

SZERKESZTŐBIZOTTSÁG

BHG

ORION

TERTA

Berecz Frigyes
Bernhardt Richárd
Eisler Péter
Dr. Gosztony Géza
Honti Ottó
Klug Miklós
Tölgyesi László

Jakubik Béla
Baracs Sándor
Csernch János
Froemel Károly
Sass Károly
Szabó Károly

Bánsághi Pál
Baján Tibor
Benedek Elek
Egerszegi Béla
Hutter Mihály

MŰSZAKI
KÖZLEMÉNYEK

Aszinkron szekvenciális hálózat tervezése
visszacsatolt memória felhasználásával

HALMI GÁBOR
TERTA

Visszacsatolt memória, mint R-S tároló

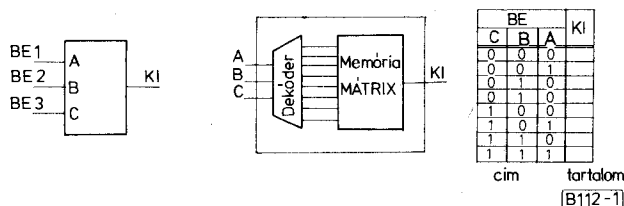
Vizsgáljuk meg az 1. ábrán látható 8×1 bites memória működését. A memória kimenetén mindig az a bit-érték (memóriatartalom) jelenik meg, amit a bemeneten levő 3 bites, binárisan kódolt cím kijelöl. Attól függően logikai 1 vagy 0, hogy az adott helyre előzőleg mi lett beírva. Ha most ennek a memóriának a kimenetét az egyik bemeneti címpontra kötjük a 2A. ábra szerint, akkor a szabadon rendelkezésünkre álló 2 bemeneti vezetékkel már csak négy memóriaterületet tudunk megcímezni. Hogy ez a memória 8 rekesze közül melyik négy lesz, azt a visszacsatolt kimeneti bit értéke fogja meghatározni. Ugyanazon bemeneti kombinációhoz attól függően fog különböző kimeneti érték tartozni, hogy melyik tárfél van a visszacsatolt kimenet által érvényesítve. Az egyik tárfélről a másikra a kimeneti bit megváltoztatásával lehet áttérni. Hogy ez milyen bemeneti kombináció hatására történik, az a memóriatartalomtól függ.

A könnyebb érthetőség kedvéért tételezzük fel, hogy a memória tartalma az alábbi. (2B. ábra):
Látható, hogy ha a bemeneti bitek a 0; 1 kombinációt veszik fel, a kimenet 1 értékű lesz, ami a másik memóriafél részére való áttérést jelent. Mindaddig ezen a memóriafélen maradunk, amíg a bemenetek az 1; 0 kombinációt nem veszik fel.

Működését tekintve ez megfelel a hagyományos kapuáramkörökből kialakított 3. ábra szerinti R-S tárolónak.

Vagy ezt átrajzolva a 4. ábra szerint, a kimenet logikai függvénye:

$$K_i = ((\overline{A \cdot B}) \cdot C) \cdot (A \cdot \overline{B}) = \overline{A}B + AC + \overline{B}C$$

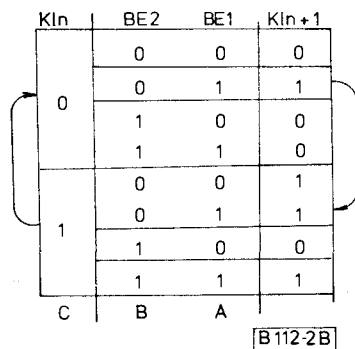
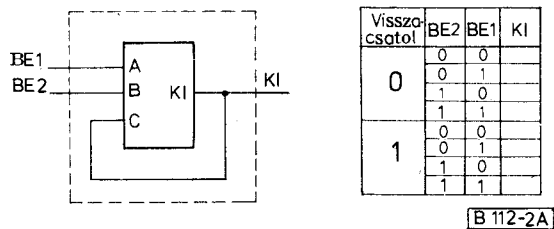


1. ábra. 2³ × 1 bites memória jelölése és blokkvázlata

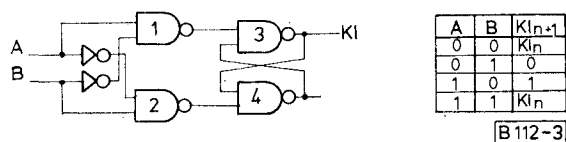
ami pontosan megfelel az előbbieken felírt memóriatáblázat tartalmának.

Könnyen belátható, hogy a memória tartalmának megváltoztatásával az adott határok közt tetszőleges billenési feltételeknek eleget tevő tároló alakítható ki.

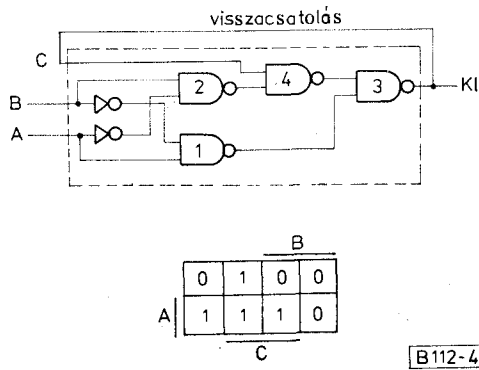
Ha a különböző bemeneti kombinációkhoz tartozó memóriállapotokat tekintjük, a példánkban szereplő visszacsatolt memóriának az 5. ábrán látható állapotdiagram szerint, 6 stabil és 2 átmeneti állapota van. Az az állapot lesz átmeneti, ahol a visszacsatolt bit bemeneti értéke nem egyezik meg az ezen cím-



2. ábra. Visszacsatolt 2³ × 1 bites memória



3. ábra. Kapuáramkörökből felépített R-S tároló



4. ábra. Visszacsatolt kombinációs áramkör, mint R—S tároló

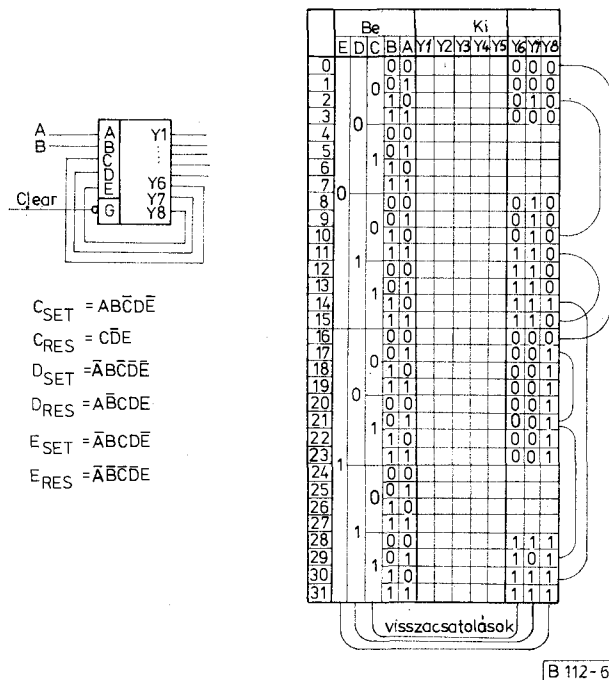
hez tartozó kimeneti memóriatartalom bitértékével. Az átmeneti állapotból a memória mindig a másik tárfél azonos bemeneti kombinációjához tartozó állapotba megy. Ez természetesen stabil állapotot kell legyen, különben rezgékeltőt kapunk.

Többszörös visszacsatolású memória

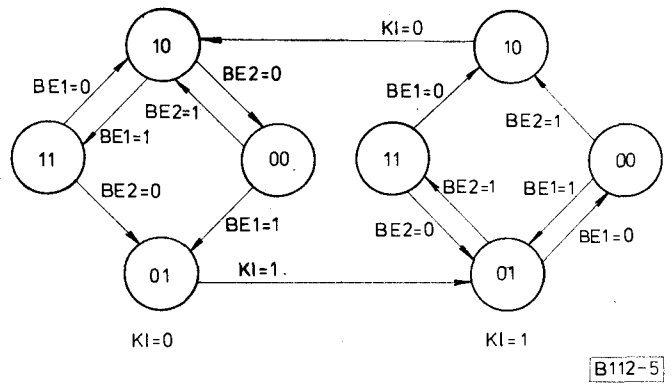
Az eddigiekben láttuk, hogy ha a memória kimenetéről egy bitet visszavezettünk a bemenetre, ezzel a tárolási területet két féltre osztottuk, amelyek közt az átmenetet a visszacsatolt bit biztosítja. Amennyiben több (n) bitet csatolunk vissza a memóriaterület is több, 2^n részre lesz felosztva. Ilyenkor az átmenet az egyes memórianegyedek, -nyolcadok stb. között történik.

Vizsgáljuk meg egy háromszorosan visszacsatolt, $2^3 \times 8$ bites memória viselkedését a 6. ábra szerint.

A könnyebb áttekinthetőség kedvéért csak a visszacsatolt kimeneti bit értékeket (K6, K7, K8) tün-



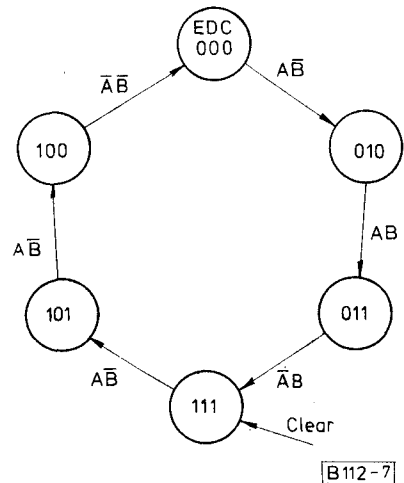
5. ábra. Visszacsatolt memória állapotdiagramja



6. ábra. Háromszorosan visszacsatolt $2^3 \times 8$ bites memória

tettük fel az igazságtáblában. A többi kimeneti bit (KI—K5) a további feladatokhoz szükséges vezérlő jelek, logikai függvények realizálására használható fel. Az átmeneti állapotokat, illetve az átmeneteket az igazságtábla melletti nyilazással jelöltük.

A választott példában a C, D, E bemenetekre való visszacsatolással megvalósított tárolóhármast a 7. ábrán látható állapotdiagram szerinti szekvenciális működést realizál.



7. ábra. Háromszorosan visszacsatolt memória állapotdiagramja

Látható, hogy a tárolók az 110 és 001 állapotokat soha nem veszik fel, az 101 állapot pedig csak átmeneti időre — a D tároló törlési idejére — áll elő. Azaz nem szükséges a memória összes, lehetséges tárolóterületét (állapotát) kihasználni, hanem csak annyit, amennyi az adott feladat megoldásához elengedhetetlen.

A ki nem használt állapotokhoz (határozatlan állapot) tartozó kimeneti bit értékek tetszőlegesek lehetnek, mégis célszerű ezeknek az értékét is meghatározni. A hagyományos logikai tervezéseknél ezeket minimalizálási célra jól ki lehetett használni, most azonban a minimalizálás nem szempont. A határozatlan állapotok kimeneti bitjeit pl. úgy célszerű megválasztani, hogy ha a tároló valamilyen zavar,

vagy téves működés folytán mégis ebbe az állapotba jutna, kerüljön ki ebből egy meghatározott, „értelmezett” állapotba menve. Azaz a határozatlan állapotok legyenek átmeneti állapotok. Más szempontok alapján lehet azt a megoldást is választani, hogy a határozatlan állapotból a rendszer ne kerülhessen ki — a kimeneti kombináció mindig önmagára mutat vissza — és ebben az esetben hibajelzés adható.

Mire jó ez?

Vizsgáljuk meg, hogy hogyan lehet egy ilyen áramkört tervezni, felhasználni.

A tervezést két úton is elkezdhetjük; vagy idő-(ütem)diagramból, vagy állapotdiagramból kiindulva.

Kiindulás idődiagramból

Abban az esetben célszerű ezt a módszert választani, ha a feladat jellege olyan, hogy idődiagrammal szemléletesen, jól áttekinthetően le tudjuk írni, illetve ha a feladat idődiagram formájában fogalmazódik meg.

a) Ilyenkor először a megkülönböztetni kívánt, azonos bemeneti kombinációhoz tartozó állapotokat jelöljük meg.

b) Megkeressük az ezen állapotok egyértelmű megkülönböztetéséhez minimálisan szükséges segédváltozók (R—S tárolók) legkedvezőbb billentési feltételeit. (Ez a legnehezebb rész a feladatmegoldásnak.) Minden egyes segédváltozó R—S tárolót egy visszacsatolással tudunk realizálni.

c) A segédváltozók és a bemeneti változók, valamint a szükséges kimeneti jelek számából meghatározzuk a szükséges tároló méretét (cím x tartalom), és kiválasztjuk a megfelelő IC típust, illetve megtervezük a memóriát.

d) A megmaradó „szabad” memóriaterületet megpróbáljuk további funkciókhoz, szolgáltatásokra felhasználni.

e) A segédváltozók billentési feltételei az átmeneti állapotok. Ezeket kikeresve, kitöltjük a memóriatartalom táblázatát. Kisméretű memóriák (max. 32×8 bit) esetén célszerű a tároló tartalmát bináris kódban (bitenként) táblázatba foglalni, míg nagyobb memóriaméret esetén ez áttekinthetetlenül nagy táblázat eredményezne, így ebben az esetben az oktális vagy hexadecimális kódú táblázatmegadás célszerűbb. Ez utóbbi megoldás viszont a szemléleteség rovására megy, azaz nagyobb figyelmet kíván.

Kiindulás állapotdiagramból

Ha a feladat jellege olyan, hogy állapotdiagramban könnyű megfogalmazni, vagy az idődiagrammal adott feladatot állapotdiagramra egyszerűen vissza lehet vezetni, akkor ezt a módszert célszerű alkalmazni.

A) Felrajzoljuk a rendszer állapotdiagramját, az állapottárolókat a visszacsatolással létrehozott tárolók jelentik.

B) A bemeneti, a visszacsatolt és a kimeneti változók száma alapján megtervezük a szükséges memóriát.

C) Az egyes állapotok közti átmenethez tartozó bemeneti kombinációk (a bemenő változók adott kombinációja) kiválasztásával meghatározzuk a memória átmeneti állapotait.

D) A megmaradó „szabad” memóriaterületet megpróbáljuk további szolgáltatásokra felhasználni.

E) Kikötjük a memóriatáblázatot.

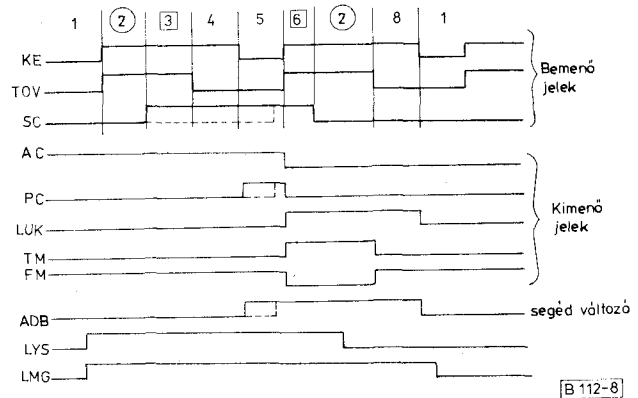
Példaként oldjuk meg a következő feladatot:

Tervezzünk illesztő-vezérlő áramkört, amely az EP—35 típusú lyukszalaglyukasztót egy BSI logikájú interfészhez illeszti.

A lyukszalaglyukasztó az általa adott KE és TOV szinkronjelekkel egy időben kívánja a kódmágnesek (KE), illetve a továbbító mágnes (TOV) gerjesztését.

A feladat jellegéből adódóan most az idődiagramos tervezés a célszerű.

A megadani kívánt idődiagram a 8. ábrán látható.



8. ábra. A példa szerinti idődiagram

A kialakítandó jelek:

- LUK kódmágnesek vezérlése
- TM továbbító mágnes vezérlése
- FM fékmágnes vezérlése
- AC adatkérés
- PC puffertároló beírás

Bejövő vezérlő jelek:

- KE kódmágnes gerjesztés
- TOV továbbító mágnes gerjesztés
- SC bejövő adat érvényes

a) A megkülönböztetni kívánt állapotok (azonos bemeneti kombinációhoz tartozó állapotok) a 2. és 7. illetve a 3. és 6.

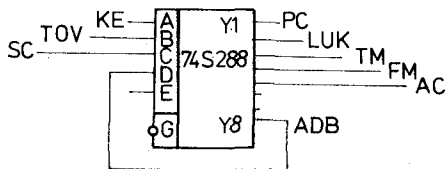
b) Ezek megkülönböztetésére az ADB-vel jelzett (adat a puffertároló) segédváltozó szolgálhat. A billentési feltételei:

$$ADB_{SET} = \overline{KE} \cdot \overline{TOV} \cdot SC \cdot \overline{ADB}$$

$$ADB_{RESET} = \overline{KE} \cdot \overline{SC} \cdot ADB$$

c) A három bemeneti változó + 1 visszacsatolás és a kimeneti változók számából (5 + 1) a minimálisra szükséges memóriaméret $2^4 \times 6$ bites. A gyakorlatban létező integrált áramköröket figyelembe véve a $2^5 \times 8$ bites memória (SN74S288) a legalkalmasabb erre a célra. Az ezzel megvalósított vezérlés a 9. ábrán látható.

d) Hogy lehet a megmaradó szabad memóriaterületet célszerűen felhasználni?



B112-9

9. ábra. A vezérlést biztosító memória

Nagyon kellemetlen, ha a lyukasztó motorja állandóan gerjesztve van. Ezért, amikor nem akarunk lyukasztani, a motorról a 220 V-os feszültséget kapcsoljuk le. A LYS (lyukasztó start) jel magas szintje alatt adjunk a motornak gerjesztést. De ha a LYS pont egy lyukasztási ciklus alatt szűnik meg, akkor a motorgerjesztést a lyukasztás befejezéséig tartjuk fent.

Ezért a memória szabad bemenetére vezessük a LYS-t, és a lyukasztó motor gerjesztését biztosító jelet LMG az egyik szabad kimeneten alakítsuk ki. Ez egy tranzistoros erősítőn keresztül a 220 V-ot kapcsoló jelfogót vezérelheti.

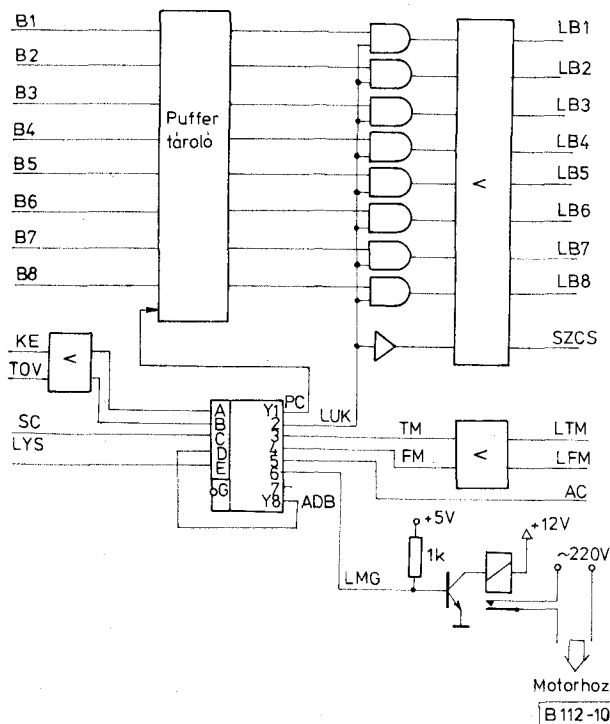
$LMG = LYS + ADB$, és a LYS jelet az ADB tároló billentési feltételénél is figyelembe kell venni.

Így a megvalósított teljes kapcsolás illetve illesztés a 10. ábrán látható.

e) Az átmeneti állapotok és a visszacsatolt (8.) bit hozzátartozó értékei:

$$\overline{KE} \cdot \overline{TOV} \cdot \overline{SC} \cdot \overline{ADB} \cdot LYS \Rightarrow 1$$

$$\overline{KE} \cdot \overline{TOV} \cdot \overline{SC} \cdot ADB \Rightarrow 0$$



B112-10

10. ábra. Az EP-35 illesztését biztosító teljes áramkör elvi kapcsolási rajza

LYS	ADB	SC	TOV	KE	ADB	LMG	AC	FM	TM	LUK	PC	
E	D	C	B	A	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1
0	0	0	0	0	0	0	1	1	0	0	0	0
0	0	1	0	0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1	1	0	0	0	0
0	1	0	0	1	0	0	0	1	1	0	0	0
0	1	1	0	0	0	0	1	1	0	0	0	0
0	1	1	0	1	0	0	0	1	1	0	0	0
0	1	1	1	0	0	0	1	1	0	0	0	0
0	1	1	1	1	0	0	1	1	0	0	0	0
1	0	0	0	0	0	1	1	0	1	1	0	0
1	0	0	1	0	0	1	1	0	1	1	0	0
1	0	1	0	0	0	1	1	0	1	1	0	0
1	0	1	0	1	0	1	1	0	1	1	0	0
1	0	1	1	0	0	1	1	0	1	1	0	0
1	0	1	1	1	0	1	1	0	1	1	0	0
1	1	0	0	0	0	1	1	0	0	0	0	0
1	1	0	0	1	0	1	1	0	0	0	0	0
1	1	0	1	0	0	1	1	0	0	0	0	0
1	1	0	1	1	0	1	1	0	0	0	0	0
1	1	1	0	0	0	1	1	0	0	0	0	0
1	1	1	0	1	0	1	1	0	0	0	0	0
1	1	1	0	1	1	0	0	1	1	0	0	1
1	1	1	1	0	0	1	1	0	0	1	1	1
1	1	1	1	1	0	0	1	1	1	1	1	1

B112-11

11. ábra. A vezérlést biztosító memória tartalma

A kimeneti jelek logikai függvényei:

$$LUK = ADB \cdot KE$$

$$TM = ADB \cdot TOV \cdot KE$$

$$FM = ADB \cdot TOV \cdot \overline{KE}$$

$$PC = ADB \cdot SC$$

$$AC = ADB \cdot KE \cdot LYS$$

$$LMG = LYS + ADB$$

Ebből a memóriatartalom könnyen kitölthető a 11. ábra szerint.

Mivel ugyanazzal a jellel tranzistoros áramkört meghajtani és TTL kapukat vezérelni nem célszerű, a 10. ábrán „->” jellel jelölt meghajtó áramkör helyett a memória még szabad kimenetén a LUK jelet duplikáljuk és a szinkroncsatornát ezzel vezérelhetjük.

Példa az állapotdiagramos tervezésre

Készítsünk háromfázisú óragenerátort, amely a bemenő Q órajelből a 12. ábra szerinti jeleket állítja elő.

A) A megkülönböztetendő állapotok a 12. ábrán 0-5 számozással vannak jelölve, az állapotdiagram a 13. ábrán látható.

Az állapotkódolás eredménye az állapotdiagramban látható, az egyes állapotokhoz tartozó kimeneti események nincsenek feltüntetve

- több integrált áramköri tokból felépített memória esetén a chip select (kiválasztó jel) nem engedélyező szintje esetén az adott tokkal megvalósított tároló törlődik. Ilyenkor célszerű a visszacsatolással a tokok közti váltást biztosítani, vagy más áramköri megoldással biztosítani, hogy a lényeges chipek mindig érvényesítve legyenek és pl. a kimenetek kapuzásával megoldani a kiválasztást,
- előfordulhat, hogy nehéz a részfeladatot úgy megfogalmazni, hogy egyértelműen kitűnjék, hogy milyen megoldást célszerű választani.

Ezen nehézségek azonban nem komolyabbak, mint

ami bármely más, újszerű tervezési elv elsődleges alkalmazásánál jelentkeznek. Megfelelő tervezői gyakorlattal ezek a problémák eltűnnek.

I R O D A L O M

- [1] Dr. *Szittyá O.*: Logikai kapcsolástan (Tankönyvkiadó Bp. 1973.).
- [2] The Semiconductor Memory Data Book. Texas Instruments Incorporated.
- [3] *Madarász L.*: Általánosan használható, programozható áramkörök AUTOMATIZÁLÁS 1979/9.
- [4] *E. Stepanek*: State-Diagramme einfacher logischer Schaltungen Nachrichtentechnik Elektronik 29/1979/H7.