

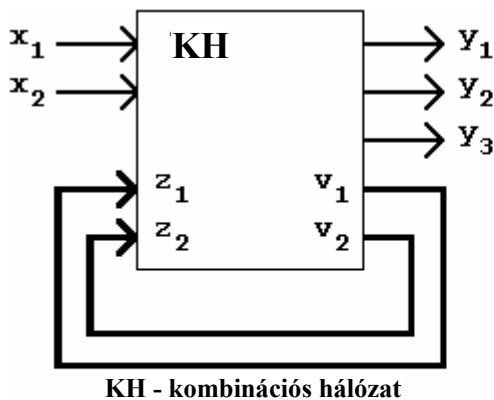
Szekvenciális hálózatok és automaták

Szekvenciális hálózatok a kombinációs hálózatokból jöhetnek létre tárolási tulajdonságok hozzáadásával.

A tárolás megvalósítása történhet

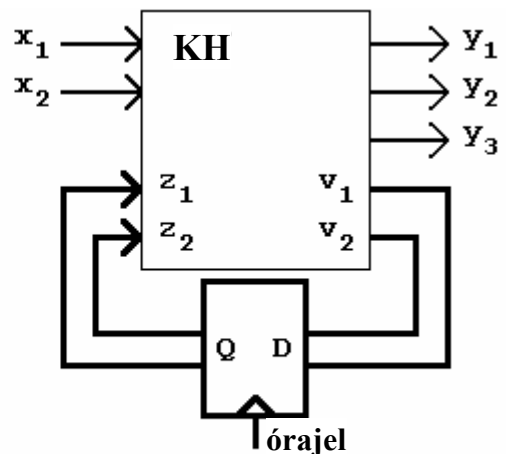
- a kapcsolás logikáját képező kombinációs hálózat kimeneteinek visszacsatolásával a bemeneteire, vagy
- flipflop-ok, regiszterek tároló-képességének kihasználásával.

A flipflop-oknál bevezetett általános ábrázolási módot továbbfejlesztve:



KH - kombinációs hálózat

Aszinkron szekvenciális hálózat



Szinkron szekvenciális hálózat

A hálózat **állapota**: az összes megelőző állapot és a pillanatnyi külső adatbemenetek értékeinek logikai következménye. Leírja a hálózat pillanatnyi helyzetét.

- Stabil állapot az, amelyiket a hálózat csak a külső bemenetek megváltozásának hatására hagy el.
- Átmeneti (köztes) állapotok két, egymásra következő stabil állapot között lépnek fel. A stabil állapotok egymásra következése (szekvenciája) egy folyamat, melyet a külső bemenetek megváltozása vált ki és automatikusan zajlik le.

Az aszinkron hálózatok működésükben általánosabbak és bonyolultabbak mint a szinkron hálózatok. A stabil állapot elérése egyedül a KH belső jelterjedési viszonyaitól függ.

A kombinációs hálózatokkal ellentétben a szekvenciális hálózatoknál nem követelmény az, hogy a mindenkor kimenet logikailag megfeleljen az aktuális bemenetnek, itt a köztes (nem „helyes”) állapotok is - a szekvenciális hálózat természeténél fogva - átmenetileg megengedettek.

Szekvenciális hálózatok és automaták

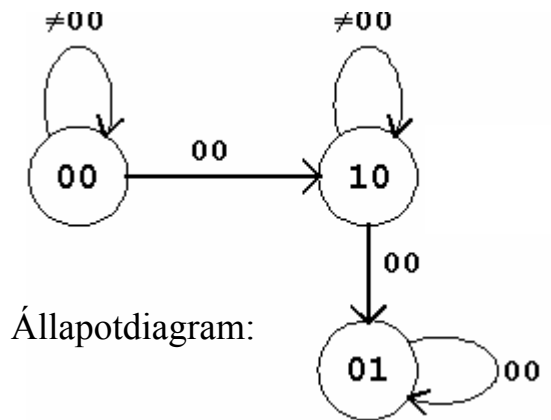
Stabilitás

Egy **aszinkron** szekvenciális hálózat akkor van stabil állapotban, ha az összes visszacsatolásra érvényes: $v_i^+ = z_i$ Igazságtáblázat:

Bemenetek				Kimenetek					Sorrend
x_1	x_2	z_1	z_2	v_1	v_2	y_1	y_2	y_3	
0	0	0	0	1	0	1	1	1	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">1</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">2</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">3</div> </div>
0	0	0	1	0	1	1	0	0	
0	0	1	0	0	1	0	1	1	
:	:	:	:	:	:	:	:	:	
1	1	1	1	0	0	1	1	1	

stabil állapot

- A példában a $v_{1,2} = „0,1”$ állapot stabil, ha a bemenet $x_{1,2} = „00”$
- A példában feltételezzük, hogy a visszacsatolt kimenetek egyidejűleg változnak.



Egy **szinkron** szekvenciális hálózatban a stabilitási feltétel állandóan adva van, tehát minden állapot stabil. A sorrendben következő állapotot mindig a következő órajel-impulzus váltja ki. A kiváltó (hatásos) órajel-élnek viszont csak a kimenetek beállása, stabilizálódása után szabad jönni.

Bemenetek				Kimenetek					Állapot	
x_1	x_2	z_1	z_2	v_1	v_2	y_1	y_2	y_3		
0	0	0	0	1	0	1	1	1		t_i
0	0	0	1	0	1	1	0	0		t_{i+2}
0	0	1	0	0	1	0	1	1		t_{i+1}
:	:	:	:	:	:	:	:	:		

Szekvenciális hálózatok és automaták

Vezérlő hálózatok (Mealy- és Moore-automaták)

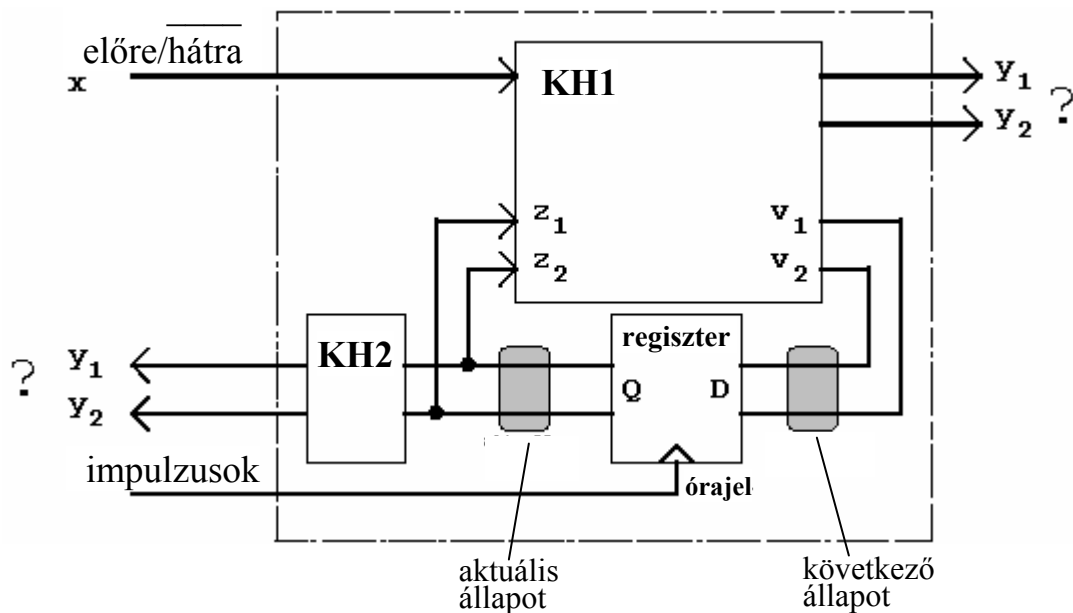
A szinkron hálózatok jelentős csoportját képezik az un. vezérlő hálózatok, automaták (állapot-vezérlés, *finite state machine*).

Jellemzőjük, hogy: a kombinációs hálózat logikáját memóriákkal valósítják meg → könnyen módosíthatók.

Mealy- és Moore-automaták

Az eddigiekben csak a szekvenciális hálózatok **állapot**aikról volt szó, az automatáknál viszont a **kívülre leadott jelek** is fontosak.

Példa: modulo 4 számláló „előre/hátra”-vezérlő bemenettel

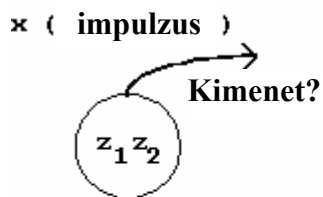


Állapot szerinti megadás:

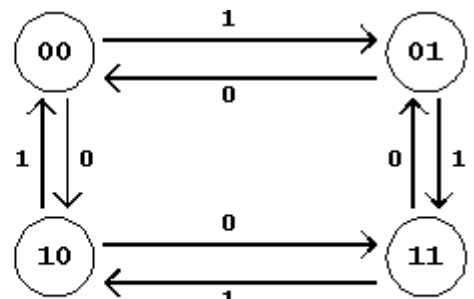
Megvalósítandó állapot-szekvencia:

00 01 11 10 00 ...

(egylépéses kód, a funkció-hazárd elkerülése miatt)



állapot



állapotdiagram

Szekvenciális hálózatok és automaták

Vezérlő hálózatok (Mealy- és Moore-automaták)

Aktuális állapot definíció szerint lehet

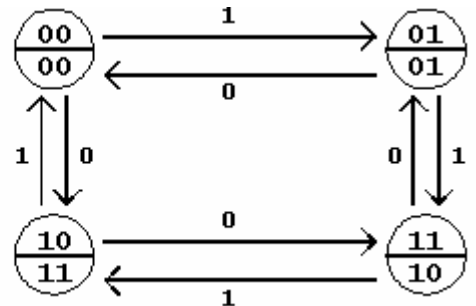
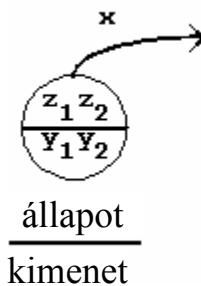
- a) visszacsatoló hálózat kimenete (z_1 és z_2) (*előnyösebb, mert stabil*)
- b) visszacsatoló hálózat bemenete (v_1 és v_2) (*közvetlenül függ az x bemenettől*)

Aktuális kimenet (y_i) definíció szerint lehet

- a) KH1 kimenete vagy
- b) KH2 kimenete (*előnyösebb, mert közvetlenül nem függ az x bemenettől*)

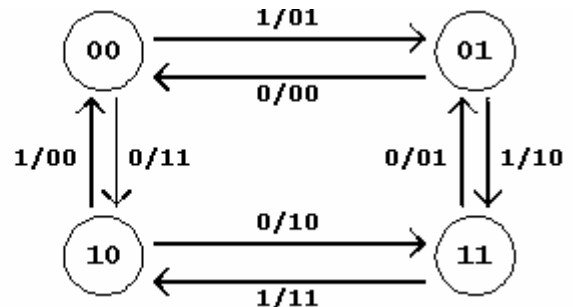
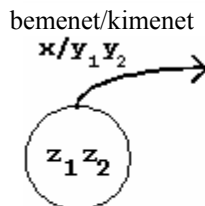
Moore-automata:

kimenet megegyezik KH2 kimenetével, azaz a bemenet csak az állapotváltozáson keresztül hat a kimenetre
→ szinkron kimenet



Mealy-automata:

a hálózat kimeneteit KH1 kimenetei képezik, azaz a kimenet közvetlenül függ a bemenettől és az állapottól is
→ aszinkron kimenet



Állapotváltozás - kimenetváltozás a Mealy-automatánál:

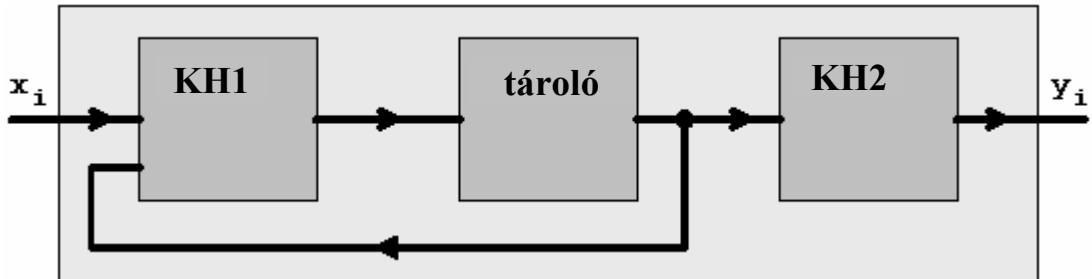
Például, ha „01” állapotban az x -bemenetre egy „1”-es kerül, akkor

- az órajeltől függetlenül az „10”-érték jelenik meg a kimeneten (az átfutási idő eltelte után)
- megtörténik az előkészítés az „11” állapotba való áttérésre, azaz ez az érték kerül a regiszter D-bemeneteire. Az új állapot viszont csak a következő hatásos órajel-él hatására áll be.

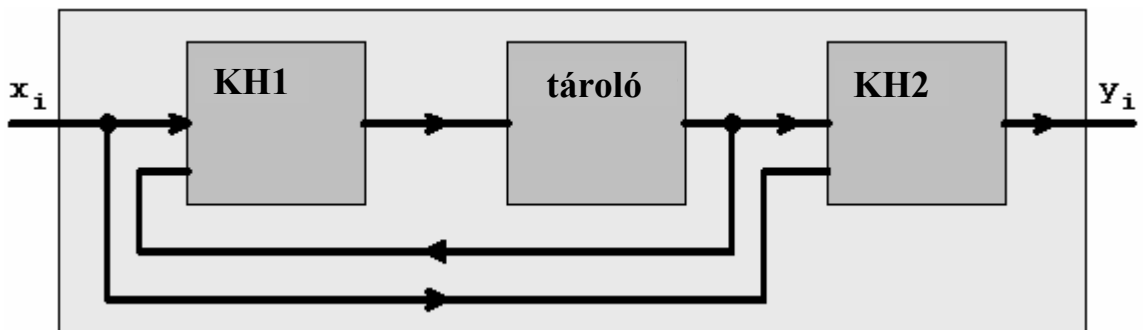
Szekvenciális hálózatok és automaták

Vezérlő hálózatok (Mealy- és Moore-automaták)

Moore-automata általános elvi kapcsolási rajza:



Mealy-automata általános elvi kapcsolási rajza:

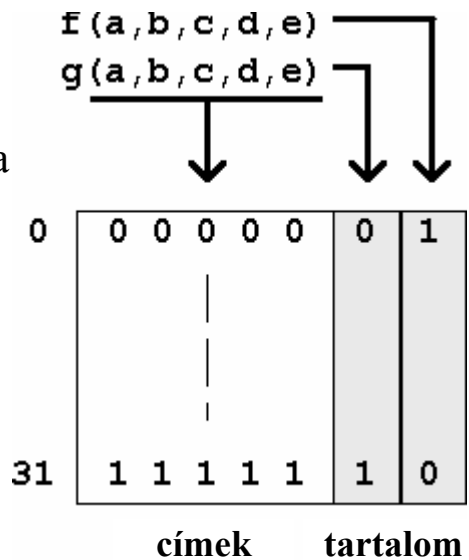


Szekvenciális hálózatok és automaták

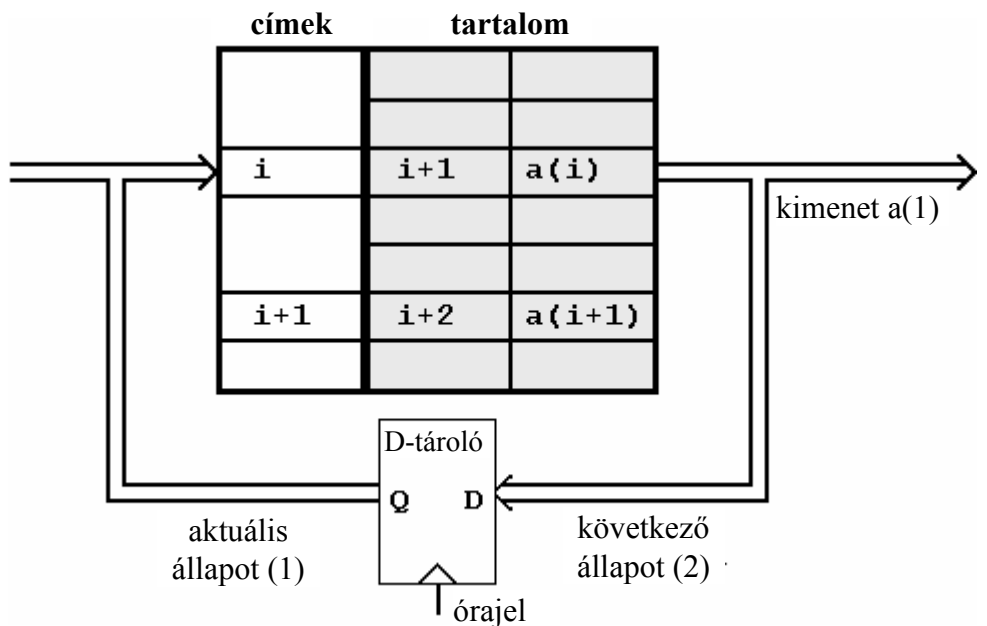
RAM-memóriás vezérlő hálózatok

RAM-memória tárolja és egyben végzi a vezérlő automata logikai funkcióját. A logikai függvény bemenő változói képezik az egyes regiszterek címét (szabad hozzáférés), a regiszterek tartalma pedig a kimeneti válasz-értékeket.

Példa: Két függvény, $f(a,b,c,d)$ és $g(a,b,c,d)$ memóriás megvalósítása.
Az össz.kapacitás: 32 x 2 tároló-cella (bit)



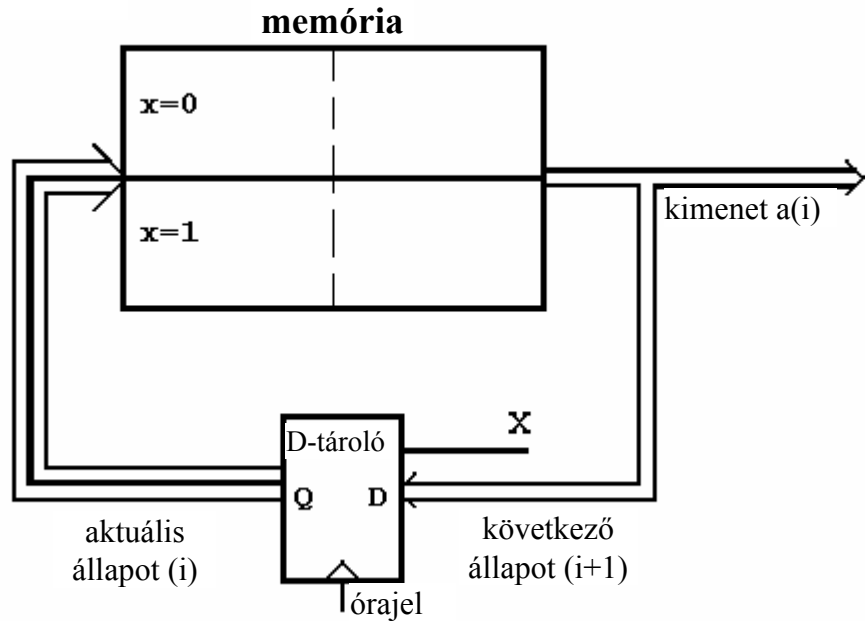
Egy általános automata memóriás megvalósítása:



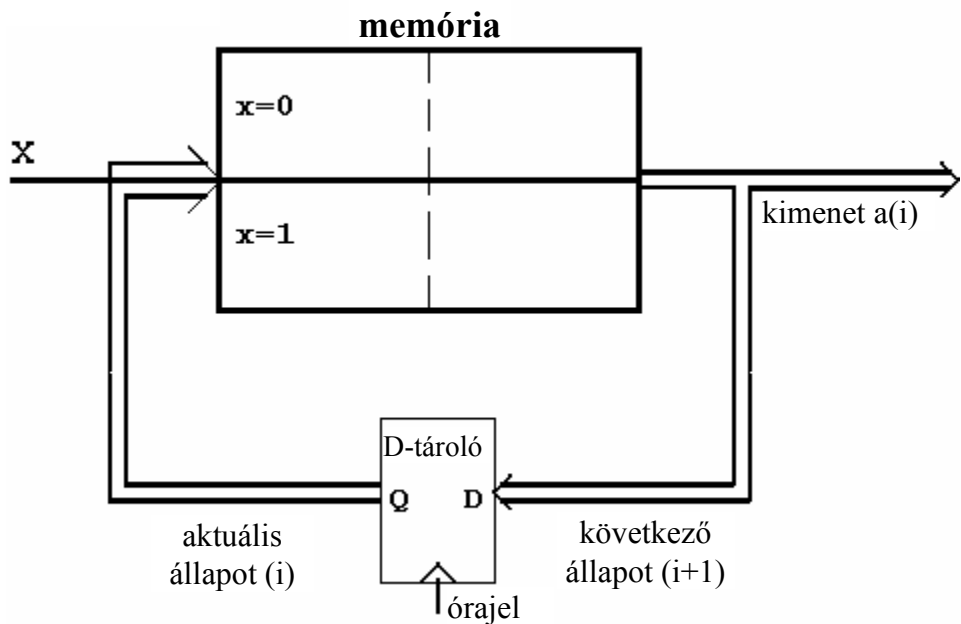
Szekvenciális hálózatok és automaták

RAM-memóriás vezérlő hálózatok

Moore-automata memóriás megvalósítása:



Mealy-automata memóriás megvalósítása:



Szekvenciális hálózatok és automaták

Mikroprogram-vezérlők

Memóriás megvalósítású vezérlőműveket elterjedten alkalmazták mikroprocesszorokban az utasítások végrehajtásának vezérlésére. (mikroprogram-vezérlők)

Egy adott mikroprocesszor-utasítást „mikroutasítások” sorozatára bontanak szét. A megfelelő sorrendet a mikroprogram-memória tárolja.

