

9 SISTEME SECVENȚIALE ASINCRONE

9.1 Numărătoare asincrone

În paragraful 7.2 s-au prezentat numărătoarele sincrone, o clasă importantă de circuite secvențiale sincrone. De obicei, alături de aceste circuite se prezintă și **numărătoarele asincrone**, numite și **divizoare de frecvență**. Am preferat totuși să le prezentăm separat, pentru că ele nu sunt sisteme sincrone, în sensul că bistabilele din structură nu primesc același semnal pe intrările de ceas. De acord că avem totuși de-a face cu bistabile cu ceas, care basculează pe front, în timp ce sistemele asincrone nu au în general semnal de ceas.

Bistabilele din numărătoarele asincrone funcționează într-un singur fel, și anume, ca divizoare de frecvență prin 2. Prin urmare, bistabilele din structură, fie că sunt de tip D sau de tip JK, sunt transformate în bistabile de tip T, având intrarea de date T conectată la 1 logic. Schimbându-și starea pe fiecare front activ al ceasului, ele divizează prin 2 frecvența semnalului aplicat pe intrarea de ceas.

Dacă analizăm o succesiune de numere binare crescătoare, observăm că valoarea bitului mai puțin semnificativ (coeficientul lui 2^0) comută la creșterea cu o unitate a oricărui număr, iar valoarea biților imediat superiori comută întotdeauna ca o consecință a tranziției în 0 a bitului de rang imediat inferior. De aici rezultă ideea conectării bistabilelor în cascadă.

Figura 9.1 reprezintă schema unui divizor de frecvență de 4 biți realizat cu bistabile T

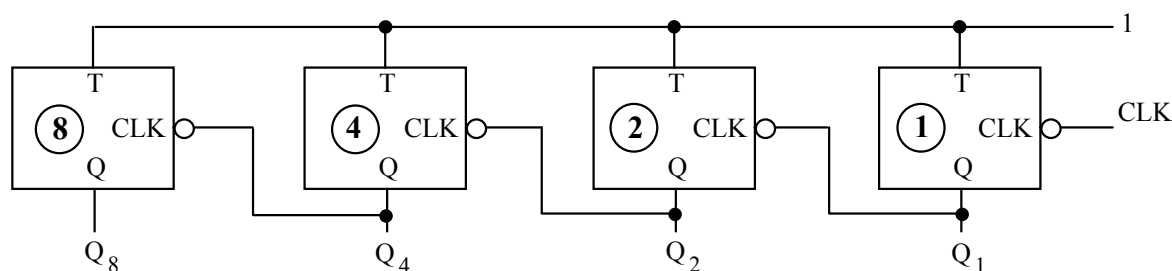
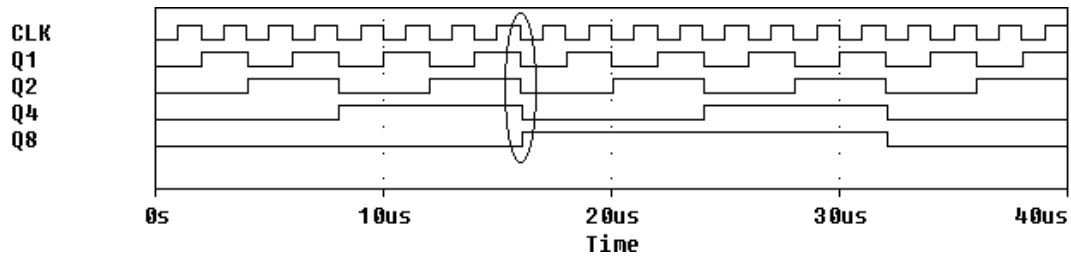
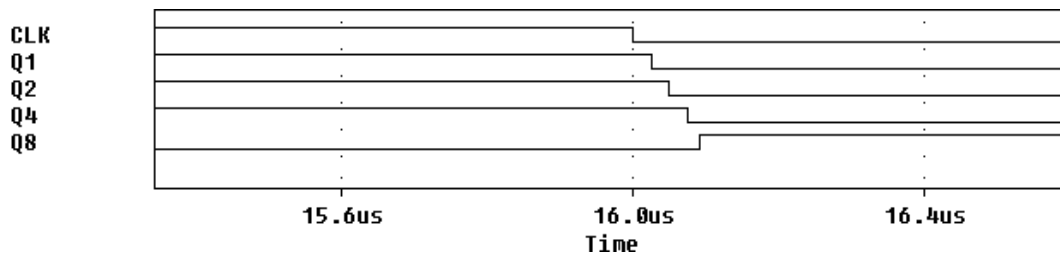
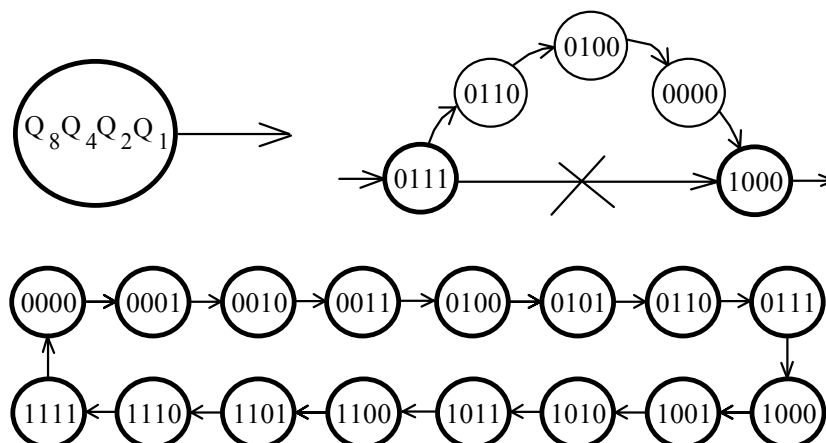


Fig. 9.1 Divizor de frecvență de 4 biți

Fig. 9.2 *Forme de undă pentru divizorul de frecvență de 4 biți*Fig. 9.3 *Forme de undă văzute printr-o “lupă de timp”*

Dacă acceptăm că bistabilele comută pe frontul descrescător al ceasului, atunci se obțin formele de undă prezentate în figura 9.2. La scara de timp la care s-a făcut simularea funcționării nu se observă nimic deosebit și diagrama stărilor pare a fi cea ideală, dată în figura 9.4. Dacă privim însă printr-o “lupă de timp”, aplicând un ZOOM IN repetat pe zona marcată din figura 9.2, vom obține formele de undă din figura 9.3 și vedem că lucrurile nu sunt chiar atât de simple.

Bistabilele din structură nu comută simultan. Frontul descrescător al semnalului CLK produce comutarea primului bistabil din dreapta, dar ieșirea lui, notată cu Q_1 , se modifică cu o mică întârziere, dată de timpul de propagare a informației prin porțile din structura bistabilului. Semnalul Q_1 este ceasul pentru bistabilul 2, iar semnalul Q_2 se modifică și el cu o mică întârziere față de Q_1 , și așa mai departe. Rezultă că avem de fapt mult mai multe stări decât cele date în figura 9.4. Dacă privim semnalele din figura 9.3 constatăm că între stările 0111 și 1000 tranziția nu se face direct, ci prin stările 0110, 0100 și 0000. Numim aceste stări de scurtă durată, **stări tranzitorii**, iar celelalte 16 stări vor fi **stări permanente**.

Fig. 9.4 *Diagrama ideală a stărilor și fragment din diagrama reală*

Este clar cum se construiește un divizor de frecvență, atunci când numărul de stări permanente este o putere a lui 2. Să vedem cum construim un astfel de circuit, dacă numărul stărilor permanente nu este o putere a lui 2.

Metoda de sinteză a divizorului de frecvență cu p stări:

1. se stabilește numărul necesar m de bistabile, rezolvând inecuația: $2^{m-1} < p < 2^m$
2. se realizează configurația de divizor de frecvență cu m bistabile
3. se detectează cu ajutorul unei funcții binare apariția stării p și se resetează numărătorul

Exemplul 9.1

Să se implementeze un divizor de frecvență cu 5 stări, folosind bistabile de tip JK și porți logice.

Conform metodei descrise mai sus, $p = 5$, deci $m = 3$. Cele 3 bistabile se conectează în cascadă, iar intrările lor se conectează la 1 logic (se transformă în T, cu $T = 1$). Detectarea stării 5 se face cu o funcție binară de 3 variabile: Q_4 , Q_2 și Q_1 . Stările 0, 1, 2, 3 și 4 sunt stări utile, deci funcția are valoarea logică 1 în aceste cazuri (resetarea bistabilelor se face pe 0 logic). Pentru starea 5 funcția este 0, iar pentru stările 6 și 7 valoarea ei nu contează. Prin minimizarea funcției cu diagrama Veitch-Karnaugh rezultă că este implementabilă cu o poartă ȘI-NU cu două intrări.

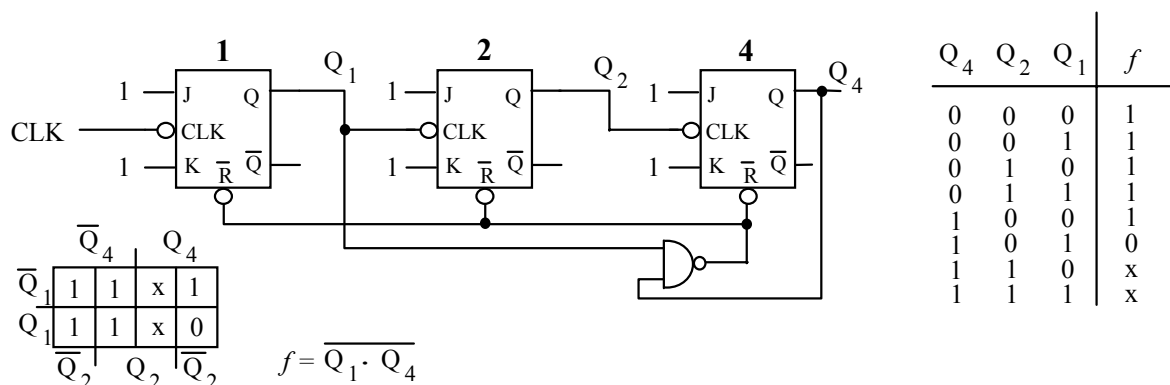


Fig. 9.5 Sinteza divizorului de frecvență cu 5 stări

Starea 5 apare pentru puțin timp, pentru că ea este cauza care resetează bistabilele din structură, dar ea este o stare tranzitorie. Durata ei este dată de timpul de propagare prin poarta ȘI-NU și prin cel mult două porți interne din structura bistabilului JK.

9.2 Hazard

Formalismul algebrei boolene folosit la analiza și sinteza structurilor combinaționale nu ținea seamă de întârzierile semnalelor prin porți. Acestea erau considerate ideale, adică aveau timp de propagare nul. Această ipoteză era de cele mai multe ori justificată, dacă ieșirile erau citite după un timp suficient de lung. Necesitatea de a ține seamă de întârzierile prin porți a apărut la structurile secvențiale, unde s-a introdus modelul logic asincron, model care explică apariția **hazardului**.

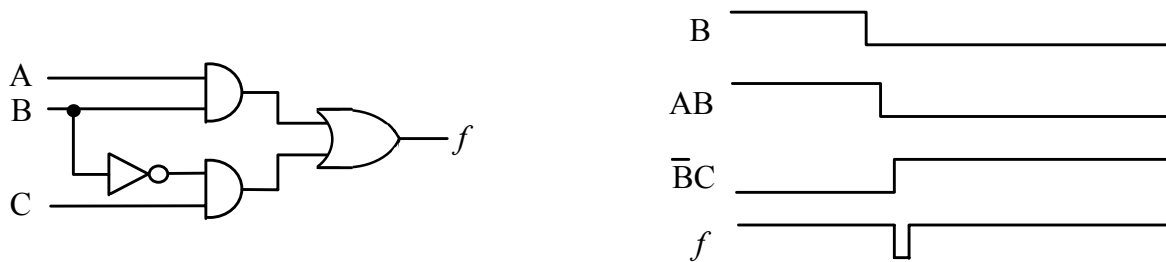


Fig. 9.6 Circuit care generează hazard static



Fig. 9.7 Evidențierea hazardului static pe diagrama Veitch-Karnaugh

În sistemele secvențiale sincrone nu se pune de obicei problema analizei hazardului, pentru că frontul activ al ceasului se aplică după ce logica combinațională a livrat valorile logice așteptate ale funcțiilor de excitație. În sistemele asincrone însă, această analiză a hazardului și găsirea unor modalități de eliminare a lui, este absolut necesară.

Se spune că un circuit logic are **hazard static** la ieșire, dacă există cel puțin o situație în care, prin modificarea valorii logice a unei variabile de intrare se produce o modificare de scurtă durată a valorii la ieșire, așa cum se poate vedea în figura 9.6. Dacă variabilele A și C sunt la nivel logic 1 și variabila B trece din 1 în 0 logic, atunci ieșirea f trece pentru un timp egal cu timpul de propagare prin poartă în 0 logic. De unde știm însă dacă un circuit are sau nu hazard, fără să mai desenăm de fiecare dată formele de undă, mai ales că este greu să reprezentăm pe forme de undă toate tranzițiile posibile ale variabilelor?

Este suficient să constatăm dacă, pentru anumite valori particulare date unui număr de $n-1$ variabile ale funcției de n variabile, funcția poate fi scrisă sub forma $x + \bar{x}$, unde x este variabila rămasă a funcției. În exemplul din figura 9.6, dacă $A = C = 1$, funcția f devine $f = B + \bar{B}$, deci are hazard pentru variabila B .

O metodă mai simplă de punere în evidență a hazardului static este metoda diagramelor Veitch-Karnaugh. Funcția dată în figura 9.6 se reprezintă prin diagrama din partea stângă a figurii 9.7. Compartimentele care conțin 1 logic sunt acoperite de două grupări diferite, care nu se intersectează. Observăm că prin schimbarea valorii logice a variabilei B se trece de la o grupare la alta, deci comutarea variabilei respective produce hazard. Eliminarea hazardului se poate face prin introducerea unei grupări suplimentare, care să includă mintermenii $\bar{A}\bar{B}C$ și ABC , după cum se vede pe diagrama din partea dreaptă a figurii 9.7. Funcția care nu mai are hazard devine: $f = AB + \bar{B}C + AC$. Evident că există circuite care produc hazard static la comutarea unei variabile din 0 în 1 logic.

Hazardul dinamic este definit ca o comutare multiplă a ieșirii datorită unei singure tranziții a intrării. Acest lucru se poate întâmpla dacă există mai multe căi cu întârzieri diferite între variabila care produce hazardul și ieșirea circuitului. Exemplul din figura 9.8 ilustrează un posibil hazard dinamic la comutarea variabilei B din 0 logic în 1 logic. S-a făcut ipoteza că poarta SAU care generează termenul $A + B$ este mai lentă decât celelalte

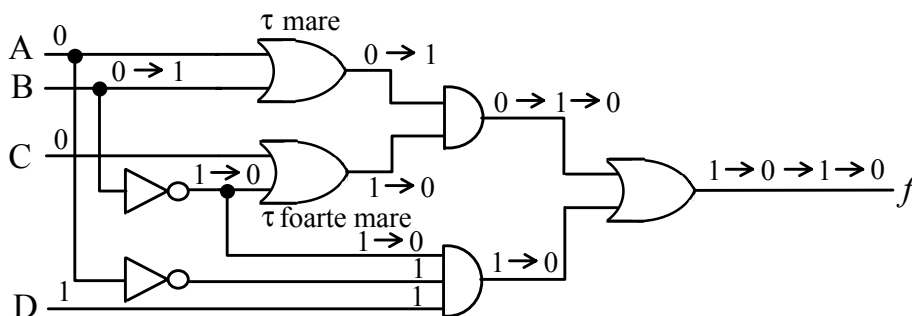


Fig. 9.8 Circuit care generează hazard dinamic

porți din circuit, iar poarta SAU care generează termenul $\overline{B} + C$ este cea mai lentă dintre toate. Găsirea hazardului dinamic dintr-un circuit este o problemă dificilă și ea rămâne, totuși, mai mult o preocupare academică, pentru că structurile combinaționale cu două nivele de logică care rezultă fie prin minimizarea funcțiilor binare, fie prin implementarea lor într-o arhitectură de tip PLD sau FPGA, nu generează hazard dinamic.

9.3 Analiza sistemelor secvențiale asincrone

Sistemele secvențiale asincrone sunt structuri de porți logice care conțin reacții de la ieșiri la intrări. Starea următoare este definită aici de întârzierile prin porți și nu de momentul apariției unuia dintre fronturile semnalului de ceas. Ne amintim că am discutat despre astfel de circuite în capitolul 6, când am prezentat structurile interne ale latch-urilor.

Pentru a face analiza unui astfel de circuit trebuie să “înterupem” buclele de reacție, astfel încât valoarea următoare memorată în fiecare buclă să fie o funcție dependentă de intrările circuitului și de valorile prezente memorate în toate buclele. Vom considera structura latch-ului de tip D din figura 9.9 și vom face analiza circuitului. Acest circuit are o singură buclă, buclă care va fi “înteruptă” prin introducerea unui buffer fictiv. Ieșirea buffer-ului este notată cu q , care este singura variabilă de stare pentru exemplul considerat.

Să mai presupunem acum că buffer-ul introdus este elementul de întârziere, iar toate celelalte porți sunt ideale, deci au timp de propagare nul. Putem calcula valoarea următoare a lui q , notată cu q^+ , cu ajutorul ecuației de stare:

$$q^+ = \overline{\overline{C \cdot D} \cdot \overline{\overline{C \cdot D}} \cdot q} = C \cdot D + \overline{C} \cdot q + D \cdot q$$

Cu ajutorul acestei ecuații putem completa tabelul tranzițiilor, care este dat în figura 9.10. Cele două stări 0 și 1 logic au fost notate cu q_0 și respectiv cu q_1 .

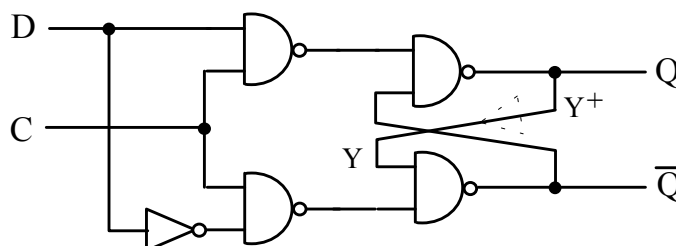


Fig. 9.9 Analiza unui latch de tip D

$q \backslash CD$	q^+			
	00	01	11	10
q_0	(q_0)	(q_0)	q_1	(q_0)
q_1	(q_1)	(q_1)	(q_1)	q_0

$q \backslash CD$	$q^+, Q \bar{Q}$			
	00	01	11	10
q_0	$(q_0, 01)$	$(q_0, 01)$	$q_1, 11$	$(q_0, 01)$
q_1	$(q_1, 10)$	$(q_1, 10)$	$(q_1, 10)$	$q_0, 01$

Fig. 9.10 Tabelul tranzițiilor și câteva tranziții pe tabelul tranzițiilor și al ieșirilor

Stările încercuite sunt **stări total stabile**, adică stări în care sistemul “buclează” de mai multe ori, ieșirea din aceste stări făcându-se numai prin modificarea intrărilor. Ieșirile sistemului sunt Q și $\bar{Q}$, iar ecuațiile ieșirilor sunt:

$$Q = C \cdot D + \bar{C} \cdot q + D \cdot q$$

$$\bar{Q} = C \cdot \bar{D} + \bar{q}$$

Rezultă tabelul tranzițiilor și al ieșirilor din figura 9.10. Tranzițiile din acest tabel sugerează faptul că o modificare simultană a intrărilor ar putea produce tranziții necontrolate. De fapt, nu există “simultan” în inginerie! Una din intrări se va modifica întotdeauna prima. Dacă ne aflăm, de exemplu, în starea total stabilă q_1 , având intrările $CD = 11$, tranziția “simultană” a lor în 00 va genera în final fie starea q_0 , dacă intrarea D se modifică prima, fie starea q_1 , dacă intrarea C se modifică prima.

Să analizăm în continuare un sistem care are mai multe bucle. Circuitul din figura 9.11 are structura unui bistabil de tip D cu declanșare pe front. Problema este să stabilim numărul minim de bucle. Chiar dacă vom considera mai multe bucle decât există, soluția analizei va fi corectă, pentru că surplusul de stări care rezultă se va reduce prin tehnicile cunoscute de reducere a stărilor de la sistemele sincrone.

Se pot evidenția 3 bucle și rezultă următoarele ecuații pentru stările următoare:

$$q_1^+ = q_2 \cdot D + q_1 \cdot CLK$$

$$q_2^+ = q_1 + \bar{CLK} + q_2 \cdot D$$

$$q_3^+ = q_1 \cdot CLK + q_1 \cdot q_3 + q_3 \cdot \bar{CLK} + q_2 \cdot q_3 \cdot D$$

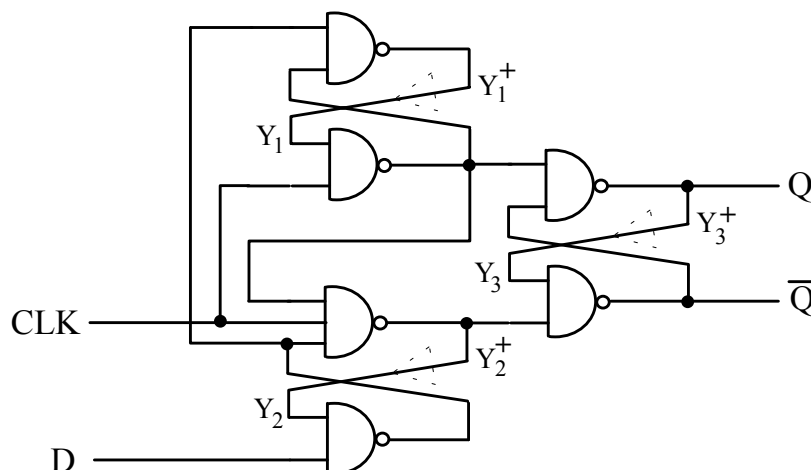


Fig. 9.11 Analiza unui bistabil de tip D

Pentru cele două ieșiri ale circuitului rezultă ecuațiile:

$$Q = q_1 \cdot CLK + q_1 \cdot q_3 + q_3 \cdot \overline{CLK} + q_2 \cdot q_3 \cdot D$$

$$\overline{Q} = \overline{q_3} + \overline{q_1} \cdot \overline{q_2} \cdot CLK + \overline{q_1} \cdot CLK \cdot \overline{D}$$

Tabelul tranzițiilor este reprezentat în figura 9.12. Cele 3 variabile de stare q_1 , q_2 și q_3 generează 8 stări interne posibile, iar stările total stabile s-au încercuit.

Dacă prin modificarea unei singure variabile de intrare se modifică cel puțin două variabile interne, atunci spunem că avem o **cursă** între cele două stări. De exemplu, dacă suntem în starea total stabilă 011 și se modifică intrările CLK și D din 00 în 10, atunci starea internă următoare devine 000, o modificare de 2 biți față de 011, deci avem o cursă.

De ce este periculoasă o cursă? Pentru că niciodată semnalele numerice nu se modifică simultan, deci tranziția $011 \rightarrow 000$ se face fie prin $011 \rightarrow 010 \rightarrow 000$, fie prin $011 \rightarrow 001 \rightarrow 000$. Vedem însă pe tabel că, oricum s-ar întâmpla, starea finală va fi tot 000, deci circuitul funcționează conform așteptărilor și o astfel de cursă se numește **cursă necritică**.

Dacă presupunem acum că pe a treia linie din tabel și pe ultima coloană, în loc de 000 avem ca stare următoare 110, atunci lucrurile se schimbă și acest lucru se poate vedea pe tabelul din partea dreaptă a figurii 9.12. De data aceasta, la modificarea intrării CLK din 0 în 1 logic tranziția din starea stabilă 011 se face fie prin $011 \rightarrow 000 \rightarrow 000$, fie prin $011 \rightarrow 000 \rightarrow 110 \rightarrow 111$, funcție de viteza și ordinea de modificare a variabilelor interne. O astfel de cursă se numește **cursă critică** și proiectantul trebuie să asigure eliminarea acestor tranziții critice. Altfel se poate întâmpla ca circuitul să funcționeze imprevizibil, starea următoare în astfel de situații depinzând de “factori ca temperatura, tensiunea de alimentare sau fazele lunii” ([Wakerly, 1990]).

Exemplul considerat nu are curse critice, deci putem construi tabelul stărilor și al ieșirilor din figura 9.13. Numai că tabelul construit are mai puține linii, deoarece am eliminat stările care nu sunt total stabile. Comportarea circuitului nu se modifică, doar unele tranziții de la o stare stabilă la alta ar putea avea mai puține stări tranzitorii. S-au eliminat și acele stări următoare care nu pot fi atinse dintr-o anumită stare total stabilă prin schimbarea unei singure variabile de intrare. Și aici putem aplica o procedură formală de reducere a stărilor, deoarece constatăm că stările 000 și 010 sunt compatibile. Tabelul redus al stărilor și ieșirilor este reprezentat tot în figura 9.13.

CLK D $q_1 q_2 q_3$	$q_1^+ q_2^+ q_3^+$			
	00	01	11	10
000	010	010	000	000
001	011	011	000	000
010	010	110	110	000
011	011	111	111	000
100	010	010	111	111
101	011	011	111	111
110	010	110	111	111
111	011	111	111	111

CLK D $q_1 q_2 q_3$	$q_1^+ q_2^+ q_3^+$			
	00	01	11	10
000	010	010	000	000
001	011	011	000	000
010	010	110	110	110
011	011	111	111	000
100	010	010	111	111
101	011	011	111	111
110	010	110	111	111
111	011	111	111	111

Fig. 9.12 Tabelul tranzițiilor și o cursă critică pentru circuitul din figura 9.11

CLK D		$q_1^+ q_2^+ q_3^+, Q \bar{Q}$				CLK D		$q_1^+ q_2^+ q_3^+, Q \bar{Q}$			
$q_1 q_2 q_3$		00	01	11	10	$q_1 q_2 q_3$		00	01	11	10
000		010,01	010,01	000,01	000,01	000		000,01	110,01	000,01	000,01
010		010,01	110,01	- , -	000,01	011		011,10	111,10	- , -	000,01
011		011,10	111,10	- , -	000,01	110		010,01	110,01	111,11	- , -
110		010,01	110,01	111,11	- , -	111		011,10	111,10	111,10	111,10
111		011,10	111,10	111,10	111,10						

Fig. 9.13 Tabelul tranzițiilor stabile și al ieșirilor și tabelul redus

Pe tabelul redus se pot verifica diverse tranziții posibile ale circuitului. Vedem, de exemplu, că dacă suntem în starea stabilă 000, având intrările $CLK D = 10$ și se modifică intrarea D în 1 logic, atunci circuitul își păstrează starea și ieșirile. Dacă trece intrarea CLK în 0 logic, atunci fie rămânem în aceeași stare, fie trecem în 110, dar ieșirile, în continuare, nu se modifică. Atunci când CLK revine la 1 logic, dacă eram în starea 000 starea și ieșirile se păstrează, iar dacă eram în 110 se trece în starea 111 cu ieșirile 10.

9.4 Sinteza sistemelor secvențiale asincrone

Pentru proiectarea unei structuri secvențiale asincrone trebuie să reprezentăm funcționarea sistemului printr-un tabel de tranziții. Fiecare stare din tabel trebuie să fie corelată cu un eveniment care are loc în circuit, asemănător cu cazul sistemelor sincrone. Pentru ca ieșirea sistemului să depindă numai de stare, se recomandă ca acest prim tabel construit, să conțină câte o stare stabilă pe fiecare rând. S-ar putea ca tabelul rezultat să conțină prea multe stări, dar vom reduce numărul lor printr-o metodă asemănătoare celei de la sistemele sincrone.

Vom încerca să proiectăm un circuit care are următoarea descriere: circuitul are două intrări, P (puls) și R (Reset), și o ieșire Y care are valoarea logică 0. Ieșirea trebuie să se poziționeze în 1 logic la fiecare tranziție de la 0 logic la 1 logic a intrării P și să revină la 0 logic la fiecare activare prin 1 logic a intrării R . Admitem că nu este posibilă o activare simultană a semnalelor de intrare. Tabelul tranzițiilor este reprezentat în figura 9.14.

P R		q^+				Y	Semnificație
q		00	01	11	10		
IDLE		IDLE	RES1	-	PLS1	0	Repaus, așteptare puls
RES1		IDLE	RES1	RES2	-	0	Reset, nici un puls
PLS1		PLS2	-	RES2	PLS1	1	Apariție puls
RES2		-	RES1	RES2	PLSN	0	Reset, puls există
PLS2		PLS2	RES1	-	PLS1	1	Lipsă puls, ieșire în 1
PLSN		IDLE	-	RES2	PLSN	0	Există puls, ieșirea în 0

Fig. 9.14 Un prim tabel al tranzițiilor

Admitem că starea de repaus, notată cu IDLE, este starea în care semnalele de intrare sunt 0, iar ieșirea Y este tot în 0 logic. La activarea semnalului de reset, $R = 1$, probabil că rămânem în aceeași stare, dar pentru a nu avea două stări stabile pe același rând preferăm să introducem o nouă stare notată cu RES1. Dacă apare un puls, $P = 1$, trecem într-o stare diferită, notată cu PLS1, stare în care avem 1 logic la ieșire.

Dacă suntem în starea prezentă RES1 și semnalul de reset de pe intrare dispare, ne întoarcem înapoi în IDLE. Dacă apare un puls, dar resetul este încă activat, atunci nu se modifică nimic, dar suntem obligați să introducem o nouă stare RES2.

PLS1 este starea în care apare pulsul. Dacă se activează și resetul, trecem în RES2, dar dacă $P = 0$, circuitul ar putea trece în IDLE, dar atunci n-ar fi corectă ieșirea, care trebuie menținută în 1 logic până la apariția unui nou reset. Din acest motiv introducem o nouă stare notată cu PLS2, care va menține ieșirea în 1 logic.

Din starea RES2, dacă semnalul de puls trece în 0 logic, se poate trece în RES1. Dacă același lucru se întâmplă și cu semnalul de reset, iar $P = 1$, ieșirea trebuie să rămână în 0 logic, deci trebuie să introducem starea PLSN. Nu putem trece în PLS1, deoarece ieșirea ar deveni 1 logic în acest caz.

Tabelul rezultat se poate reduce, relativ ușor, prin punerea în evidență a stărilor compatibile și este reprezentat în figura 9.15. Observăm că IDLE și RES1 au aceleași stări următoare și aceleași ieșiri. Deci ele sunt compatibile, generând o singură stare în tabelul redus, stare notată cu IDLE. Stările PLS1 și PLS2 sunt și ele compatibile, din aceleași motive. Starea corespunzătoare din tabelul redus s-a notat cu PLS. În sfârșit, RES2 și PLSN sunt compatibile, fiind înlocuite în tabelul redus prin starea RES.

Urmează în continuare să codificăm stările din tabelul redus. Având 3 stări, codul binar minimal este pe 2 biți, dar de data aceasta codificarea nu mai poate fi întâmplătoare. La sistemele sincrone, chiar dacă alegeam codurile la întâmplare, obțineam un circuit care funcționa foarte bine, singura problemă fiind complexitatea lui, care putea fi mult mai mare decât cea obținută printr-o codificare optimă. Acum însă, suntem obligați să eliminăm cursele critice, dacă vrem ca sistemul să funcționeze conform așteptărilor. Altfel, vom avea o funcționare imprevizibilă a sistemului.

Eliminarea curselor critice presupune de obicei o creștere a numărului de stări. Pentru eliminarea *totală* a curselor, două stări între care au loc tranziții trebuie să primească coduri adiacente, adică coduri care diferă printr-un singur bit. Dacă construim diagrama de adiacență a stărilor, care este dată tot în figura 9.15, observăm că este imposibil să atribuim celor 3 stări coduri adiacente. Singura soluție este să introducem o stare suplimentară, așa cum se poate vedea în figura 9.16. S-a introdus starea RESA, iar tranziția de la PLS la RES se face prin RESA. Deși această tranziție este de două ori mai lentă, acest lucru de obicei nu contează, iar codificarea stărilor propusă în figura 9.16 elimină cursele.

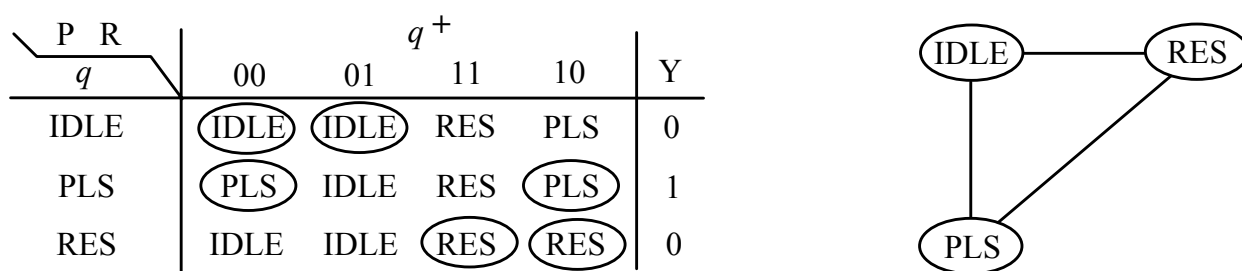


Fig. 9.15 Tabelul redus al stărilor și ieșirii împreună cu diagrama de adiacență a stărilor

P R q	q^+				Y
	00	01	11	10	
IDLE	IDLE	IDLE	RES	PLS	0
PLS	PLS	IDLE	RESA	PLS	1
RESA	—	—	RES	—	—
RES	IDLE	IDLE	RES	RES	0

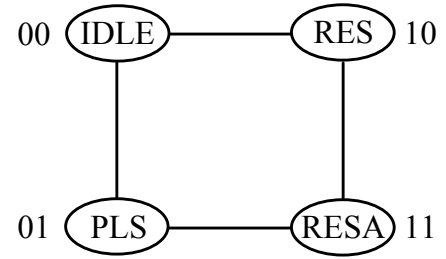


Fig. 9.16 Tabelul fără curse al stărilor și ieșirii și diagrama de adiacență

Sigur că se pot găsi și alte coduri adiacente care elimină cursele. Dacă introducem acum codurile alese în tabelul din figura 9.16, acest nou tabel se poate descompune în 3 diagrame Veitch-Karnaugh, două de câte 4 variabile pentru funcțiile q_1^+ și q_2^+ , și una de două variabile pentru ieșirea Y , care depinde numai de q_1 și q_2 . Ecuațiile care se obțin prin minimizarea celor trei funcții sunt:

$$q_1^+ = P \cdot R + P \cdot q_1$$

$$q_2^+ = q_2 \cdot \bar{R} + \bar{q}_1 \cdot q_2 \cdot P + \bar{q}_1 \cdot P \cdot \bar{R}$$

$$Y = q_2$$

Schema logică a circuitului căutat este dată în figura 9.17. Reamintim că logica combinațională rezultată nu trebuie să aibă hazard static.

În sistemele reale, mai putem avea unele surprize. Dacă pe una din căile logicii combinaționale prin care se propagă semnalul de intrare avem o întârziere prea mare, este posibil să se producă o tranziție eronată la comutarea unei singure linii de intrare. Se spune că avem un **hazard esențial**. Acest tip de hazard ar putea fi detectat pe tabelul tranzițiilor, dacă, pornind dintr-o stare total stabilă, starea total stabilă atinsă după trei modificări succesive ale unei singure variabile de intrare este diferită de starea total stabilă atinsă după o singură modificare a variabilei respective de intrare. Pentru a avea un astfel de hazard, circuitul trebuie să aibă cel puțin 3 stări. Latch-urile nu au hazard esențial, dar circuitul nostru are. Eliminarea hazardului esențial se poate face prin mărirea întârzierilor pe buclele de reacție.

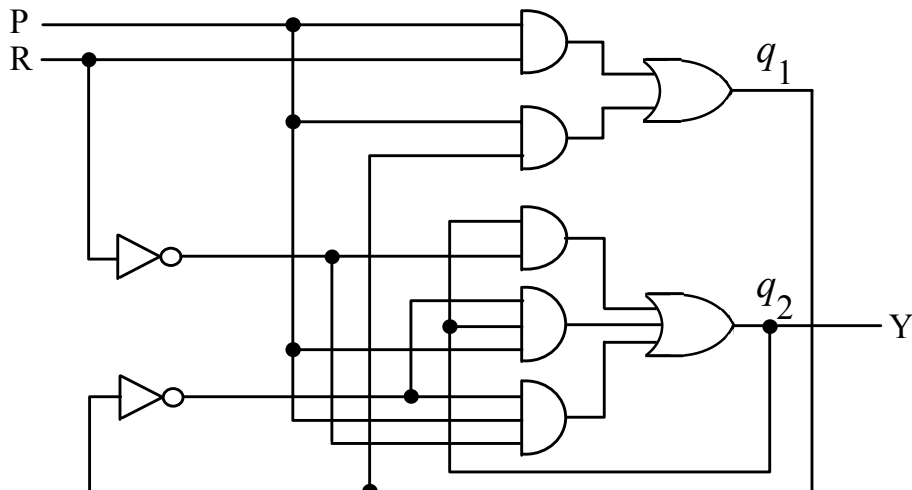
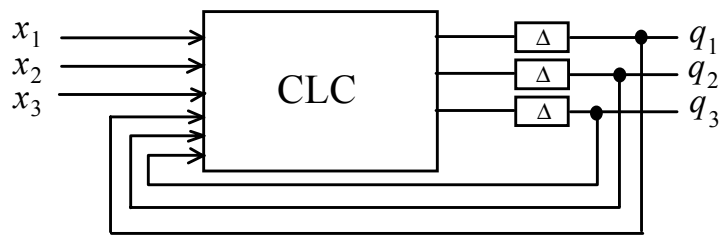


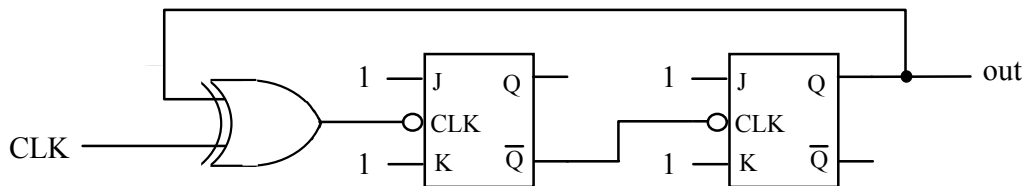
Fig. 9.17 Schema logică a circuitului rezultat

Probleme

- 9.1** Să se proiecteze un circuit basculant cu 3 bucle, unde Δ reprezintă întârzierea proprie circuitelor logice și traseelor de conexiune. În repaus, intrările x_1 , x_2 și x_3 sunt în 0 logic. Atunci când intrarea x_i trece în 1 logic, ieșirea y_i trece în 1 logic dacă era în 0 logic, sau își păstrează starea dacă era în 1 logic. La revenirea intrării x_i în 0 logic, ieșirile y_i își păstrează vechea valoare. Admitem că se modifică o singură intrare, iar comenzile se exclud reciproc, adică $x_1x_2 = x_1x_3 = x_2x_3 = x_1x_2x_3 = 0$.



- 9.2** Să se stabilească cronograma și diagrama stărilor pentru divizorul de frecvență din figură. Arătați care este numărul de stări permanente și raportul de divizare.



- 9.3** Se consideră următoarea expresie booleană:

$$f = (x_1x_2 + \bar{x}_1x_3)(x_1(x_4 + x_5) + \bar{x}_1\bar{x}_3\bar{x}_4x_5)$$

și circuitul combinațional corespunzător.

- există hazard static la modificarea unei singure variabile?
- există hazard dinamic la modificarea unei singure variabile?
- să se implementeze fără hazard funcția din diagrama de mai jos:

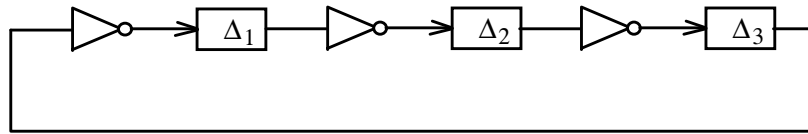
	$\bar{x}_1$	x_1	
$\bar{x}_3$	1	1	1
x_3	1	1	1
	$\bar{x}_2$	x_2	$\bar{x}_2$

([Friedman, 1986])

- 9.4** Să se proiecteze un circuit asincron care are două intrări binare x_1 și x_2 și o ieșire binară y . Pornind din starea stabilă inițială 1, cu $x_1 = x_2 = 0$ și $y = 0$, ieșirea își modifică valoarea dacă și numai dacă x_1 își modifică valoarea de 3 ori la rând sau x_2 își modifică valoarea de 2 ori la rând. Admitem că nu este posibilă o modificare simultană a celor două intrări.

([Friedman, 1986])

- 9.5 Să se analizeze sistemul secvențial din figură, reprezentând cronograma, tabelul tranzițiilor și diagrama stărilor. Arătați că circuitul este un oscilator și stabiliți perioada de oscilație.



([Mange, 1987])

- 9.6 Să se proiecteze un latch cu două intrări de control, C_1 și C_2 , și trei intrări de date D_1 , D_2 și D_3 . Latch-ul este “deschis” numai dacă ambele intrări de control sunt 1, și va memora valoarea logică 1, dacă oricare din intrările de date este pe 1 logic.

([Wakerly, 1990])

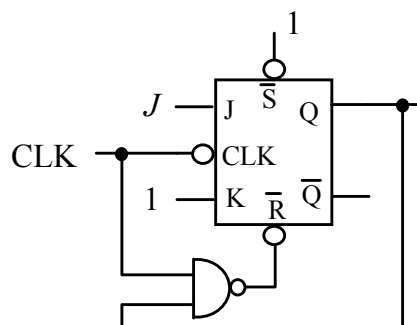
- 9.7 Să se reproiecteze circuitul descris în paragraful 9.4, cu următoarea modificare a caietului de sarcini: circuitul trebuie să detecteze orice modificare apărută pe intrarea P, adică atât o tranziție de la 0 logic la 1 logic, cât și o tranziție de la 1 logic la 0 logic.

([Wakerly, 1990])

- 9.8 Să se proiecteze un bistabil D cu basculare dublă pe front, respectiv care basculează pe ambele fronturi ale semnalului de ceas.

([Wakerly, 1990])

- 9.9 Schema din figură este cea a unui sistem secvențial cvasi-sincron. Admițând că schema bistabilului JK este cea cunoscută, iar poarta ȘI-NU are o întârziere, să se construiască tabelul tranzițiilor și să se analizeze eventualele curse. Pentru ce valori relative ale întârzierilor dispăre orice cursă critică?



([Mange, 1987])

- 9.10 Să se proiecteze un sistem cu două intrări, EN și CLKIN, și o ieșire, CLKOUT, cu următorul comportament. Perioada ceasului este intervalul dintre două fronturi crescătoare ale lui CLKIN. Dacă EN este activat pe durata unei perioade de ceas, atunci CLKOUT este identic cu CLKIN pe durata următoarei perioade de ceas. Dacă $EN = 0$ pe toată perioada, $CLKOUT = 1$ pe toată perioada următoare. Dacă EN are ambele valori logice pe durata unei perioade, CLKOUT va avea valoarea logică complementară celei prezente, pe toată durata perioadei următoare.

([Wakerly, 1990])