

III. modul: Sorrendi hálózatok kezdeti állapotának beállítási lehetőségei

1. lecke: Szinkron és aszinkron sorrendi hálózatok kezdeti állapotának beállítása

Cél:

A lecke célja, hogy a hallgató megismerje a különféle sorrendi hálózatok kezdeti állapotának beállítási lehetőségeit. Képes legyen megállapítani a specifikáció szerint előírt kezdeti állapot beállítását, és amennyiben szükséges, meg tudja azt oldani az erre a célra alkalmas segédbeemenetek felhasználásával vagy megtervezésével.

Követelmények:

Ön akkor sajátította el megfelelően a tananyagot, ha

- ismeri a sorrendi hálózatok kezdeti állapotának ismérveit
- meg tudja állapítani egy adott feladat megoldása során a specifikációban előírt kezdeti állapot beállítását
- az adott célra alkalmas 'preset' és 'clear' segédbeemenetek felhasználásával vagy megtervezésével be tudja állítani a szükséges kezdeti állapotot egy szekvenciális hálózatban

Időszükséglet:

A tananyag elsajátításához kb. 60 percre van szükség.

Kulcsfogalmak:

- kezdeti állapot
- preset, clear
- RESET logika

Tevékenység: tanulmányozza a leckében bemutatott kezdeti állapot beállítási lehetőségeket és megoldásokat. Szimulátorban tervezze meg az áramköröket és a program segítségével ellenőrizze és elemezze működésüket!

1.1 Szinkron szekvenciális hálózatok kezdeti állapotának beállítása

Általánosságban elmondható, hogy a kezdeti állapotkódok beállítását tulajdonképpen nemcsak utólag, hanem a tervezéssel párhuzamosan is elvégezhetnénk, ha a bemenő jelek listájára felvennénk az '*R*' jelet, és már a szimbolikus összevont állapottáblán is figyelembe vennénk a lehetséges bemeneti kombinációk között. Ennek hátránya, hogy egyetlen járulékos bemeneti jel is jelentősen bonyolítja a tervezési folyamat valamennyi fázisát. Ezért célszerű ezt a lépést a tervezési folyamat végére hagyni, és a lehetséges megoldásokat részleteiben megvizsgálni.

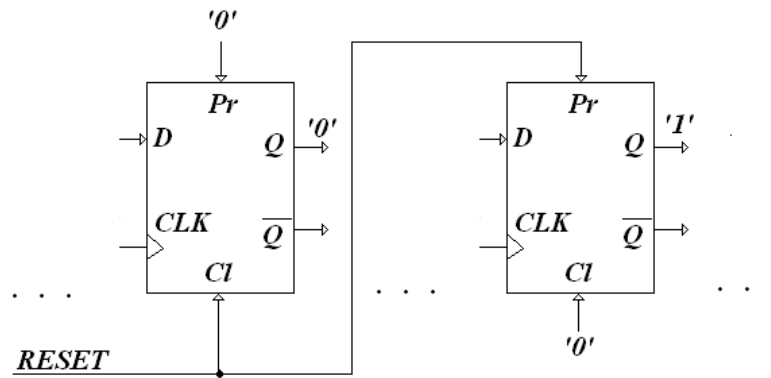
Szinkron sorrendi hálózatokat mindig 'mester-szolga' tárolókkal valósítunk meg, így a kezdeti állapot beállítása a flip-flopok kezdeti állapotainak beállítását jelenti. Ilyenkor - az adott *MS* tároló kialakításától függően - két lehetőség közül választhatunk:

1. a kezdeti állapot beállítása a 'preset' (Pr) és a 'clear' (Cl) segédbemenetek kihasználásával,
2. a kezdeti állapot beállítása az f_y hálózat kiegészítésével.

1.1.1 Kezdeti állapot beállítása a 'preset' és 'clear' segédbemenetek felhasználásával

Szinkron hálózattervezési mintapéldáink megoldásában a kezdeti állapot beállítását lehetővé tevő kiegészítések megtervezésekor kihasználtuk a flip-flopok 'preset' és 'clear' segédbemeneteit. A kezdeti állapot kódja példáinkban mindig csupa '0'-ból állt, így a flip-flopok 'preset' bemenetét '0'-ra kapcsoltuk, és valamennyi 'clear' bemenetére rákapcsoltuk a kezdeti állapotot kikényszerítő '*R*' ('reset') bemeneti jelet.

Ezt a módszert általánosítsuk tetszőleges kezdeti állapot kódra! A 1.1.1.a ábra egy fiktív *D-MS* flip-flop sorban (állapot-regiszterben) két különböző kezdeti értékű flip-flopot mutat. Nyilvánvaló, hogy a '0' kezdeti értékűek 'clear', míg az '1' kezdeti értékűek 'preset' bemenetét aktivizáljuk a '*RESET*' jellel.



1.1.1.a ábra[1]

Példa a 'preset'(Pr) és 'clear'(Cl) segédbemenetek felhasználása a kezdeti állapot beállításához

1.1.2 Kezdeti állapot beállítása az f_y hálózat kiegészítésével

Amennyiben az alkalmazott tárolónk nem tartalmaz külön 'preset' és 'clear' segédbemenetet, akkor egy kiegészítő hálózat megtervezésével és beiktatásával lehet megvalósítani a szükséges kezdeti állapotot beállító 'Pr' és 'Cl' funkciót.

D-MS flip-flop kiegészítése 'Pr' és 'Cl' segédbemenettel:

A D flip-flopok esetén alkalmazandó kiegészítő hálózatok megtervezéséhez a szükséges igazságtáblák a 1.1.2.a ábrán láthatók:

D_i	RESET	D_i'
0	0	0
0	1	0
1	0	1
1	1	0

D_j	RESET	D_j'
0	0	0
0	1	1
1	0	1
1	1	1

1.1.2.a ábra[1]

Igazságtáblák a D-MS tároló 'Pr' és 'Cl' segédbemeneteinek tervezéséhez

A baloldali tábla azt az esetet mutatja, amikor a 'clear' bemenethez tartozó RESET jel, valamint a D adatbemenetre (D_i) eredetileg csatlakozó f_y hálózati által megadott értékkombinációk feltüntetésével meghatározzuk, hogy milyen szükséges logikai szintet kell kapcsolni a 'clear' funkció elérése érdekében a D flip-flopunk adatbemenetére (D'_i).

A jobboldali táblázat pedig azt a helyzetet mutatja, amikor a 'preset' bemenethez tartozó RESET jel, valamint a D adatbemenetre (D_j) eredetileg csatlakozó f_y hálózati által megadott értékkombinációk feltüntetésével meghatározzuk, hogy milyen szükséges logikai szintet kell kapcsolni a 'preset' funkció elérése érdekében a D flip-flopunk adatbemenetére (D'_j).

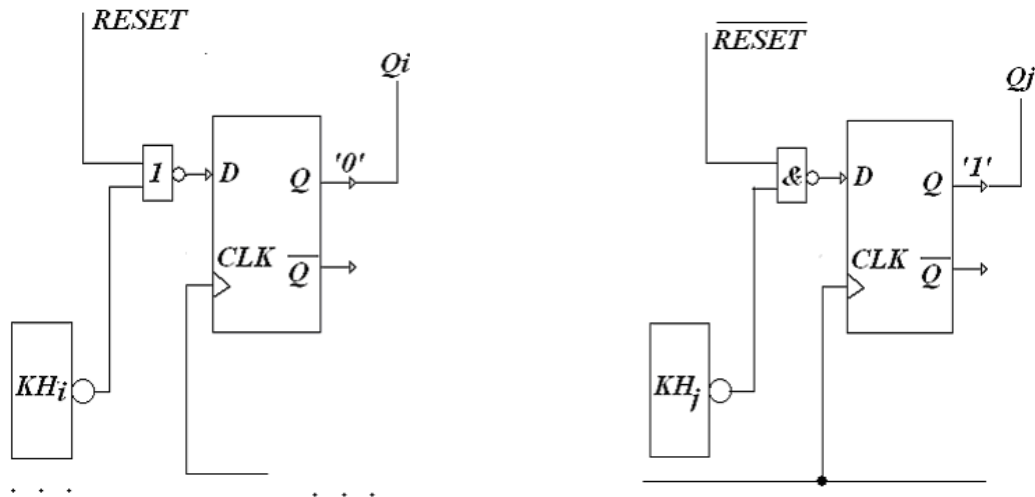
A táblázatból kiolvasott algebrai alakok:

$$D'_i = D_i \overline{RESET} = \overline{\overline{D_i} + RESET}$$

$$D'_j = \overline{\overline{D_j} RESET}$$

Megjegyzés: D'_j felírása esetében érdemes a szokásos mintermes kanonikus alakot „átfordítani” az egyszerűbb formulává, vagyis a kimenet egyetlen '0' értékéhez megadni a formula negáltját, természetesen a vonatkozó De-Morgan azonosságok felhasználása alapján.

A szükséges átalakításokat tartalmazó realizációk a 1.1.2.b ábrán láthatók:



1.1.2.b ábra[1]

A 'clear' (bal oldali séma) és a 'preset' (jobb oldali séma) kiegészítő hálózatok beillesztése D flip-flopok kezdeti állapotainak beállítására. A $\bar{D}_i$ és $\bar{D}_j$ logikai szintek előállításának érdekében a KH_i és KH_j hálózatok végére invertereket kell tennünk, ezeket a körök szimbolizálják.

JK-MS flip-flop kiegészítése 'Pr' és 'Cl' segédbemenettel:

A JK flip-flopok esetén alkalmazandó kiegészítő hálózatok megtervezéséhez a szükséges igazságtábla párok a 1.1.2.c ábrán láthatók:

J_i	RESET	J'_i	K_i	RESET	K'_i
0	0	0	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	1	0	1	1	1

J_j	RESET	J'_j	K_j	RESET	K'_j
0	0	0	0	0	0
0	1	1	0	1	0
1	0	1	1	0	1
1	1	1	1	1	0

1.1.2.c ábra[1]

Igazságtáblák a JK-MS tároló 'Pr' és 'Cl' segédbemeneteinek tervezéséhez

A felső táblapár azt az esetet mutatja, amikor a 'clear' bemenethez tartozó RESET jel, valamint a J és K bemenetekre ($J_i; K_i$) eredetileg csatlakozó f_y hálózati által megadott értékkombinációk feltüntetésével meghatározzuk, hogy milyen szükséges logikai szintet kell kapcsolni a 'clear' funkció elérése érdekében a JK flip-flopunk J és K bemenetére ($J'_i; K'_i$).

Az alsó táblázatpár pedig azt a helyzetet mutatja, amikor a 'preset' bemenethez tartozó RESET jel, valamint a J és K bemenetre ($J_j; K_j$) eredetileg csatlakozó f_y hálózati által megadott értékkombinációk feltüntetésével meghatározzuk, hogy milyen szükséges logikai szintet kell kapcsolni a 'preset' funkció elérése érdekében a JK flip-flopunk J és K bemenetére ($J'_j; K'_j$).

A D flip-flop esetében alkalmazott függvényfelírási gondolatmenet alapján az egyes algebrai alakok a következőképpen adódnak:

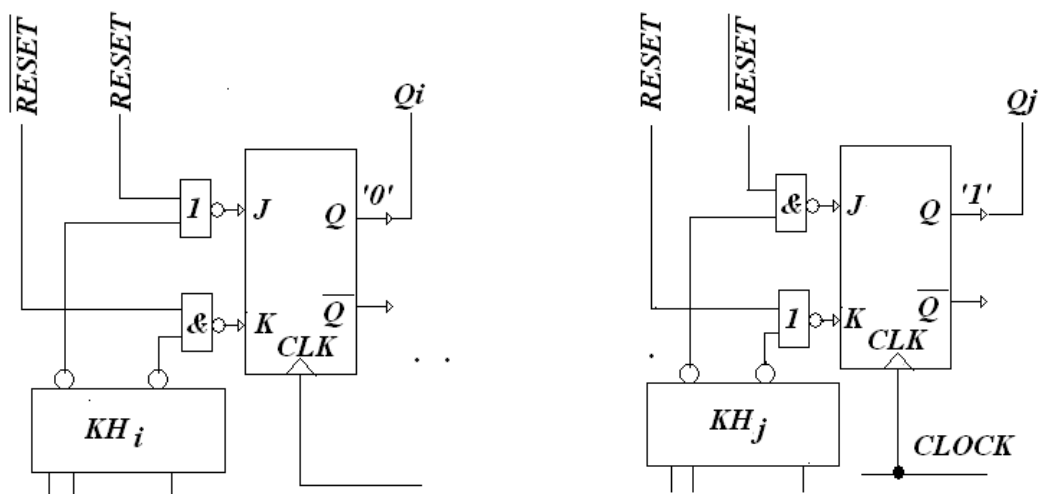
$$J'_i = \overline{\overline{J_i} + RESET}$$

$$K'_i = \overline{\overline{K_i} \overline{RESET}}$$

$$J'_j = \overline{\overline{J_j} \overline{RESET}}$$

$$K'_j = \overline{\overline{K_j} + RESET}$$

A megvalósított kapuszentű realizáció a 1.1.2.d ábrán látható:



1.1.2.d ábra[1]

A 'clear' (bal oldali séma) és a 'preset' (jobb oldali séma) kiegészítő hálózatok beillesztése JK flip-flopok kezdeti állapotainak beállítására. A $\bar{J}_i$ és $\bar{K}_i$ valamint $\bar{J}_j$ és $\bar{K}_j$ logikai szintek előállításának érdekében a KH_i és KH_j hálózatok végeire invertereket kell tennünk, ezeket a körök szimbolizálják.

Megjegyzés: a fent bemutatott módszer minden esetben biztosítja a kezdeti állapot beállítását a szekunder változók és a bemenetek aktuális állapotától függetlenül. Elvi megközelítés szerint a kezdeti állapotot a szekunder változók aktuális állapotának a módosításával is elő lehet állítani, ami ugyan egy egyszerűbb módszert jelent, de fontos megjegyezni, hogy nem biztosítja minden esetben a bemenetektől függetlenül a kezdeti állapot beállítását, ezért fejezetünkben ezt nem is tárgyaljuk.

1.2 Aszinkron szekvenciális hálózatok kezdeti állapotának beállítása

Aszinkron sorrendi hálózatok kezdeti állapotának beállításánál alapvetően a már megismert kétféle visszacsatolási séma alapján kell a szükséges kiegészítő hálózatot megtervezni illetve alkalmazni:

1. a kezdeti állapot beállítása közvetlen visszacsatolású aszinkron sorrendi hálózatokban,

2. a kezdeti állapot beállítása SR tárolóval visszacsatolt aszinkron sorrendi hálózatokban.

1.2.1 A kezdeti állapot beállítása közvetlen visszacsatolású aszinkron sorrendi hálózatokban

Ezzel a módszerrel már találkoztunk a korábbi mintafeladatok megoldása során. Tekintsük most újra át a megvalósítás lényegét: szekunder változónként fel kell hasítanunk a visszacsatolást, és be kell illesztenünk egy-egy olyan kiegészítő hálózatot, amely $RESET=0$ esetén a kiszámolt szekunder változó (eredeti) értéket, $RESET=1$ esetén pedig a kívánt kezdő értéket (0'-át vagy 1'-et) helyezi a megfelelő kimenetre. Jelöljük most ezt a kimenetet Y' -vel. Ezek alapján a kezdeti állapotot beállító segéd hálózatok igazságtábláit és logikai megoldásait a 1.2.1.a ábrán látható táblázatpár mutatja:

Y_i	RESET	Y_i'
0	0	0
0	1	0
1	0	1
1	1	0

Y_j	RESET	Y_j'
0	0	0
0	1	1
1	0	1
1	1	1

1.2.1.a ábra[1]

A 'RESET' logika értelmezése közvetlenül visszacsatolt aszinkron sorrendi hálózat kezdeti állapotainak beállításhoz

A baloldali tábla azt az esetet mutatja, amikor a '0' szükséges kezdeti állapotot beállító RESET jel, valamint az eredetileg visszacsatolódó f_y hálózati által megadott értékkombinációk feltüntetésével meghatározzuk, hogy milyen szükséges logikai szintet kell kapcsolni a '0' logikai értékkel bíró kezdeti állapot elérése érdekében a visszacsatoló jelet fogadó hálózat bemenetére.

A jobboldali tábla pedig azt az esetet mutatja, amikor az '1' szükséges kezdeti állapotot beállító RESET jel, valamint az eredetileg visszacsatolódó f_y hálózati által megadott

értékkombinációk feltüntetésével meghatározzuk, hogy milyen szükséges logikai szintet kell kapcsolni az '1' logikai értékkel bíró kezdeti állapot elérése érdekében a visszacsatoló jelet fogadó hálózat bemenetére.

Természetesen a fent bemutatott gondolatmenet alapján adódó logikai implementáció mindig az adott feladat specifikációja szerint illesztendő a megvalósítandó realizációba.

Megjegyzés: a kész realizáció ellenőrzésekor (pl. szimuláció) arra is kell ügyelni, hogy a kezdeti állapotot beállító RESET jelnek önmagában garantálnia kell, hogy a kezdeti állapot megjelenik a kombinációs hálózat kimenetein. Azt azonban, hogy ez a RESET jel eltűnése után stabilan meg is maradjon, azt a bemeneti kombinációt kell alkalmazni, amelyre a stabil kezdőállapotot előírtuk.

1.2.2 Kezdeti állapot beállítása SR tárolókkal visszacsatolt aszinkron sorrendi hálózatokban

Az SR tárolókkal visszacsatolt aszinkron sorrendi hálózatok esetében is a kezdeti állapot biztosításához sokszor elengedhetetlen feltétel, hogy a tárolók kimeneteit is a megfelelő logikai értékre állítsuk. 'Preset' és 'clear' segédbemenetekkel nem rendelkező SR tárolók esetében a kezdeti értéket beállító jelünket fogadó kiegészítő hálózat megtervezéséhez nyújt segítséget a RESET logikán felépülő igazságtábla pár, amely a 1.2.2.a ábrán látható:

S_i	RESET	S'_i	R_i	RESET	R'_i
0	0	0	0	0	0
0	1	0	0	1	1
1	0	1	1	0	1
1	1	0	1	1	1

S_j	RESET	S'_j	R_j	RESET	R'_j
0	0	0	0	0	0
0	1	1	0	1	0
1	0	1	1	0	1
1	1	1	1	1	0

1.2.2.a ábra[1]

Igazságtáblák az SR tároló 'Pr' és 'Cl' segédbemeneteinek tervezéséhez

A felső táblapár azt az esetet mutatja, amikor a 'clear' bemenethez tartozó RESET jel, valamint az S és R bemenetekre ($S_i; R_i$) eredetileg csatlakozó f_y hálózati által megadott értékkombinációk feltüntetésével meghatározzuk, hogy milyen szükséges logikai szintet kell kapcsolni a 'clear' funkció elérése érdekében az SR tárolónk S és R bemenetére ($S'_i; R'_i$).

Az alsó táblázatpár pedig azt a helyzetet mutatja, amikor a 'preset' bemenethez tartozó RESET jel, valamint az S és R bemenetre ($S_j; R_j$) eredetileg csatlakozó f_y hálózati által megadott értékkombinációk feltüntetésével meghatározzuk, hogy milyen szükséges logikai szintet kell kapcsolni a 'preset' funkció elérése érdekében az SR tárolónk S és R bemenetére ($S'_j; R'_j$).

A D flip-flop esetében alkalmazott függvényfelírási gondolatmenet alapján az egyes S és R algebrai alakok – a JK flip-flop módosult J és K bemenetére kapott kifejezésekkel hasonlóságot mutatva – a következőképpen adódnak:

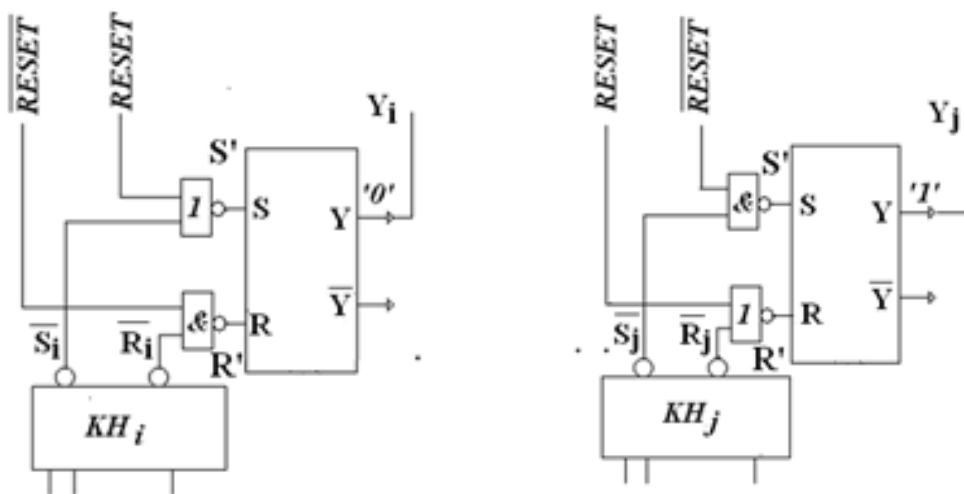
$$S'_i = \overline{\overline{S_i} + RESET}$$

$$R'_i = \overline{\overline{R_i} \overline{RESET}}$$

$$S'_j = \overline{\overline{S_j} \overline{RESET}}$$

$$R'_j = \overline{\overline{R_j} + RESET}$$

A realizáció a 1.2.2.b ábrán látható:



1.2.2.b ábra[1]

A 'clear' (bal oldali séma) és a 'preset' (jobb oldali séma) kiegészítő hálózatok beillesztése SR tároló kezdeti állapotainak beállítására. A $\bar{S}_i$ és $\bar{R}_i$ valamint $\bar{S}_j$ és $\bar{R}_j$ logikai szintek előállításának érdekében a KH_i és KH_j hálózatok végeire invertereket kell tennünk, ezeket a körök szimbolizálják.

Megjegyzés: ez a módszer csak a kezdeti állapothoz tartozó bemeneti érték/kombináció biztosítása mellett garantálja a stabil kezdeti állapot beállítását! Elvi lehetőség van még a szekunder változók módosítása által beállítani a szükséges kezdeti állapotot az SR tárolókkal visszacsatolt aszinkron sorrendi hálózatok esetében, de ennek hátránya, hogy a kialakítás a bemenetekre megadott kezdeti bemeneti kombinációval együtt sem biztosítja mindig a kezdeti állapot beállítását.

Felhasznált irodalom:

[1] Dr. Keresztes Péter: Digitális hálózatok (HEFOP elektronikus Jegyzet)