

V. modul: Állapotkódolási módszerek

1. lecke: Állapotkódolási módszerek szinkron és aszinkron sorrendi hálózatokban

Cél:

A lecke célja, hogy a hallgató megismerje a szinkron és aszinkron sorrendi hálózatok állapotkódolási lehetőségeit és azok alkalmazásának módszereit.

Követelmények:

Ön akkor sajátította el megfelelően a tananyagot, ha

- ismeri a szinkron és aszinkron sorrendi hálózatok különböző állapotkódolási módszereit, valamint képes az egyes hálózattípusok adott specifikációjához igazodva a megfelelő kódolási eljárás alkalmazására.

Időszükséglet:

A tananyag elsajátításához kb. 180 percre van szükség.

Kulcsfogalmak:

- „bit per state” elv
- szomszédos kódolás elve
- HT partíció
- instabil állapotok beillesztésének elve
- Tracey-Unger módszer
- hazárd kód
- „leselkedő” hazárd állapot

Tevékenység: tanulmányozza a leckében bemutatott példákon keresztül a különböző állapotkódolási módszereket, és ezek alapján néhány korábbi mintapéldát szabadon kiválasztva végezzen a tanult módszerek alapján állapotkódolásokat!

1. Állapotkódolási módszerek

1.1 Szinkron sorrendi hálózatok állapotkódolási módszerei

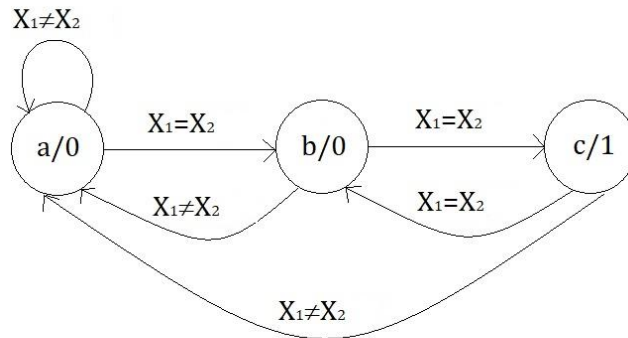
Szinkron sorrendi hálózatok tervezése során az állapotok megkülönböztetésére generált kódok esetében nincs versenyhelyzet veszély, így az állapotkódolás arra irányul, hogy a legegyszerűbb struktúrát alakítsuk ki. Ebben a fejezetben néhány gyakrabban használt kódolási eljárás kerül bemutatásra.

1.1.1 '1'-es súlyú állapotkódolás elve '(bit per state)'

Korábbi szinkron sorrendi hálózattervezési példáink során már találkoztunk azzal a speciális kódolási eljárással (2.1.3.2.), melynek során az egyes állapotokhoz rendelt kódszóban csak egyetlen bit helyén szerepel '1'-es. Ezt a kódolási eljárást neveztük '1'-es súlyú kódolásnak ('bit per state', one hot encoding'). Azt is megemlítettük, hogy elsősorban szinkron sorrendi hálózatok VLSI megvalósításakor alkalmazzák ezt a gyorsan célravezető módszert, hiszen ebben az esetben minden egyes szimbolikus állapothoz egy D-MS flip-flopot rendelünk.

Megjegyzés: elsősorban FPGA-k (Field-Programmable Gate Array - programozható logikai kapukat tartalmazó hálózat) esetében szokásos tervezési módszer az '1'-es súlyozású kódolás használata, hiszen ezekben az áramkörökben nagyszámú regiszter('egy bites' D-latch) található, de viszonylag kevés bemenetszámú a hozzá tartozó kombinációs hálózat. Ezért itt sokkal fontosabb, hogy egyszerű legyen a vezérlő függvény, aminek felírását mutattuk be egy korábbi mintapéldán (2.1.3.2.) keresztül.

A 2.1.1.2 pontban bemutatott mintafeladat állapotainak megkülönböztetésére '1'-es súlyozású kódokat is alkalmazhatunk. A feladat Moore-típusú gráfja a 1.1.1.a ábrán látható:



1.1.1.a ábra

Az első szinkron sorrendi hálózati mintafeladat Moore-típusú állapotgráfja

Alkalmazzunk 3 darab D flip-flopokat a realizációhoz, és kódoljuk az egyes állapotokat a következők szerint:

	Q_a	Q_b	Q_c
$a :$	1	0	0
$b :$	0	1	0
$c :$	0	0	1

A D bemenetek vezérlésének megvalósítása ilyenkor rendkívül egyszerű, és az állapotgráf alapján – a korábban már megismert módon – elvégezhető: minden egyes D bemenetre akkor és csak akkor kell '1'-et kapcsolni, amikor az általa reprezentált tároló kimenetnek '0'-ról '1'-re kell változnia, azaz amikor az adott flip-flop által reprezentált állapotnak be kell állnia. Ez pedig az állapotgráfból kiolvasható. További előny, hogy a kezdeti (a) alapállapotot beállító 'R' jelet is azonnal beilleszthetjük a függvényekbe.

Ezek alapján az egyes tároló vezérlések a következők:

$$D_a = R + \overline{R}(Q_a \overline{E} + Q_c \overline{E}) = R + \overline{E}(Q_a + Q_c)$$

$$D_b = \overline{R}E(Q_a + Q_c)$$

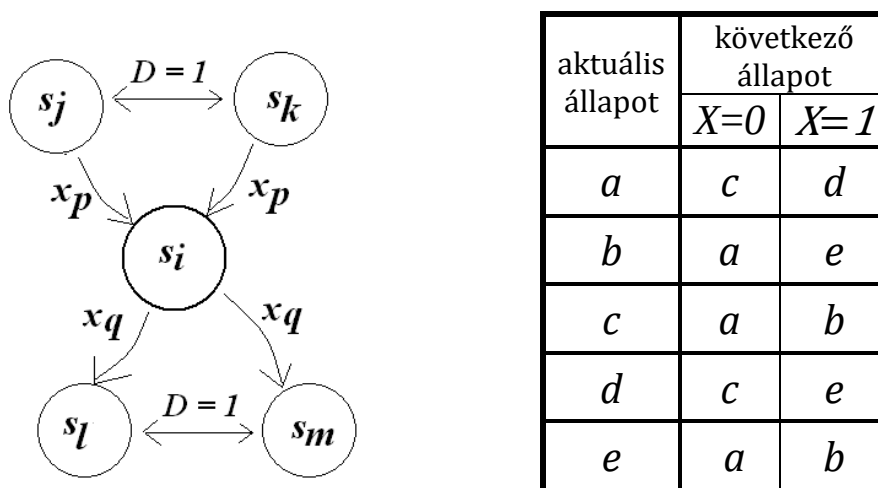
$$D_c = \overline{R}Q_b E$$

ahol ' E ' a két bemenet($x_1; x_2$) EXNOR függvénye.

1.1.2 Szomszédos kódolás elve

A szomszédos állapotok kódolásának elvét más néven heurisztikus módszernek is nevezik, mert a gyakorlati tervezés során tapasztalható, hogy az f_y hálózat egyszerűsítésére jótékony hatással vannak bizonyos állapotkód viszonyok. Egyszerűbb lesz a hálózat, ha egy adott állapot következő állapotainak kódjai bemenő kombinációként szomszédosak, illetve csak egy szekunder változóban különböznek egymástól, azaz a kódok közötti távolság egységnyi ($D=1$). Ugyancsak előnyös, ha azok az állapotok, amelyek valamely adott állapotnak az elődjei, bemenő kombinációként szomszédos kódúak.

Példaként tekintsük a 1.1.2.a ábrán szereplő állapotgráffal és egy lehetséges előzetes szimbolikus állapottáblával megadott szinkron sorrendi hálózatot:



1.1.2.a ábra[1]

Szomszédos állapotkódok és egy előzetes szimbolikus állapottábla

Az ábra bemutatja a szomszédos kódolást kifejező állapotgráfon a leírt előnyös kódolási helyzeteket. Az ábra jobb oldalán egy előzetes szimbolikus állapottábla látható, amelyen megkísérlünk szomszédos állapotkódokat találni. A fenti feltételek együtt nyilvánvalóan nem mindig teljesíthetők Ellentmondás esetén az egyszerűbb megoldásra vezető feltétel teljesítését kell előnyben részesíteni. Mivel a Karnaugh táblán a szomszédosság előnyösen szemléltethető, a szekunder változók számának megfelelő méretű Karnaugh

táblán ábrázolhatjuk a feltételek szerint valamilyen mértékben teljesített követelményeket, és az állapotkódokat egyszerűen kiolvashatjuk.

A példa szerinti állapottáblát a 1.1.2.b ábrán látható formába dolgozzuk át, azaz elkészítjük az állapotok utódjait és az állapotok elődeit bemutató táblázatokat:

állapotok	állapotok: „utódok”	
	$X=0$	$X=1$
a	c	d
b	a	e
c	a	b
d	c	e
e	a	b

állapotok	állapotok: „elődök”	
	$X=0$	$X=1$
a	b,c,e	
b		c,e
c	a,d	
d	a	
e		b,d

1.1.2.b ábra

Utódok és elődök a 1.1.2.a ábra állapottáblája alapján

Az állapotkódok optimális elhelyezését a 1.1.2.c ábrán szereplő táblázatok mutatják:

Közös utódja van $X = 0$ -ra:

b,c
b,e
c,e
a,d

Közös utódja van $X = 1$ -re:

b,d
c,e

Közös elődje van $X = 0$ -ra:

c,d

Közös elődje van $X = 1$ -re

-

Szomszédos kódok:

<i>mindkettő</i>	<i>közös utód</i>	<i>közös előd</i>
—	a,d	c,d
	b,c	
	b,d	
	b,e	
	c,e	

Q1 Q2 Q3	0	0	1	1
	0	1	1	0
0	e	b		
1	c	d	a	

1.1.2.c ábra[1]

Az állapotok elhelyezése Karnaugh táblán, az előnyös szomszédosságok legnagyobb arányú biztosítására

Kódoljuk be az állapotokat az alábbiak szerint!

	Q_1	Q_2	Q_3
a :	1	1	1
b :	0	1	0
c :	0	0	1
d :	0	1	1
e :	0	0	0
-	1	0	0
-	1	1	0
-	1	0	1

Ebből írjuk fel a kódolt állapot táblát (1.1.2.d ábra):

kódolt aktuális állapot, $Q_1^n \quad Q_2^n \quad Q_3^n$	kódolt következő állapot					
	X = 0			X = 1		
	D_1	D_2	D_3	D_1	D_2	D_3
1 1 1	0	0	1	0	1	1
0 1 0	1	1	1	0	0	0
0 0 1	1	1	1	0	1	0
0 1 1	0	0	1	0	0	0
0 0 0	0	1	0	1	1	1
1 0 0	-	-	-	-	-	-
1 1 0	-	-	-	-	-	-
1 0 1	-	-	-	-	-	-

1.1.2.d ábra

A kódolt állapot tábla

A kódolt állapot táblából felírt Karnaugh táblák a 1.1.2.e ábrán láthatók:

D1

		Q3				
		X	0	0	1	1
Q1Q2			0	1	1	0
00		1		1		
01	1					
11	-	-				
10	-	-	-	-		

D2

		Q3				
		X	0	0	1	1
Q1Q2			0	1	1	0
00	1	1	1	1		
01	1					
11	-	-	1			
10	-	-	-	-		

D3	Q3				
	X	0	0	1	1
Q1Q2		0	1	1	0
00		1			1
01	1				1
11	-	-	1		1
10	-	-	-		-

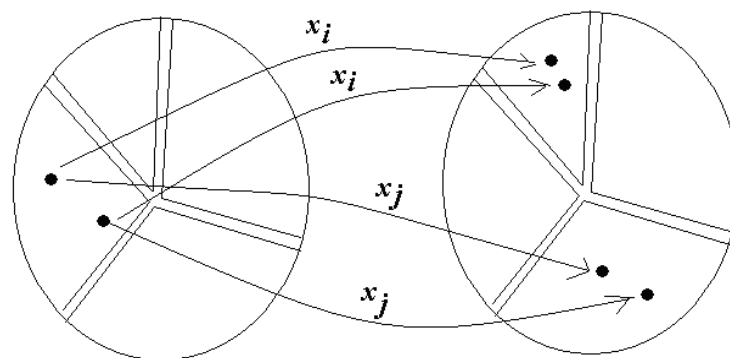
1.1.2.e ábra[1]

A kódolt állapottábla alapján megszerkesztett Karnaugh táblák, és a bennük létrehozott prímimplikánsok

1.1.3 A helyettesítési tulajdonságú (HT) partíciók alapján végzett kódolás elve: önfüggő szekunder változó csoportok keresése és kialakítása

Ez a módszer az állapothalmazon értelmezett partíciópárok elméletén alapul. A partíciópárok tulajdonságai, és azok összefüggése az állapotkódolással általánosíthatók. Az általánosítás eredményeképpen a következő állítások képezik a módszer elvi alapját: ha egy szinkron sorrendi hálózat állapotváltozóit csoportokra bontjuk, akkor a csoportokhoz komponenseket rendelhetünk. Minden egyes komponenshez két partíciót tartozik. Az egyik partíció azon állapotokat sorolja egy osztályba, amelyeket a komponens egyformán kódol. A másik partíció azokat az állapotokat sorolja egy osztályba, amelyeket a komponensre kapcsolódó környezet egyformán kódol. A környezeti partíció és a komponens partíció úgynevezett partíciópárt alkot.

Mivel a komponens a következő állapotának kialakításakor egy adott bemeneti kombináció esetén nem tesz különbséget két, a környezete által azonosan kódolt állapot között, a környezeti partíció ugyanazon osztályába tartozó állapotok következő állapotai bemeneti kombinációnként a komponens partíciónak is ugyanazon blokkjában vannak. Ezért nevezzük ezt a partíció kettőst partíciópárnak. A komponens partíciók metszete a legfinomabb partíciót eredményezi. Ha egy komponens bemeneteire nem kapcsolódnak más komponensek állapotváltozói, akkor a komponens partíció önmagával alkot partíciópárt (1.1.3.a ábra). Az ilyen partíciót helyettesítési tulajdonságú (HT) partíciónak nevezik.



1.1.3.a ábra[1]

A HT partíció önmagával partíció-párt alkot

Az HT partíciós komponenshez tartozó hálózat egyszerűbb, hiszen a többi komponens állapotváltozói nem tartoznak a bemenetei közé.

Célszerű tehát olyan komponensekre bontani a hálózatot, amelyek közül minél több független a többitől, azaz a szekunder változók önfüggő csoportjait reprezentálják. A tervezési módszer lényege, hogy az előzetes, szimbolikus állapottábla alapján HT partíciókat keresünk. Ha találunk nem triviális (nem a legfinomabb és nem a legdurvább) HT partíciót, akkor azzal önfüggő hálózat komponenset valósíthatunk meg.

Megjegyzés: ebben a leckerészben az önfüggő szekunder változó csoport keresés alapján végzett kódolási eljárás elméleti háttérének bemutatására kerül sor, gyakorlati példán keresztüli szemléltetését az alábbi linken lehet megtekinteni:
<http://www.sze.hu/~somi/Digit%e1lis%20h%e1l%ef3zatok/>

1.2 Aszinkron sorrendi hálózatok állapotkódolása

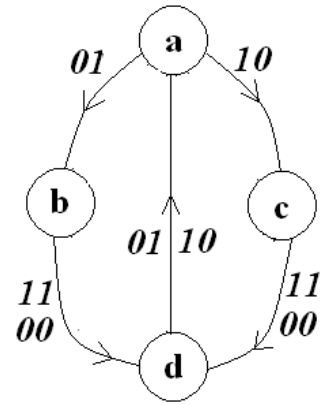
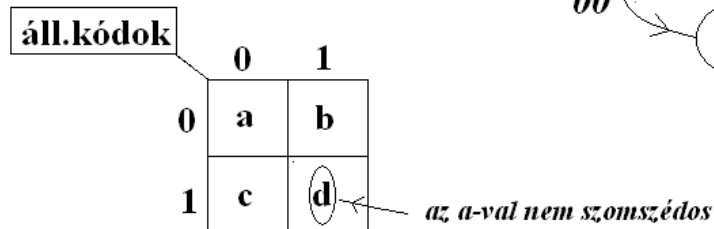
Aszinkron sorrendi hálózatok állapotkódolásakor a legfontosabb feladat a kritikus versenyhelyzetek elkerülése. Ennek eléréséhez kétféle állapotkódolási módszert lehet alkalmazni. Az egyik inkább próbálgatásos, ún. intuitív módszer(1.2.1), a másik egy elméletileg megalapozott, szisztematikus eljárás(1.2.2.).

1.2.1 Instabil állapotok beillesztésének elve (intuitív módszer)

Az eddigi feladatmegoldásoknál – aszinkron sorrendi hálózatok esetében – azt a módszert alkalmaztuk az egymást követő stabil állapotok kódjának megválasztásakor, hogy minden állapot átmenetre biztosítottuk a kódok egyetlen szekunder változó értékében való változását, hiszen a kritikus versenyhelyzet forrása éppen a többszörös változás. Vannak azonban esetek, amikor az állapotok számának kettes alapú logaritmus és a szekunder változók száma közeli értékek, így „be vagyunk szorítva”, és az ilyen kódolás nem is létezik. Ilyenkor növelni kell a szekunder változók számát. Segítségükkel instabil állapotokkal bővítjük az összevont állapottáblát. Az instabil állapotokkal való bővítést úgy kell megoldani, hogy a stabil állapotok egymás utáni sorrendje ne változzék, ugyanakkor az új instabil állapotokat úgy kell kódolni, hogy az egymás után következő tranziens kódok között csak egy szekunder változó értékében legyen különbség.

A fentiek szemléltetéseként nézzük a 1.2.1.a ábrán látható példát:

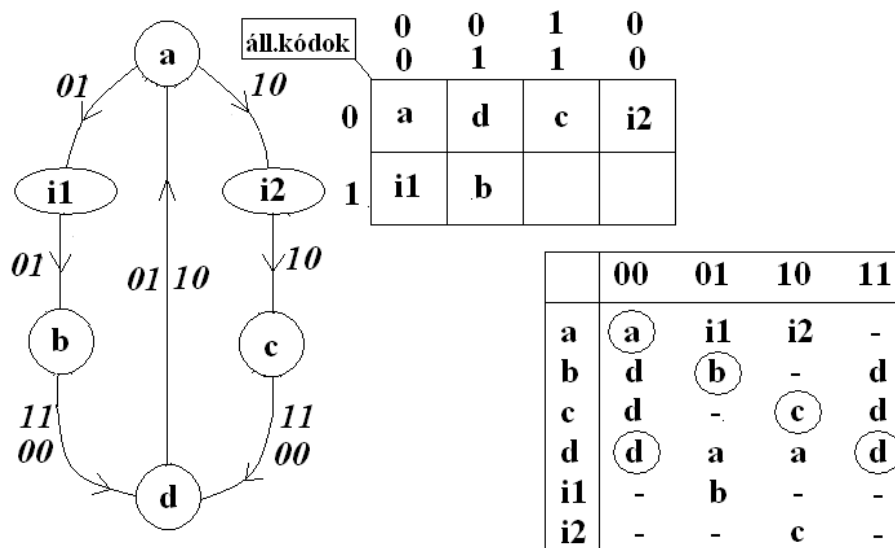
	00	01	10	11
a	a	b	c	-
b	d	b	-	d
c	d	-	c	d
d	d	a	a	d



1.2.1.a ábra[1]

Egy aszinkron sorrendi hálózat előzetes szimbolikus állapottáblája és a hozzá tartozó állapotgráfja

Be kell szűrnünk két instabil állapotot (i_1 és i_2), és ezzel egy új szekunder változót is! Az így kibővített állapothalmaz elemeinek kódjait már meg lehet úgy választani, hogy az egymás után következő tranziens állapotok kódjai szomszédosak legyenek. Lényeges tulajdonság, hogy sem az i_1 , sem az i_2 állapot soha sem stabilizálódik, de áthidaló tranziens szerepet töltenek be. Ez jól követhető az új állapot átmeneti táblázat segítségével is (1.2.1.b ábra):



1.2.1.b ábra[1]

Bővítés szomszédos kódú instabil állapotokkal

1.2.2 A Tracey-Unger módszer elve (szisztematikus módszer)

Aszinkron sorrendi hálózatokban a szekunder változók között kritikus versenyhelyzet akkor áll elő, ha egy stabil állapotból kiindulva megváltoztatjuk a bemeneti kombinációt, és ennek hatására olyan átmeneti állapotkód áll elő, amelynek sorában és az adott bemeneti kombináció oszlopában ez az állapotkód szerepel. A nem kívánt átmeneti állapotkódot hazárd kódnak nevezzük.

A Tracey-Unger módszer lényege, hogy a normális (tervezett) állapotátmenethez tartozó kiinduló és cél állapotok kódjai legalább egy adott szekunder változóban megegyezzenek, és ebben a változóban mindketten különbözzenek a hazárd kódtól.

Ilyenkor ugyanis ez a szekunder változó az átmenet során állandó marad, és így soha sem áll elő a hazárd állapot kódja.

Példa:

adott egy kétbemenetű (DC) aszinkron sorrendi hálózat összevont szimbolikus állapottáblája(1.2.2.a ábra):

		köv.áll / z			
akt.áll	D C				
		0 0	0 1	1 0	1 1
s1		s1 / 0	s1 / 0	s2 / 0	s1 / 0
s2		s1 / 0	- / -	s2 / 0	s3 / 1
s3		s4 / 1	s3 / 1	s3 / 1	s3 / 1
s4		s4 / 1	s1 / 0	s3 / 1	- / -

1.2.2.a ábra[1]

Egy kétbemenetű (DC) aszinkron sorrendi hálózat összevont szimbolikus állapottáblája

A kódolási eljárást az összevont szimbolikus állapottábla analízisével kezdjük: első lépésként egy listára felvesszük a stabil-stabil állapot átmeneteket, és hozzájuk írjuk azoknak a „leselkedő” hazard állapotoknak a nevét, amelyek az átmenet során felléphetnek, azaz azokat, amelyek a stabil célállapot oszlopában szerepelnek. Az analízis eredményeként kapott listás táblázat a 1.2.2.b ábrán látható:

00->01	00->10	01->00	01->11	10->00	10->11	11->01	11->10
s4->s1 s3 1	s1->s2 s3 2	s3->s4 s1 4	-	s2->s1 s4 5	s2->s3 s1 6	-	s1->s2 s3
	s4->s3 s2 3			s3->s4 s1			

1.2.2.b ábra[1]

Az összes lehetséges stabil-stabil állapotátmenet listázása a „leselkedők” (hazard kódos állapotok) feltüntetésével

Láthatjuk például, hogy a '00' → 01' megengedett bemeneti váltáshoz egyetlen stabil-stabil állapot átmenet olvasható ki, nevezetesen az s_4 állapotból az s_1 állapotba való átmenet. Ezt egyedül az s_3 állapot tranziens kódja veszélyezteti, tehát itt a leselkedő állapot az s_3 . Az ennek megfelelő kódolási szabályt a 1.2.2.c ábrán látható kódolási táblázatban az Y_1 szekunder változó képviseli, mégpedig azzal, hogy az oszlopában feltüntetett állapotkódban mind az s_1 , mind az s_4 '0'-val jelenik meg, a leselkedő s_3 viszont '1'-el! Az s_2 állapot Y_1 pozícióbeli kódrésze közömbös. Így szerkesztjük végig a táblázatot, ismétlésekbe nem bocsátkozunk, és a fordított irányú, de azonos leselkedőt mutató átmeneteket is értelemszerűen csak egyszer tüntetjük fel (ld. az áthúzott táblázat elemek: 1.2.2.b ábra).

	Y_1 1 2	Y_2 1 2	Y_3 1 2	Y_4 1 2	Y_5 1 2	Y_6 1 2
s_1	0 1	0 1	- -	1 0	0 1	1 0
s_2	- -	0 1	1 0	- -	0 1	0 1
s_3	1 0	1 0	0 1	0 1	- -	0 1
s_4	0 1	- -	0 1	0 1	1 0	- -

1.2.2.c ábra[1]

A stabil-stabil átmenetek és a hozzájuk tartozó leselkedők Tracey-Unger módszer szerinti kódolási lehetőségei

Következő lépésként vizsgáljuk meg az összes szekunder változóra a szabály alapján felírt lehetséges kódkiosztásokat. A szabályok csak a kötelező különbségtételt írják elő, azt hogy melyik szekunder változó legyen '1' és melyik legyen '0', azt nem. Ügyeljünk azonban arra, hogy ahol az adott szabály nem kényszerít az adott állapotváltozóra értéket, oda közömbös bejegyzést tegyünk. Belátható, hogy a közömbös bejegyzések tetszőleges konkretizálásával máris kritikus versenyhelyzettől mentes állapotkódot kaptunk. Ugyanakkor felismerhetjük, hogy a közömbös bejegyzések kihasználásával az oszlopokat összevonhatjuk. Ha az összevonás nehézségekkel jár, cseréljük fel szabadon az egyes oszlopokban az '1' és a '0' bejegyzéseket, és próbálkozunk újra az összevonással. A 1.2.2.d ábrán látható táblázatban egy lehetséges állapot összevonással kialakult csoportosítást végeztünk:

	$Y1$ 1 2	$Y2$ 1 2	$Y3$ 1 2	$Y4$ 1 2	$Y5$ 1 2	$Y6$ 1 2
$s1$	0 1	0 1	- -	1 0	0 1	1 0
$s2$	- -	0 1	1 0	- -	0 1	0 1
$s3$	1 0	1 0	0 1	0 1	- -	0 1
$s4$	0 1	- -	0 1	0 1	1 0	- -

1.2.2.d ábra[1]

Egy lehetséges szekunder változó csoport összevonás a kódolási lista alapján

Az összevont állapotváltozók indexei (Y_p : „piros” csoport; Y_z : „zöld” csoport) mutatják, mely oszlopokat sikerült összevonni.

Ezzel kialakult, hogy a versenyhelyzet-mentes kód végül is két szekunder változóval biztosítható (1.2.2.e ábra):

	Y_p	Y_z
$s1$	0	0
$s2$	0	1
$s3$	1	1
$s4$	1	0

1.2.2.e ábra[1]

A mintafeladat állapotainak kritikus versenyhelyzet-mentes kódolása két szekunder változó felhasználásával

Az összevont szimbolikus állapottáblából származtatott kódolt állapottábla a 1.2.2.f ábrán látható:

		$Y_p \dot{Y}_z / Z$			
Y_p^v	Y_z^v	$0\ 0$	$0\ 1$	$1\ 0$	$1\ 1$
0	0	$\textcircled{00}/0$	$\textcircled{00}/0$	$0\ 1/0$	$\textcircled{00}/0$
0	1	$0\ 0/0$	$--/-$	$\textcircled{01}/0$	$1\ 1/1$
1	1	$1\ 0/1$	$\textcircled{11}/1$	$\textcircled{11}/1$	$\textcircled{11}/1$
1	0	$\textcircled{10}/1$	$0\ 0/0$	$1\ 1/1$	$--/-$

1.2.2.f ábra[1]

A mintafeladat kódolt állapotáblája

A feladat megoldásának további lépéseit a már megismert módon kell elvégezni.

Felhasznált irodalom:

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