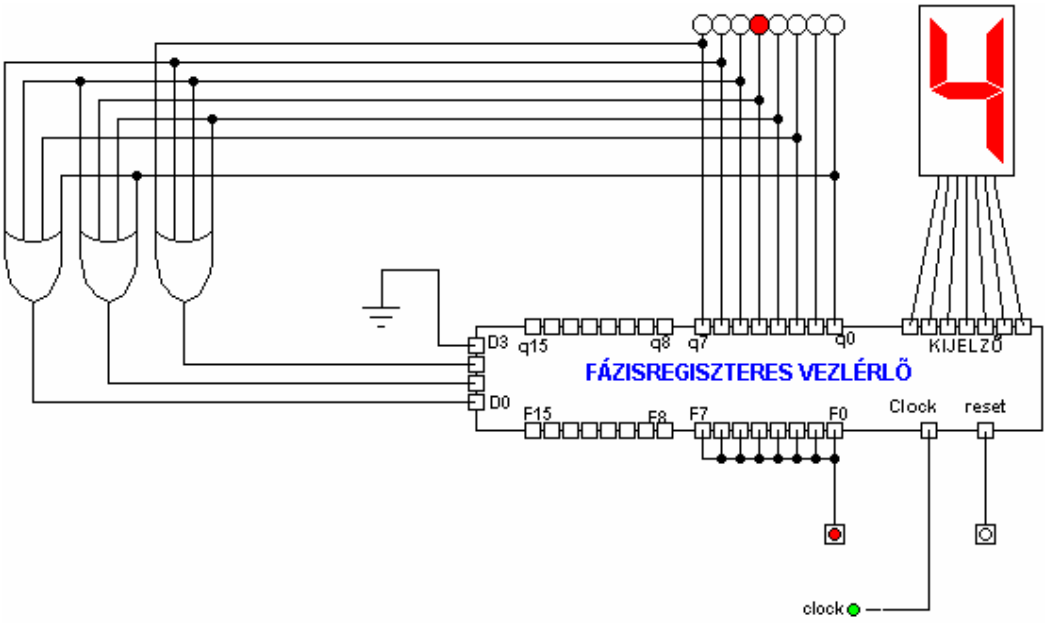
Megoldás:
A bemeneti feltétel 0 értéke mellett a szükséges ugrások megegyeznek a legelső ROM-tartalommal):

állapot	CBA
q0	011
q1	000
q2	001
q3	110
q4	010
q5	111
q6	101
q7	100

Ebből:

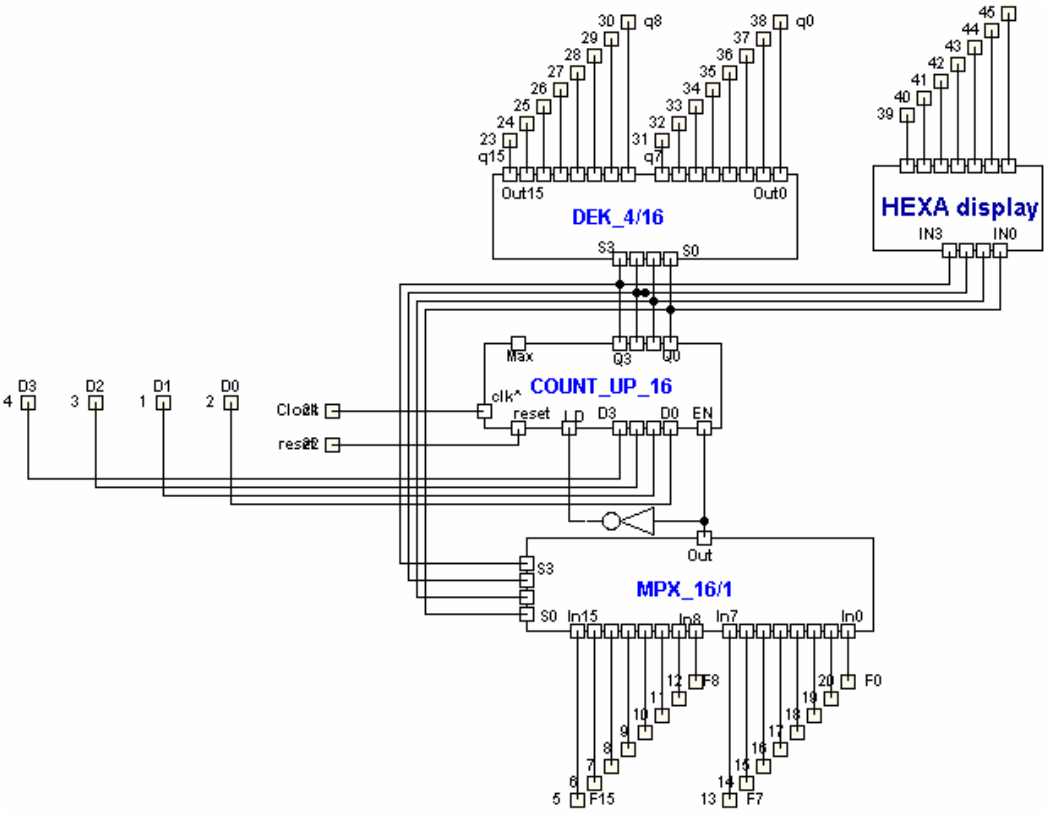
C = q3+q5+q6+q7
B = q0+q3+q4+q5
A = q0+q2+q5+q6

Fazisregiszteres_szamlalo.dwm:



Megjegyzés: ez a megoldás a „normál” számlálási sorrendben valójában 0-tól 8-ig (és nem 7-ig) számlál! Ha nem 16, hanem 8 fázisú vezérlőt használtunk volna, akkor működne a kiírásnak megfelelően.

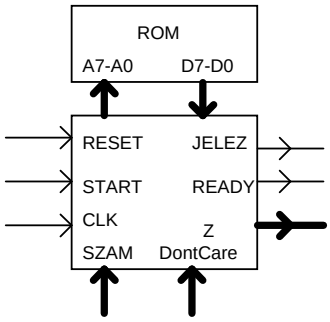
Maga a fázisregiszteres vezérlő makró:
Fazisreg_vezerlo_macro.dwm:



Gyakorlásként csináljunk egyes állapotokban vezérlő (ENG és MÜK) jeleket!

3.4. Egy 2009-es ZH példa.

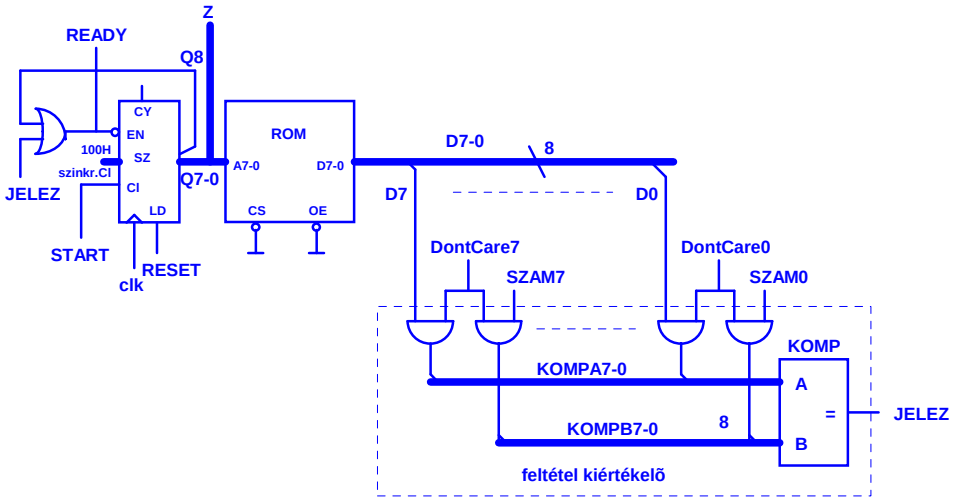
Tervezze meg egy olyan egység részletes funkcionális blokkvázlatát, amely megkeresi egy 256 byte-os ROM-ban levő, az alább megadott feltételeket kielégítő ***első*** adatot és kiadja ennek címét a kimenetén. Ha talált ilyen számot, azt a JELEZ kimeneten adott 1-gyel mutatja. Ha végzett a feladattal, azt a READY kimenetén adott 1-gyel 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 bementre 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.



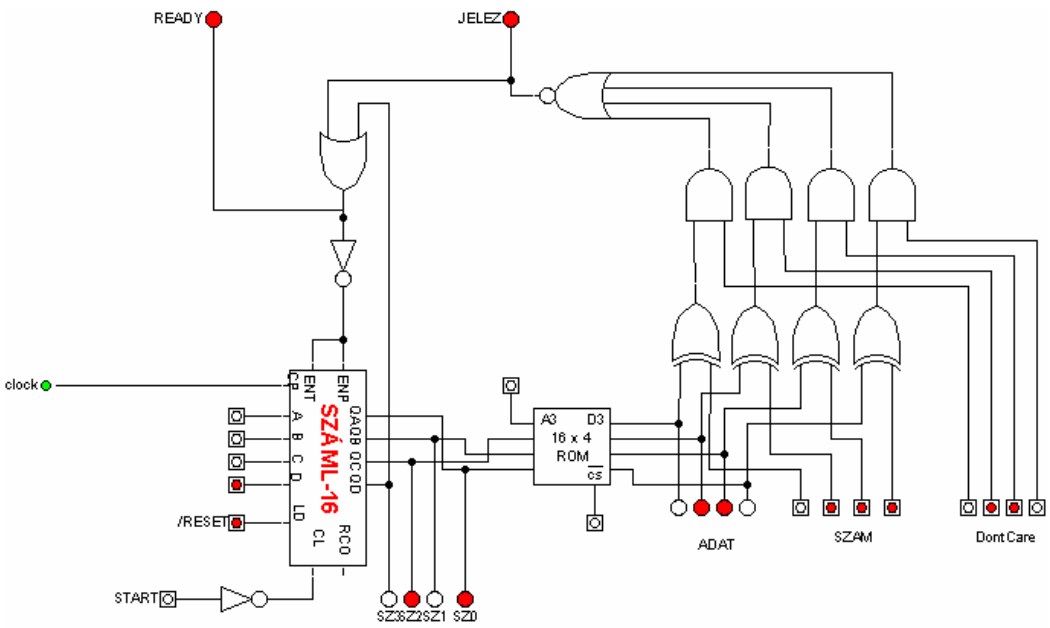
Külön lapon rajzolja le a megoldását! A feladatot kétféleképpen is megoldhatja:

- 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)

Megoldás:
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:
Kereses_ROMban.dwm:



Az alapelv:

- 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.

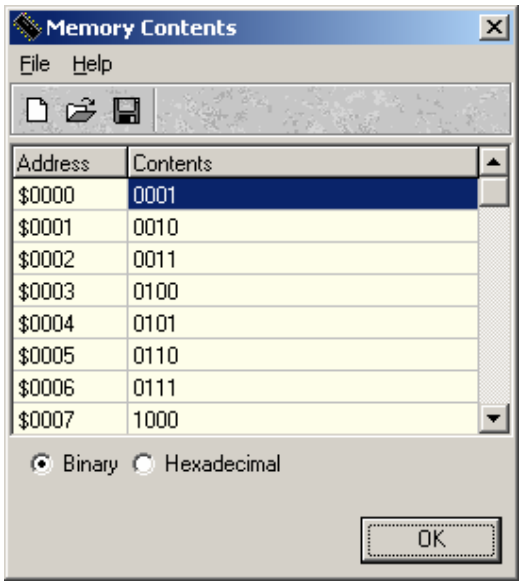
Megjegyzés: Ezt a feladatot a „régi könyvtárral” dolgoztuk ki, az ott szereplő SZÁML-16 tulajdonságai:

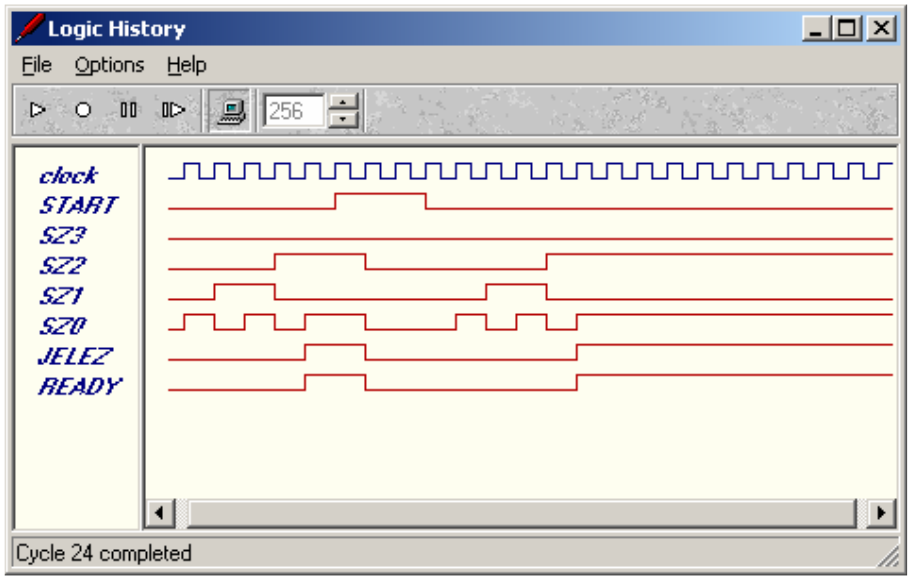
- aszinkron aktív low clear
- szinkron aktív high load
- az engedélyezés aktív high

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

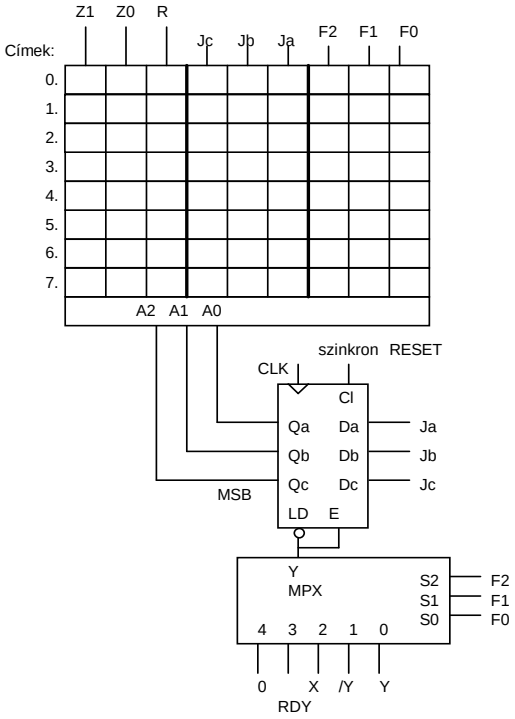
- Az ADAT-ot és a SZAM-ot bitenként mod2-zük (az egyezést a 0 mod2 kimenet jelenti,
- A DontCare maszkkal bitenként És kapcsolatba hozva ezeket, az ÉS kapu kimenetek 0-val jelzik, ha az adott sorszámú biteket nem kell figyelembe venni, VAGY ha egyformák.
- Az összes ilyen kimenet VAGY kapcsolata akkor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

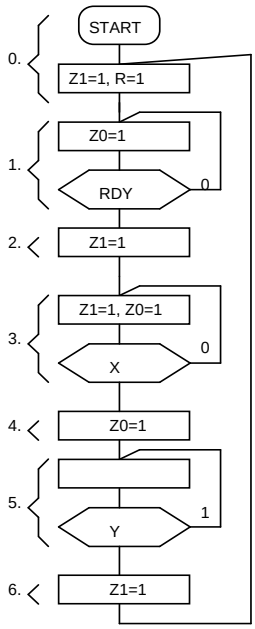
A fenti kapcsolást az alábbi ROM tartalommal teszteltük és az áramkör nagyon helyesen az 5-ös ROM címen találta meg az első keresett adatot. Itt aztán meg is állt, amint a mellékelt időiagram is mutatja.



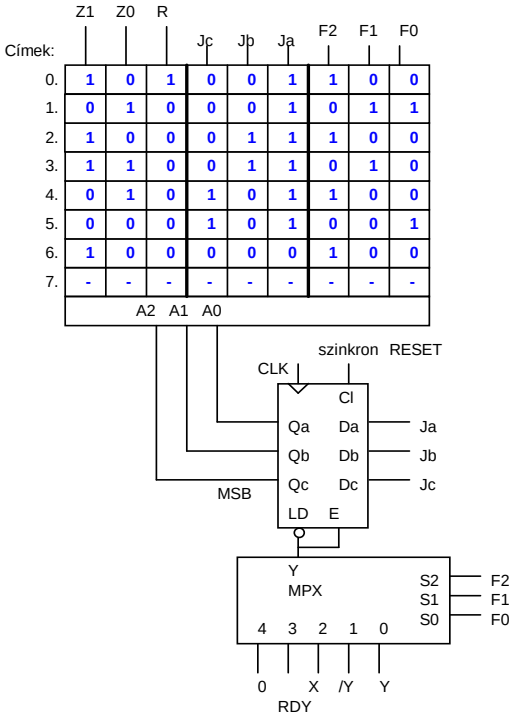


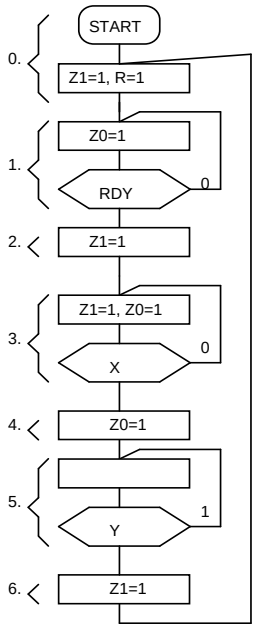
4.4. Egy 2010-es ZH példa:
Adott egy számlálós címképzésű mikroprogramozott vezérlő. A folyamatábra alapján töltse ki a mikroprogram ROM tartalmát!



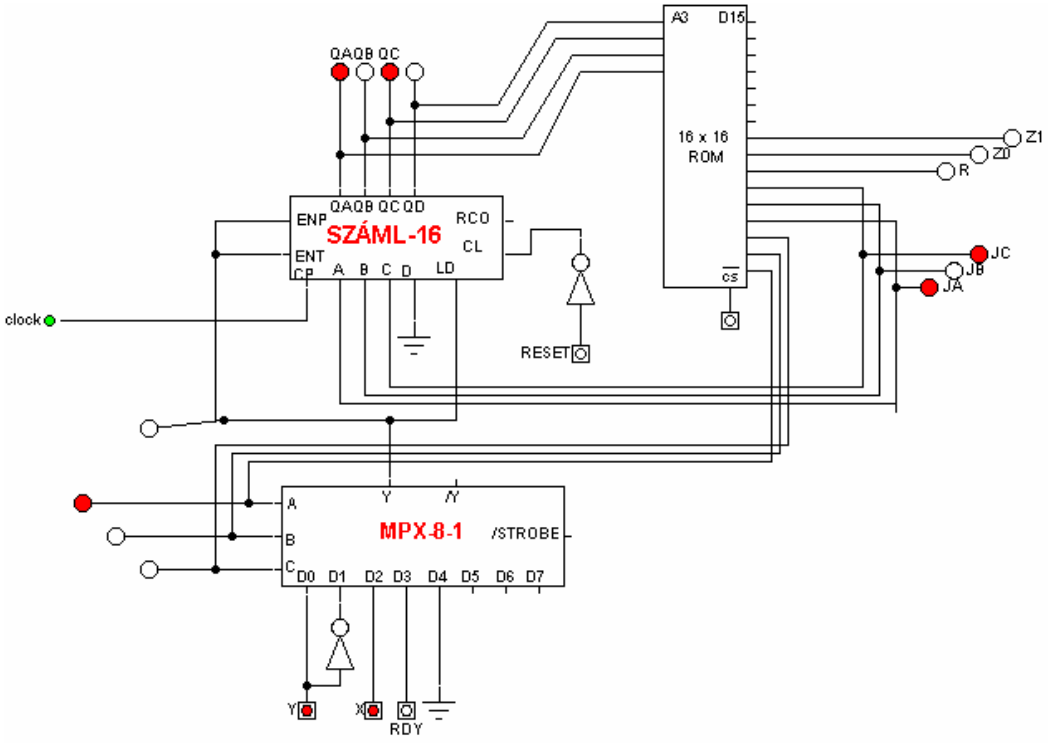


Megoldás:

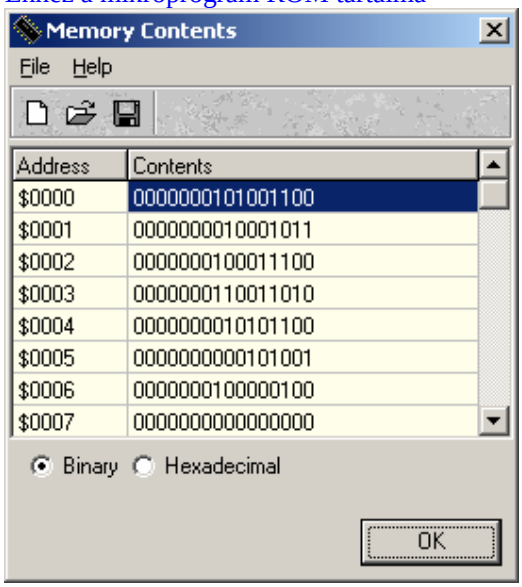




A vezérlő kapcsolását felépítettük DigitalWorks-ben:
Mikroprogram_keszites.dwm:

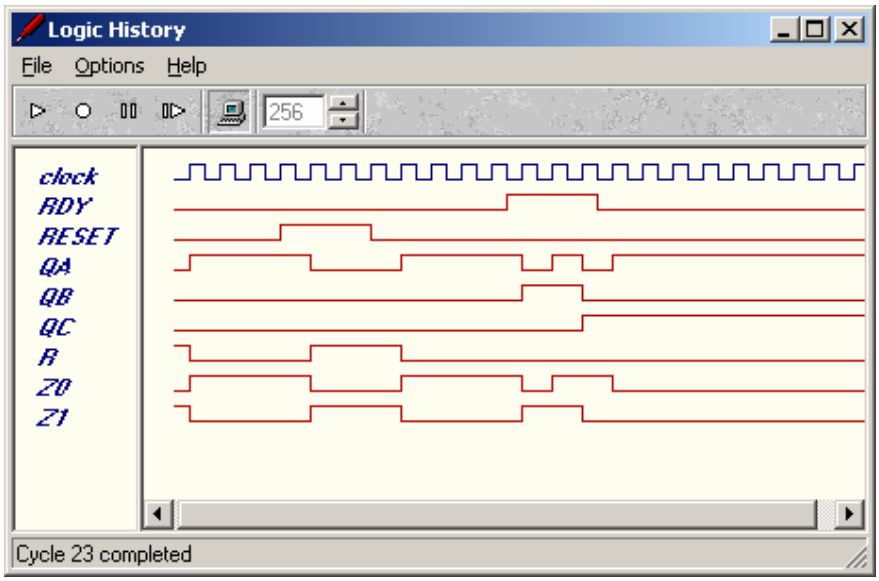


Ehhez a mikroprogram ROM tartalma



Fenti ROM képben 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) továbbszalad az 5-ös ütemig, ahol egyelőre reménytelenül várakozik az Y=0 jelre.
F3_idődiagram



Kövessük az adott folyamatábra szerinti működést!
Változtassunk kis részleteket a folyamatábrán, mit kell ehhez módosítani az idődiagramon?