

Dan NICULA

ELECTRONICĂ DIGITALĂ

Carte de învățatură 2.0



Editura Universității *TRANSILVANIA* din Brașov
ISBN 978-606-19-0563-8

2015

Lecția 19

Proiectarea sistemelor digitale

19.1 Noțiuni teoretice

Proiectarea sistemelor digitale complexe nu urmează un anumit algoritm.

Categoric, sunt necesare toate cunoștințele despre circuite logice combinaționale și secvențiale.

Este nevoie și de o experiență care să confere o anumită siguranță în gândirea unor structuri potrivite pentru rezolvarea problemelor. Însă, mai este nevoie de ”ceva”.

Acel ”ceva” nu îl poate da nici această lecție. Lecția aceasta poate însă să verifice dacă acel ”ceva” există.

19.2 Pentru cei ce vor doar să promoveze examenul

Nu este necesară parcurgerea acestui capitol.

19.3 Pentru cei ce vor să învețe

1. Proiectați un circuit de secvențiere care comandă un semafor. Circuitul are trei ieșiri active în 1 logic: *verde*, *galben* și *roșu*. Secvența de comandă este următoarea:
 - a) roșu aprins 14 secunde
 - b) roșu și galben aprinse 3 secunde
 - c) verde aprins 12 secunde
 - d) galben aprins 4 secunde

Proiectați circuitul utilizând un numărător cu presetare potrivit și logică adițională. Presupuneți că semnalul de ceas are frecvența de 1 Hz.

Soluție

Semnalul de ceas are frecvența de 1 Hz, deci perioada este de 1 secundă.

- Se poate proiecta un automat sincron cu $14 + 3 + 12 + 4 = 33$ de stări (6 biți de stare) având ieșirile conform specificațiilor.
- O alternativă constă în utilizarea unui numărător modulo 33 (cu 33 de stări, 6 biți de codificare a stării) și generarea ieșirilor din circuite combinaționale conform ecuațiilor:



$ro\acute{s}u = (stare < 17)$
 $galben = (stare \in [14 - 16]) \text{ sau } (stare > 28)$
 $verde = (stare \in [17 - 28])$

Implementarea comparațiilor pentru ieșiri necesită circuite combinaționale cu câte 6 intrări.

- O altă alternativă constă în implementarea unui registru de deplasare de 33 de biți, inițializat cu un singur bit egal cu 1, care la fiecare tact își rotește conținutul. Bitul egal cu 1 desemnează starea curentă. Este, de fapt, o implementare de automat cu codificarea stărilor "one-hot". Circuitul combinațional al ieșirilor va fi format din porți OR care au intrările provenite de la biții asociați stărilor specificate:

$ro\acute{s}u = S_{[0]} + S_{[1]} + \dots + S_{[16]}$
 $galben = S_{[14]} + S_{[15]} + S_{[16]} + S_{[29]} + S_{[30]} + S_{[31]} + S_{[32]}$
 $verde = S_{[17]} + S_{[18]} + \dots + S_{[27]} + S_{[28]}$

Se recomandă plasarea unui set de registre pe ieșiri.

- Acest tip de problemă poate fi abordată și sub forma unui automat care controlează secvențele specificate cu temporizări programabile. Temporizarea este implementată cu un singur numărător presetabil, capabil să numere durata cea mai lungă (14 secunde, 4 biți). Figura 19.1 prezintă schema de principiu a circuitului proiectat. Secvențierea se realizează cu un registru de rotire de 4 biți activat când numărătorul de temporizare ajunge la 0. Tot atunci, numărătorul se încarcă cu noua valoare a temporizării, determinată de secvența curentă.

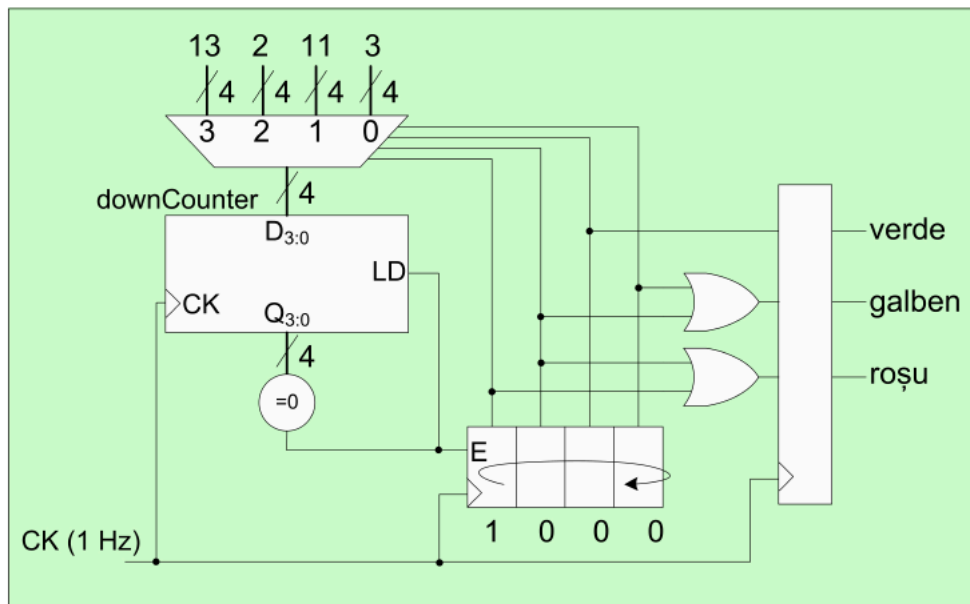


Figura 19.1 Schema de principiu a semaforului descris la problema 1, implementat cu un registru de deplasare asociat celor 4 secvențe și un numărător presetabil pentru temporizare.

- Proiectați un circuit care să genereze secvența periodică prezentată în figura 19.17-a.
 - folosind un numărător și un decodificator;
 - folosind bistabile D.

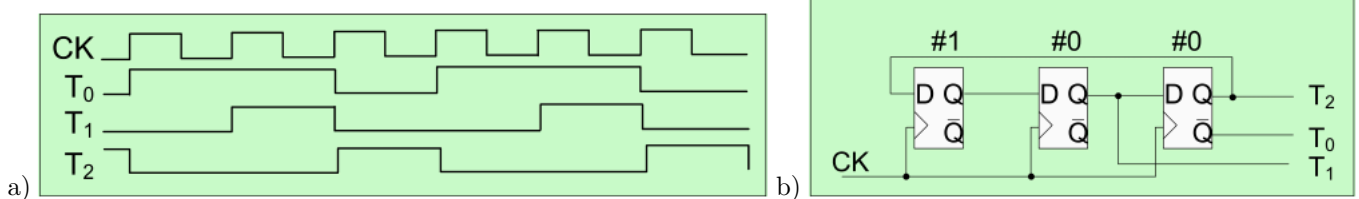


Figura 19.2 a) Forme de undă referite de problema 2. b) Circuitul.



Soluție

Se observă pe formele de undă o secvență ciclică de 3 stări. Automatul se poate implementa cu 2 bistabile.

a) Implementarea cu numărător necesită un numărător de 2 biți care numără între stările 0 și 2. Cei 2 biți ai numărătorului se conectează la intrarea unui decodificator 2:4. Ieșirile T_2, T_1, T_0 se obțin din ieșirile decodificatorului astfel:

$$T_0 = O_0 + O_1$$

$$T_1 = O_1$$

$$T_2 = O_2$$

b) O alternativă de implementare pornește de la observația că T_1 se poate obține printr-un registru de 3 biți, inițializat cu 100, care se rotește. Se observă că T_2 se poate genera prin întârzierea semnalului T_1 cu un tact, iar T_0 este semnalul negat al lui T_1 . Circuitul este prezentat în figura 19.17-b.

3. Proiectați un automat sincron cu o intrare X și o ieșire Z. Ieșirea este activată în 1 logic pentru o perioadă de ceas la apariția fiecărei secvențe seriale 0100, atâta timp cât nu a apărut secvența 1000. Ca exemplu, figura 19.3 prezintă două cazuri de secvențe X și Z (observați poziția unde se activează ieșirea Z). Modificați circuitul astfel încât să reia detectarea secvenței 0100 după primirea secvenței 0111.

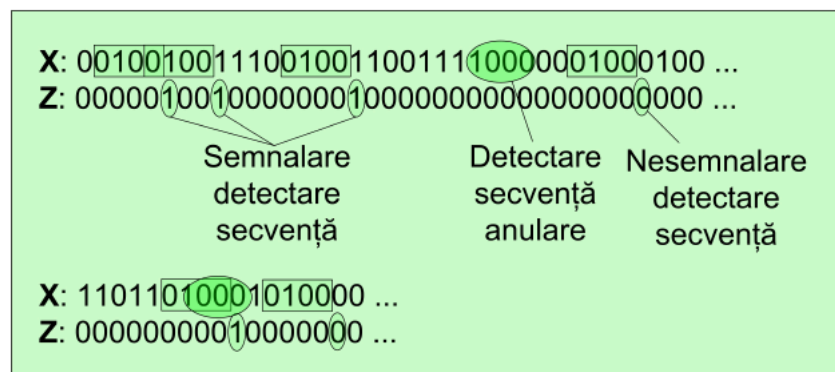


Figura 19.3 Forme de undă pentru automatul de detectare a unei secvențe condiționat de apariția alteia, problema 3.

Soluție

Detectarea apariției unei secvențe seriale se face prin întârzierea valorii de intrare prin bistabile și colectarea ieșirilor bistabilelor (valorile intrării la momente de timp succesive) printr-o poartă AND cu intrări negate sau ne-negate (conform secvenței). Un asemenea circuit detectează apariția secvenței 1000. La apariția acesteia se setează valoarea unui bistabil RS, având semnificația "secvența 1000 a fost primită". Un circuit similar detectează apariția secvenței 0100 și condiționează semnalizarea pe ieșire de starea bistabilului RS. Circuitul este prezentat în figură 19.4.

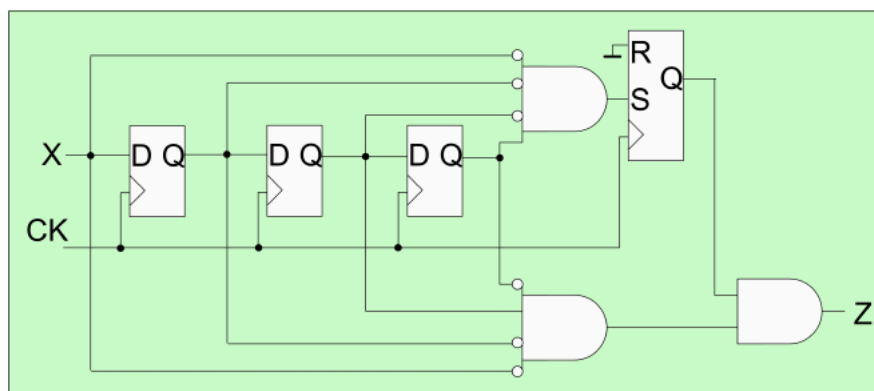


Figura 19.4 Automatul de detectare a unei secvențe condiționat de apariția alteia, problema 3.



4. Într-un sistem de comunicație serială, un mesaj poate începe fie cu secvența 11 fie cu secvența 1001. Să se realizeze un circuit secvențial sincron pentru detectarea secvențelor de început de mesaj. Intrarea se numește DATA iar singura ieșire se numește GĂSIT. Când RESET = 1, automatul intră în starea inițială și GĂSIT = 0. Când RESET = 0, automatul este în mod normal de operare. Circuitul semnalizează cu GĂSIT=1 după detectarea primei secvențe 11 sau 1001. Ieșirea GĂSIT revine la 0 doar după ce RESET devine 1 din nou. Proiectați un automat Moore care să implementeze circuitul de recunoaștere a secvențelor de început. Deduceți tabelul de tranziții al automatului și precizați semnificația fiecărei stări. Nu se vor utiliza mai mult de 5 stări.

Soluție

Circuitul este realizat cu un registru de deplasare cu intrarea DATA care permite testarea succesiunii de 4 valori ale intrării. Circuitele de detectare a secvențelor de început de mesaj se realizează cu porți AND. La detectarea uneia din secvențe se setează valoarea unui bistabil RS. Ieșirea GĂSIT este conectată la ieșirea acestui bistabil. Intrarea de RESET este conectată la intrarea de reset asincron a tuturor bistabilelor. Circuitul este prezentat în figura 19.5-a.

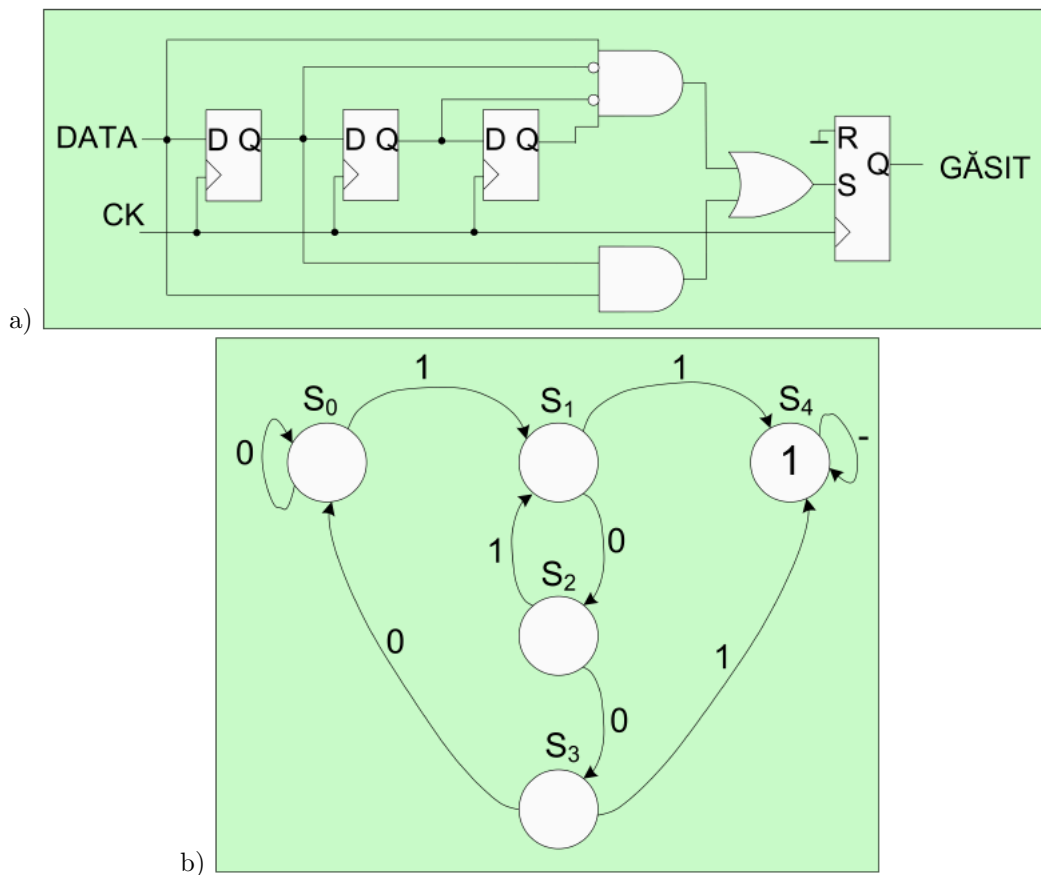


Figura 19.5 a) Automatul descris la problema 4. b) Graful de tranziții al automatului Moore.

Automatul Moore asociat are graficul prezentat în figura 19.5-b. Stările au următoarele semnificații:

S_0 - stare inițială, nu s-a primit nimic din secvența de intrare;

S_1 - s-a primit un 1;

S_2 - s-a primit o succesiune 10;

S_3 - s-a primit o succesiune 100;

S_4 - s-a primit o succesiune 11 sau 1001 și se activează ieșirea; această stare nu se mai părăsește decât la activarea reset-ului asincron.

5. Proiectați un numărător modulo 5 sau modulo 12, utilizând un numărător de 4 biți presetabil și logică adițională. Valoarea este selectată cu ajutorul unui bit de intrare.

Soluție

Se optează pentru numărarea în sens descrescător.

Pentru $SEL = 1$, se realizează numărarea modulo 12, între 11 și 0.



Pentru $SEL = 0$, se realizează numărarea modulo 5, între 4 și 0. La atingerea stării 0000, se activează intrarea de încărcare a număratorului presetabil. Valoarea cu care se încarcă numărătorul este selectată cu un multiplexor pe baza intrării SEL . Circuitul este prezentat în figura 19.6.

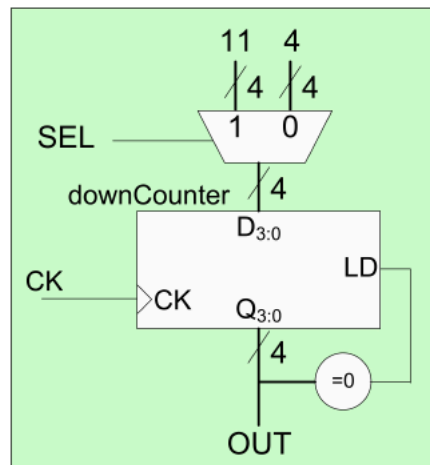


Figura 19.6 Numărător modulo 12 sau modulo 5, problema 5.

6. Proiectați un circuit de întârziere a unui semnal cu 1, 2, 3 sau 4 perioade de ceas. Durata întârzierii este selectată cu o intrare de 2 biți.

Soluție

Pentru selectarea duratei întârzierii se poate folosi un multiplexor 4:1 cu cei 2 biți de intrare conectați la intrările de selecție. Întârzierile semnalului de intrare sunt realizate cu ajutorul a 4 bistabile D conectate în serie ca în figura 19.7-a. De observat că circuitul din figura 19.7-b prezintă o temporizare similară, însă ieșirea acestuia este direct din registru (nu din circuit combinațional). Varianta **b)** este recomandată.

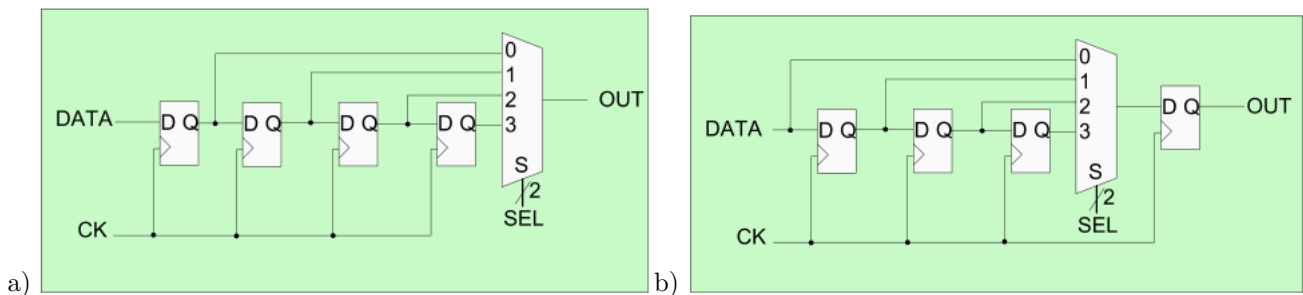


Figura 19.7 Întârzieri variabile pentru un semnal, problema 6.

7. Proiectați un circuit secvențial care să efectueze următoarele operații:

S_1	S_0	$Q_{[3:0]}^+$	Acțiune
0	0	Q	Păstrează starea
0	1	0000	Inițializează starea
1	0	$Q \oplus D_{in}$	XOR între starea curentă și intrarea de 4 biți
1	1	$Q + D_{in}$	Adună intrarea la starea curentă

Soluție

Circuitul se bazează pe un registru de 4 biți peste care se închide o buclă ce determină starea viitoare a registrului, conform tabelului, pe baza intrărilor de selecție. Figura 19.8 prezintă circuitul propus.

8. Proiectați un circuit de conversie a numerelor reprezentate în complement față de 2. Numărul este încărcat în paralel într-un registru și apoi deplasat serial. La ieșire, numărul este convertit și încărcat serial într-un registru de destinație de unde este preluat în paralel în momentul activării unui semnal.



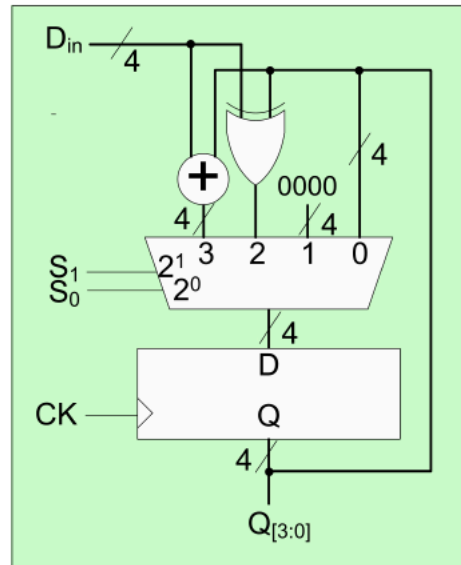


Figura 19.8 Circuitul proiectat la problema 7.

9. Proiectați un circuit secvențial sincron care să convertească un număr reprezentat în complement față de 2 în numărul corespunzător cu semn schimbat. Numărul este introdus la intrarea circuitului câte un bit pe tact, începând cu bitul cel mai puțin semnificativ. Apariția numărului pe intrarea de date (D_{in}) este marcată prin activarea intrării de validare ($En = 1$). Între două numere consecutive există o pauză de minim un tact (perioadă în care $En = 0$). Numărul cu semn schimbat este prezentat secvențial pe ieșirea $Dout$, începând cu următorul tact după primirea celui mai puțin semnificativ bit (primul). $Dout = 0$ pe durata când $En = 0$. Se cer organigrama și schemă bloc a circuitului propus.
10. Proiectați un circuit de verificare a protocolului cerere/confirmare. La apariția unei erori de protocol se setează un bit de eroare. Acesta se menține activ până la activarea semnalului de reset asincron. Protocolul de comunicare are următoarele reguli:
 - Modulul client activează o cerere prin setarea ieșirii $Rq = 1$ (Engl. "Request").
 - Modulul server observă cererea și când o satisface informează clientul prin $Ack = 1$ (Engl. "Acknowledge"). Ieșirea Ack este activă un singur tact, întotdeauna ca efect al unei cereri.
 - Imediat ce observă confirmarea, modulul client inactivează cererea $Rq = 0$.
 Abaterile de la protocol sunt considerate "erori de protocol" și determină activarea $ERR = 1$.
11. Protocolul USB implementează o codare/decodare NRZI (Engl. "Non-Return to Zero Inverted") asupra datelor seriale trimise în pachete. În codarea NRZI, un "1" este reprezentat de o menținere a valorii logice, iar un "0" este reprezentat de o modificare a valorii logice. Un șir de "0" produce la ieșire un semnal care comută la fiecare tact. Un șir de "1" produce la ieșire un semnal constant (fără tranziții). Figura 19.9 prezintă un flux de date de intrare și codarea NRZI asociată acestora.

Să se proiecteze circuite de codare și decodare NRZI.

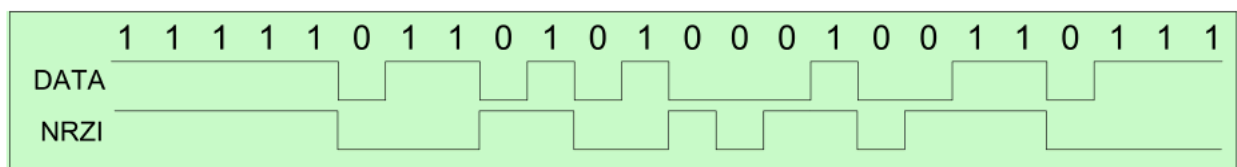


Figura 19.9 Date codate NRZI (protocol USB).

Soluție

Pe formele de undă din figura 19.9 se observă că ieșirea $NRZI$ reacționează combinațional la valoarea intrării $DATA$. Din enunț reiese că valoarea curentă a ieșirii $NRZI$ depinde de valoarea curentă a intrării $DATA$ și de valoarea anterioară a $NRZI$. Valoarea anterioară a $NRZI$, denumită $NRZI_{del}$ se obține printr-un bistabil D.



Conform enunțului, se obține ecuația ieșirii:

$$NRZI = \overline{DATA} \oplus NRZI_{del}$$

Pentru decodare, se deduce ecuația:

$$DATA = \overline{NRZI} \oplus NRZI_{del}$$

Circuitele de codare și de decodare a datelor NRZI sunt prezentate în figura 19.10.



Figura 19.10 Circuite de codare și decodare NRZI.

12. Pentru a asigura tranziții adecvate ale semnalului, protocolul USB realizează o prelucrare a fluxului de date seriale denumită *"bit stuffing"*. Algoritmul presupune inserarea unei valori "0" după fiecare 6 valori consecutive egale cu "1". La recepție, bitul inserat suplimentar trebuie detectat și descărcat din fluxul de date seriale. Figura 19.11 prezintă un flux de date de intrare procesat pe baza algoritmului *"bit stuffing"* la transmitere și la recepție.

Să se proiecteze circuite utilizate pentru *"bit stuffing"* la transmitere și la recepție.

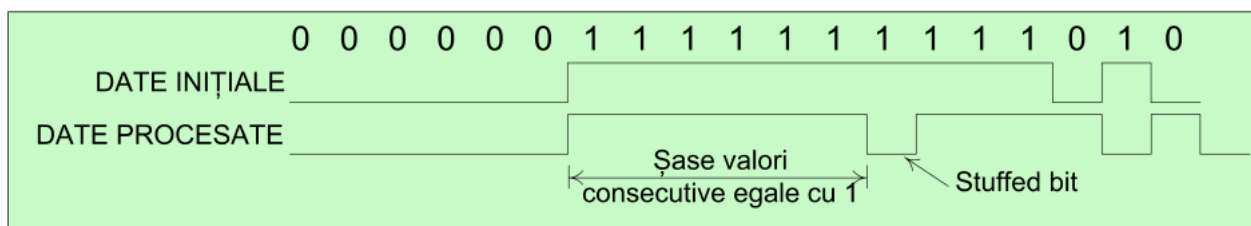


Figura 19.11 Data prelucrate *"bit stuffing"* (protocol USB).

13. Proiectați un automat care semnalează pe ieșire existența pe intrare a 3 sau mai mulți biți secvențiali și consecutivi egali cu 1.

Soluție

Dacă intrarea se întârzie printr-un registru de deplasare de 3 biți, la ieșirile bistabilelor se pot accesa simultan valorile intrării în ultimele 3 tacte. Cu o poartă AND cu 3 intrări se poate implementa condiția impusă. Circuitul este prezentat în figura 19.12. Se remarcă faptul că pentru întârziere se pot folosi și numai două bistabile, caz în care ar trebui folosită valoarea intrării. În acest caz, ieșirea ar fi activată combinațional (simultan cu al treilea bit egal cu 1 la intrare). Bistabilul plasat pe ieșire are, de asemenea, rolul de a genera ieșirea circuitului din registru.

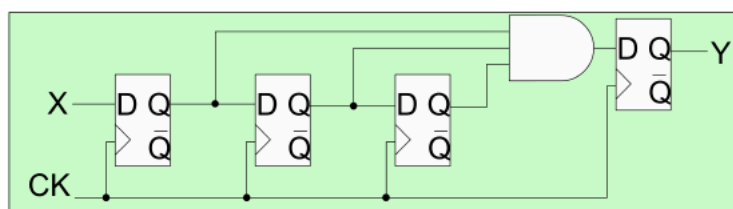


Figura 19.12 Automat care depistează o succesiune de 3 sau mai mulți biți succesivi egali cu 1.

Dacă la poarta AND se pun inversoare pe anumite intrări, se poate modifica secvența depistată. De exemplu, prin inversarea logică a intrării din mijloc, se obține un circuit care semnalează apariția pe intrare a succesiunii de stări 101.

14. Proiectați un automat care semnalează pe ieșire existența pe cele două intrări a 3 sau mai multe coincidențe secvențiale și consecutive ale celor două intrări.

Soluție

Automatul se poate implementa ca și cel din problema 13 cu adăugarea unei porți XNOR pe cele două intrări. Ideea se bazează pe observația că poarta XOR cu două intrări semnalează la ieșire faptul că cele două intrări sunt diferite, iar poarta XNOR își activează ieșirea în 1 în cazul egalității celor două intrări. Verificarea coincidenței celor două intrări este echivalentă cu verificarea ieșirii egale cu 1 pentru poarta XNOR.

15. Proiectați un automat care semnalează pe ieșire existența pe intrare a 3 sau mai mulți biți secvențiali și consecutivi egali.

Soluție

Biții de intrare se întârzie prin bistabile D pentru a avea acces la 3 biți succesivi. Se pune condiția ca 3 biți consecutivi să fie toți egali cu 1 sau toți egali cu 0. Rezultă circuitul prezentat în figura 19.13.

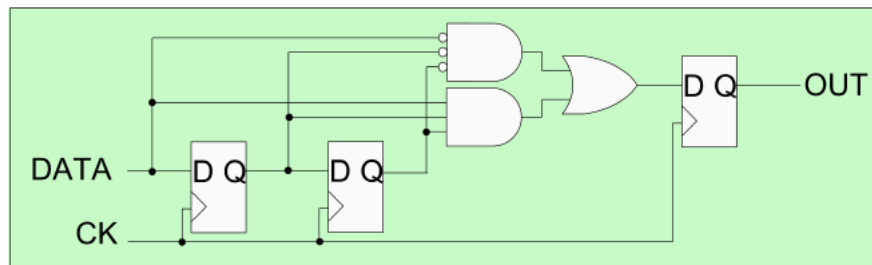


Figura 19.13 Circuit care semnalează 3 sau mai mulți biți consecutivi egali.

16. Proiectați un automat care semnalează pe ieșire prezența a 5 concidențe dintre datele primite pe cele două intrări.
17. Proiectați un automat care determină și semnalează pe ieșire valoarea majoritară a valorilor primite pe intrare în ultimele 3 perioade de ceas. Determinați secvența de ieșire în cazul aplicării la intrare a secvenței 00111110001100101010100111.
18. Proiectați un automat care semnalează pe ieșire detectarea unui șir de biți aplicați secvențial pe intrare cu valorile 10111. Pattern-ul semnalat se poate suprapune peste o instanță anterioară. Figura 19.14 prezintă un exemplu de intrare X și ieșirea Z asociată.

X: 01010111001011001011101111001...
Z: 00000000100000000000010001000...

Figura 19.14 Exemplu de forme de undă pentru automatul care depistează secvența 10111.

Rezolvați aceeași problemă pentru secvențele: 11011, 00111, 01011.

Propuneți un circuit care permite particularizarea ulterior a secvenței depistate de lungime 5 biți.

Propuneți un circuit care permite particularizarea ulterior a secvenței depistate de lungime maximă 5 biți (poate fi și mai scurtă).

19. Proiectați un automat cu două intrări și o ieșire având următoarea funcționare:
- Pe una din intrări se primesc grupuri de câte 4 biți, cu bitul cel mai semnificativ primul.
 - Pe a doua intrare se primesc pulsuri de câte o singură perioadă de ceas, nu mai dese decât un puls la 4 perioade de ceas. A doua intrare marchează momentul când pe prima intrare există primul bit din grupul de 4 biți.
 - Între două grupuri de 4 biți pot fi și pauze. Pe durata pauzei, valorile primei intrări sunt ignorate.
 - Ieșirea este activată dacă la finalul unei secvențe de 4 biți validă se constată că secvența nu reprezintă cifre în format BCD (între 0 și 9).



20. Proiectați un circuit generator de 8 faze.

Soluție

O soluție este reprezentată de un numărător de 3 biți, urmat de un DCD 3:8. Cele 8 faze se obțin la ieșirile decodificatorului. Pe aceste ieșiri se pot adăuga bistabile. Frecvența maximă de lucru este determinată de timpul de propagare al decodificatorului.

O alternativă mai bună o constituie un registru de deplasare de 8 biți în inel inițializat cu valoarea 0000_0001. La fiecare tact, conținutul registrului se rotește și activează o altă fază. Ieșirile sunt direct din bistabile iar circuit combinațional nu există.

21. Proiectați o unitate aritmetică pe N biți care, pe baza intrării de selecție S și a intrării de transport C_i , implementează cele 4 funcții din tabel:

S	C_i	Funcție
0	0	$F = A + B$ (sumă)
0	1	$F = A + 1$ (increment)
1	0	$F = A - 1$ (decrement)
1	1	$F = A + \overline{B} + 1$ (diferență)

19.4 Pentru cei ce vor să devină profesioniști

1. Implementați cu circuit ROM automatul "TAP Controller" (Engl. "Test Access Port Controller") al cărui graf de tranziții este prezentat în figura 19.15.

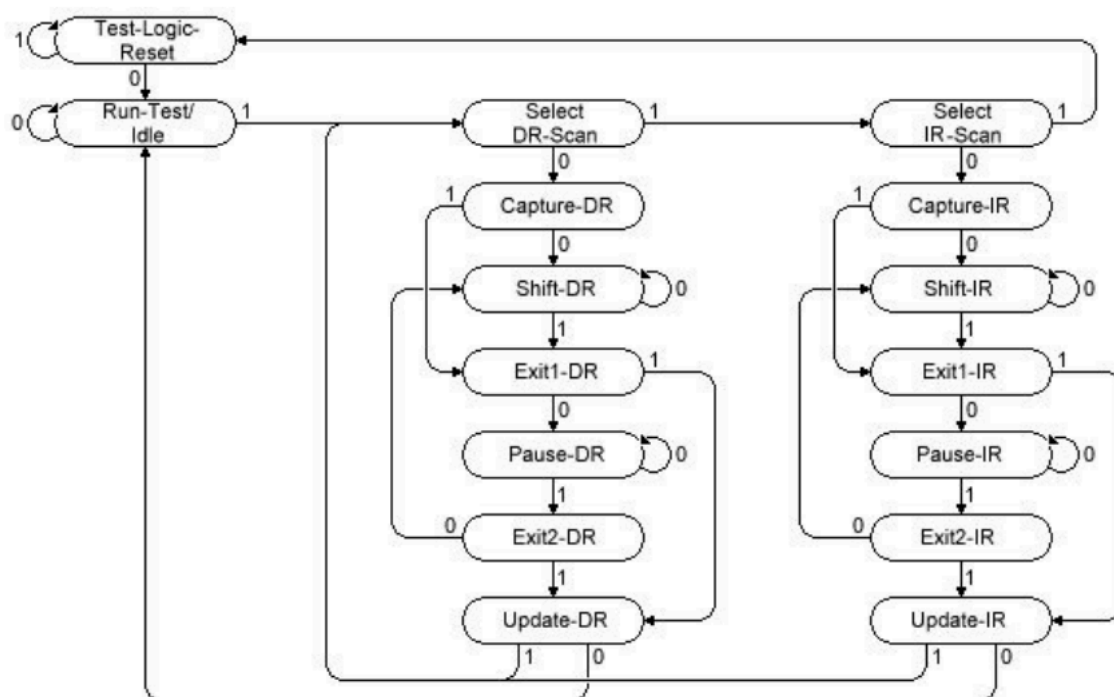


Figura 19.15 Graf de tranziții pentru TAP Controller.

2. Să se proiecteze un circuit secvențial de negare a numerelor reprezentate în complement față de 2, primite la intrare secvențial, cel mai puțin semnificativ bit primul. Adică, dacă la intrare se primește 5, la ieșire apare -5, dacă se primește -6, la ieșire apare 6. O intrare suplimentară se activează simultan cu ultimul bit al datelor de intrare (cel mai semnificativ). O ieșire suplimentară se activează simultan cu primul bit al rezultatului (cel mai puțin semnificativ). Datele seriale sunt prezentate la intrare într-un flux continuu.

Exemplu:



- axa timpului este de la stânga la dreapta
- numerele sunt codificate în complement față de 2, primul bit primit de sistem este cel mai puțin semnificativ (reprezentat în partea stângă a axei temporale)
- $DataIn = 5|_{10} = 00_0101|_2$. Biții se vor primi unul câte unul, în ordinea: 101000 (primul bit transmis, aflat în partea stângă a reprezentării temporale, este cel mai puțin semnificativ bit, aflat în partea dreaptă a scrierii binare).
- $DataIn = -8|_{10} = 1111_1000|_2$. Biții se vor primi unul câte unul, în ordinea: 00011111.

```

DataIn      10100000011111x
MarkerIn    00000100000001x
DataOut      x11011100010000
MarkerOut    010000010000000

```

3. Să se proiecteze un circuit digital sincron care să detecteze apăsarea unui buton. Circuitul generează două semnale:

1. *butonPush*: un puls la fiecare apăsare pe buton (lățimea unui puls este egală cu o perioadă de ceas),
2. *butonAutoRepeat*: un tren de impulsuri pentru fiecare apăsare pe buton care durează mai mult de o secundă (lățimea unui puls este egală cu o perioadă de ceas, intervalul între două pulsuri succesive este de 100 ms).

Schema bloc a sistemului proiectat este prezentată în figura 19.16.

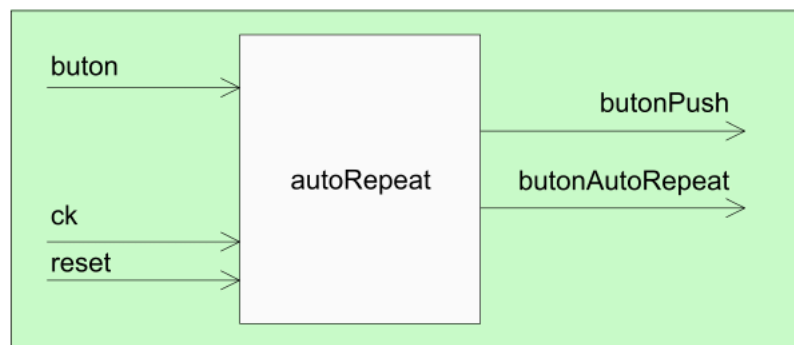


Figura 19.16 Schema bloc a circuitului de generare de pulsuri (problema 3).

Tabel cu descrierea porturilor:

Nume Port	Direcție	Dimensiune	Descriere port
ck	IN	1	Semnal de ceas cu frecvența de 50MHz.
reset	IN	1	Semnal de reset asincron, activ pe 1.
buton	IN	1	Intrare provenită de la un buton cu apăsare, activă 1. Se consideră că intrarea este deja filtrată și toate zgomotele mecanice eliminate.
butonPush	OUT	1	Ieșire pe care apare un puls (lățime o perioadă de ceas) la fiecare apăsare pe buton.
butonAutoRepeat	OUT	1	Ieșire pe care apar pulsuri repetitive (lățime o perioadă de ceas, interval între două pulsuri de 100 ms) în cazul în care butonul este ținut apăsat mai mult de 1 secundă.

Sistemul este sincron, având semnalul de ceas **ck** (frecvență de 50MHz și factor de umplere 50%) și semnalul asincron de inițializare **reset**. Intrarea **buton** provine de la un buton cu acționare manuală. Se consideră că intrarea este deja filtrată (**buton** = 1 semnifică "buton apăsat", **buton** = 0 semnifică "buton relaxat").

O apăsare pe buton nu poate fi mai scurtă decât 100 ms. La fiecare apăsare pe buton, indiferent de durata acesteia, se generează un puls pe ieșirea **butonPush** (lățime o perioadă de tact).

Dacă apăsarea pe buton este mai lungă decât 1 secundă, circuitul generează pe ieșirea **butonAutoRepeat** un șir repetitiv de impulsuri, cu cadența de 10 pulsuri pe secundă (lățimea unui puls este o perioadă de tact, intervalul între două pulsuri este de 100 ms).

Forme de undă asociate sistemului de generare de pulsuri sunt prezentate în figura 19.17.



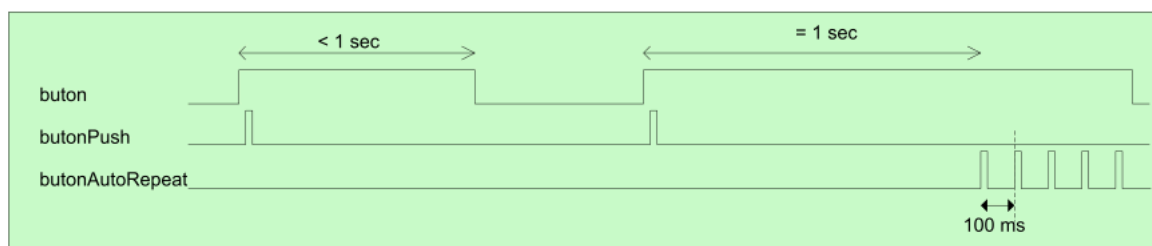


Figura 19.17 Forme de undă asociate sistemului de generare de pulsuri (problema 3).

