

Dan NICULA

ELECTRONICĂ DIGITALĂ
Carte de învățătură 2.0



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

Lecția 20

Subiecte de examen

20.1 Subiecte 2012 - 1

1. Întrebări [2pct]

- [0.2] Definiți noțiunea de "multiplexor 2:1".

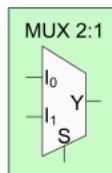
Soluție

Multiplexorul 2:1 este un circuit logic combinațional cu o intrare de selecție, două intrări de date și o ieșire. Pe baza intrării de selecție se decide care din cele două intrări de date se transmite la ieșire.

Ecuatia logică a ieșirii multiplexorului 2:1 este:

$$Y = S \cdot I_1 + \bar{S} \cdot I_0$$

Simbolul multiplexorul 2:1 este:



- [0.2] Definiți noțiunea de "bistabil D".

Soluție

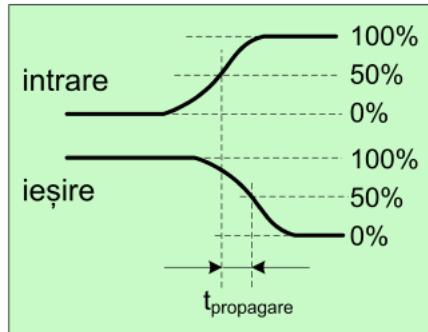
Bistabilul D este un circuit logic secvențial care prezintă două stări stabile. Comutarea bistabilului se face doar în momentele fronturilor active ale semnalului de ceas. La fiecare front activ al semnalului de ceas bistabilul D eșantionează intrarea și își păstrează starea până la următorul front activ al semnalului de ceas. Tabelul de determinare a stării viitoare a bistabilului D și simbolul acestuia sunt:

D	Q^+	Acțiune	DFF
0	0	copiază intrarea	
1	1	copiază intrarea	

- [0.2] Definiți noțiunea de "timp de propagare a unei porți logice".

Soluție

Timpul de propagare a unei porți logice este intervalul de timp necesar propagării unui semnal între intrarea și ieșirea unei porți logice. Timpul de propagare se măsoară la nivelul de 50% al semnalelor de intrare și de ieșire.



- [0.2] Definiți noțiunea de "LFSR - Linear Feedback Shift Register".

- [0.2] Definiți noțiunea de "organigramă".

- [0.5] Justificați valoarea de adevăr a expresiei:

$$\overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot C + A \cdot B \cdot D = \overline{A} \cdot C + A \cdot B \cdot C \cdot D + A \cdot B \cdot \overline{C} \cdot D + A \cdot B \cdot D$$

- [0.5] Reprezentați operația prezentată cu numere în baza zece, cu operanzi reprezentați în complement față de 2, pe 16 biți.

$$2012 + (-12) = 2000|_{10}$$

2. Circuite logice combinaționale [2pct]

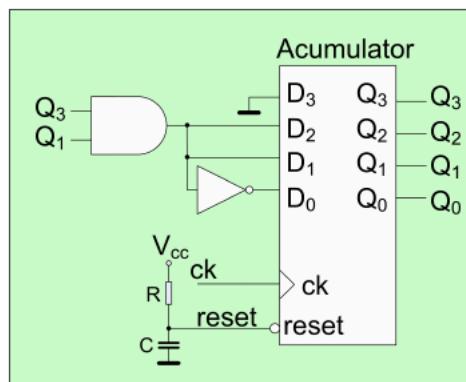
Se consideră funcția:

$$F = \sum(6, 8, 11, 12, 14, 15) + d(2, 3)$$

- [0.5] Minimizați funcția logică și implementați-o cu porți NAND.
- [0.5] Evaluati "costul" implementării.
- [0.5] Propuneți o soluție de eliminare a hazardului combinațional și evaluați costul suplimentar al acesteia.
- [0.5] Implementați funcția cu MUX 8:1 și un număr minim de porți logice suplimentare.

3. Analiza circuitelor logice secvențiale [2pct]

Se consideră circuitul:



La fiecare front crescător de ceas, registrul accumulator pe 4 biți acumulează valoarea de pe intrare. Adică, dacă starea curentă este Q și pe intrare valoarea este D atunci starea viitoare va fi $Q^+ = Q + D$. Funcționarea accumulatorului este descrisă de codul Verilog:

```
input [3:0]      D;          // intrarea de date (4 biti)
output [3:0]     Q;          // iesirea accumulatorului (4 biti)
always @ (posedge ck or negedge reset)
if (!reset) Q <= 4'b0000; else
    Q <= Q + D;           // initializare (reset activ "0")
                           // acumularea intrarii
```

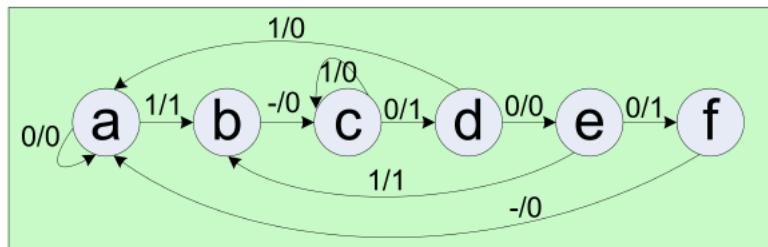
- [0.5] Explicați în limbaj natural funcționarea circuitului.
- [1.0] Reprezentați grafic comportamentului circuitului.



- [0.5] Ce rol are condensatorul prezent în schemă?

4. Sinteza circuitelor logice secvențiale [2pct]

Se consideră graful de tranziții:

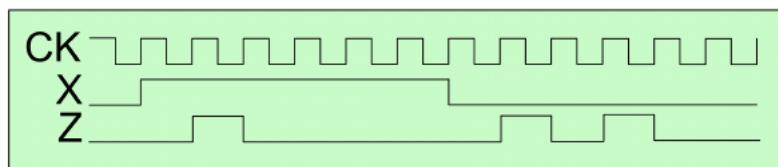


- [2.0] Implementați automatul cu bistabile T și porți NAND.

5. Proiectare sistem digital [1pct]

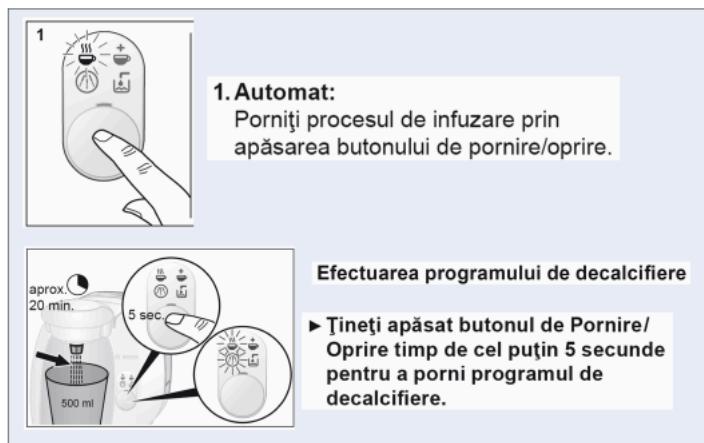
- [1.0] Proiectați un sistem secvențial sincron cu o intrare X și o ieșire Z. La fiecare front pozitiv al semnalului X apare un puls pe ieșirea Z. La fiecare front negativ al semnalului X apare o secvență de două pulsuri pe ieșirea Z.

Formele de undă și temporizarea sunt prezentate în figură.



6. Bonus [2pct]

Manualul de utilizare a automatului de cafea  specifică următoarele:



Proiectați un sistem digital secvențial sincron, având frecvența de ceas de 100 MHz, care preia intrarea de la **butonul pornire/oprire** și generează **două impulsuri** (cu întârziere o perioadă de ceas) pentru începerea celor două funcții: *automat* și *decalcifiere*.



20.2 Subiect 2012 - 2

1. Întrebări [2pct]

- [0.2] Definiți noțiunea de "demultiplexor 1:2".

Soluție

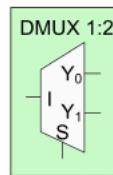
Demultiplexorul 1:2 este un circuit logic combinațional cu o intrare de selecție, o intrare de date și două ieșiri. Pe baza intrării de selecție se decide pe care din cele două ieșiri de date se transmite intrarea.

Ecuatiile logice ale ieșirilor demultiplexorului 1:2 sunt:

$$Y_1 = S \cdot I$$

$$Y_0 = \overline{S} \cdot I$$

Simbolul demultiplexorul 1:2 este:

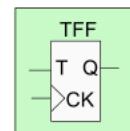


- [0.2] Definiți noțiunea de "bistabil T".

Soluție

Bistabilul T este un circuit logic secvențial care memorează starea, activ pe frontul semnalului de ceas. În momentul frontului activ al semnalului de ceas, dacă $T = 1$, bistabilul își complementează starea, iar dacă $T = 0$, bistabilul își păstrează starea. În rest, pe o perioadă de semnal de ceas, până la următorul front activ al acestuia, bistabilul memorează starea. Tabelul de determinare a stării viitoare a bistabilului T și simbolul acestuia sunt:

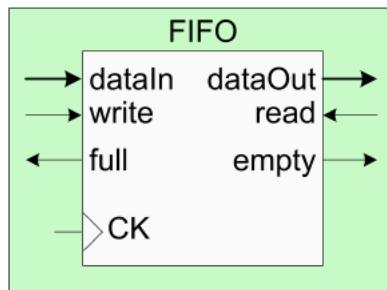
T	Q^+	Acțiune	
		0	1
0	Q	păstrează starea	
1	\overline{Q}	complementează starea	



- [0.2] Definiți noțiunea de "memorie FIFO".

Soluție

Memoria FIFO (Engl. "First In First Out") este o formă de organizare a memoriei fără adrese, accesul la date făcându-se după regula "prima dată scrisă este prima dată citită". Memoriile FIFO sunt utilizate pentru implementarea unor cozi de date. Organizarea memoriei FIFO constă într-o memorie RAM și pointere pentru adrese de scriere și citire implementate cu numărătoare. Simbolul bloc al unei memorii FIFO este:



Se remarcă prezența unor ieșiri care indică starea memoriei:

full = memorie plină, nu se mai acceptă scrieri;

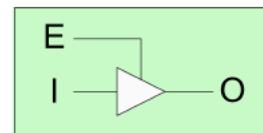
empty = memorie goală, nu se mai acceptă citiri.

- [0.2] Definiți noțiunea de "poartă logică în 3 stări".

Soluție

Poarta logică în 3 stări prezintă, pe lângă cele două stări binare (0 și 1), o stare caracterizată printr-o impedanță de ieșire mare, ideal infinită. Ieșirea portii logice în stare de înaltă impedanță, simbolizată HiZ (Engl. "High Z") poate fi conectată la ieșirea altrei porți logice, fără a afecta funcționalitatea acesteia. Poarta în 3 stări are două intrări: o *intrare de date* și o *intrare de validare*. Tabelul de funcționare și simbolul unei porți în 3 stări sunt:

E	I	O
0	-	HiZ
1	0	0
1	1	1



- [0.2] Definiți noțiunea de "automat Mealy".

Soluție

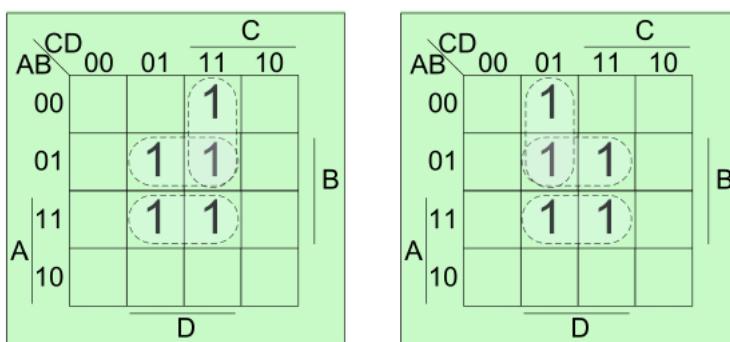
Automatul Mealy este un automat la care ieșirea depinde atât de starea curentă cât și de intrări.

- [0.5] Justificați valoarea de adevăr a expresiei:

$$A \cdot B \cdot D + \bar{A} \cdot B \cdot D + \bar{A} \cdot C \cdot D = A \cdot B \cdot D + \bar{A} \cdot B \cdot D + \bar{A} \cdot \bar{C} \cdot D$$

Soluție

Se construiesc diagramele V-K asociate expresiilor din stânga și din dreapta.



Se observă că expresia este falsă:

$$\sum(3, 5, 7, 13, 15) \neq \sum(2, 5, 7, 13, 15)$$

- [0.5] Reprezentați operația prezentată cu numere în baza zece, cu operanzi reprezentați în complement față de 2, pe 16 biți.

$$(-2012) + 12 = (-2000)|_{10}$$

Soluție

$$2012|_{10} = 1024 + 512 + 256 + 128 + 64 + 16 + 8 + 4 = 111_1101_1100|_2 = 0000_0111_1101_1100|_2$$

$$-2012|_{10} = 1111_1000_0010_0100|_2$$

$$12|_{10} = 0000_0000_0000_1100|_2$$

$$2000|_{10} = 0000_0111_1101_0000|_2$$

$$-2000|_{10} = 1111_1000_0011_0000|_2$$

$$\begin{array}{r}
 1111_1000_0010_0100+ \\
 0000_0000_0000_1100 \\
 \hline
 1111_1000_0011_0000
 \end{array}$$

2. Circuite logice combinaționale [2pct]

Se consideră funcția:

$$F = \sum(2, 3, 6, 8, 10) + d(5, 7)$$

- [0.5] Minimizați funcția logică și implementați-o cu porti NAND.
- [0.5] Evaluati "costul" implementării.
- [0.5] Propuneți o soluție de eliminare a hazardului combinațional și evaluați costul suplimentar al acestieia.
- [0.5] Implementați funcția cu MUX 8:1 și un număr minim de porți logice suplimentare.

Soluție

Diagrama V-K:

		CD		C		
		00	01	11	10	
AB	00	-	-	1	1	B
	01	-	-	1		
A	11	-	-	1		
	10	1			1	D

Expresia minimizată: $F(A, B, C, D) = \bar{A} \cdot C + A \cdot \bar{B} \cdot \bar{D}$

Necesar de porti logice:

2 NAND x 2 = 4 intrări

1 NAND x 3 = 3 intrări

3 NOT x 1 = 3 intrări

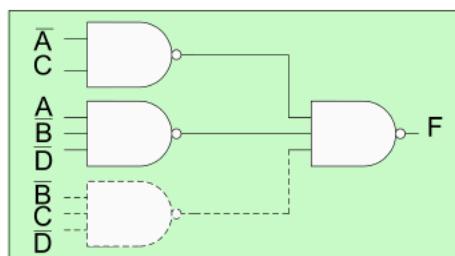
Total: 6 porti (10 intrări)

Implementarea funcției prezintă hazard combinațional ce se poate elibera prin considerarea unei suprafete suplimentare: $\bar{B} \cdot C \cdot \bar{D}$, necesitând suplimentar pentru implementare 1 poartă NAND x 3, și încă o intrare.

Cost total al implementării, fără hazard combinațional: 7 porti (14 intrări).

		CD		C		
		00	01	11	10	
AB	00	-	-	1	1	B
	01	-	-	1		
A	11	-	-	1		
	10	1			1	D

Implementarea cu porti NAND este:



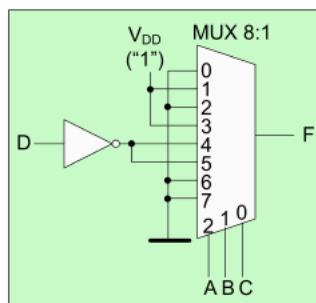
Din diagrama V-K următoare se deduc funcțiile reziduu D :

		CD		C		
		00	01	11	10	
AB	00	-	-	1	1	B
	01	-	-	1		
A	11	-	-	1		
	10	1			1	D

		AB		C		
		00	01	11	10	
CD	0	-	-	1		B
	1	-	-	1		
AB	0	-	-	1		
	1	D	D			

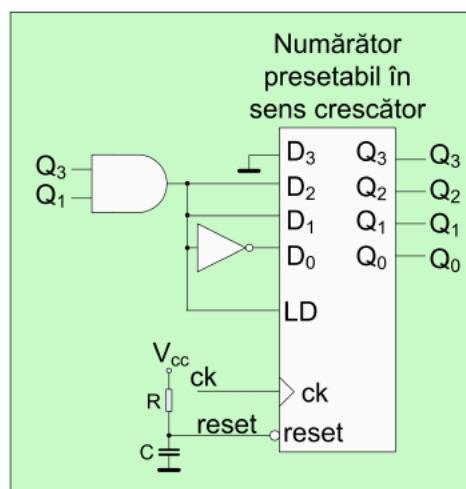


Implementarea este:



3. Analiza circuitelor logice secvențiale [2pct]

Se consideră circuitul:



Numărătorul presetabil comută sincron cu frontul pozitiv al semnalului de ceas. Dacă $LD = 1$, atunci numărătorul se încarcă cu valoarea afișată pe intrarea D . Dacă $LD = 0$, atunci numărătorul se incrementează, starea viitoare fiind $Q^+ = Q + 1$. Funcționarea numărătorului presetabil este descrisă în Verilog astfel:

```
input[3:0]      D;                      // intrarea de date presetate (4 biti)
output[3:0]     Q;                      // iesirea numaratorului (4 biti)
always @(posedge ck or negedge reset)
if (!reset) Q <= 4'b0000; else          // initializare (reset activ "0")
if (LD)         Q <= D; else             // se incarca noua stare
    Q <= Q + 1;                         // se incrementeaza starea
```

- [0.5] Explicați în limbaj natural funcționarea circuitului.
- [1.0] Reprezentați grafic comportamentul circuitului.
- [0.5] Ce rol are condensatorul prezent în schemă?

Soluție

Circuitul prezintă un numărător de 4 biți, în sens crescător, presetabil.

Din circuit, se deduce ecuația intrării de încărcare a numărătorului:

$$LD = Q_3 \cdot Q_1$$

Deci, numărătorul se încarcă dacă starea prezentă satisfac condiția $LD = Q_3 \cdot Q_1 = 1$ adică $Q_3 = 1$ și $Q_1 = 1$, adică starea respectă pattern-ul $Q_3Q_2Q_1Q_0 = 1X1X$. Pattern-ul este respectat de stările $10|_{10} = 1010|_2$, $11|_{10} = 1011|_2$, $14|_{10} = 1110|_2$, $15|_{10} = 1011|_2$.

Din circuit se deduc ecuațiile intrărilor de date:

$$D_3 = 0$$

$$D_2 = Q_3 \cdot Q_1$$

$$D_1 = Q_3 \cdot Q_1$$

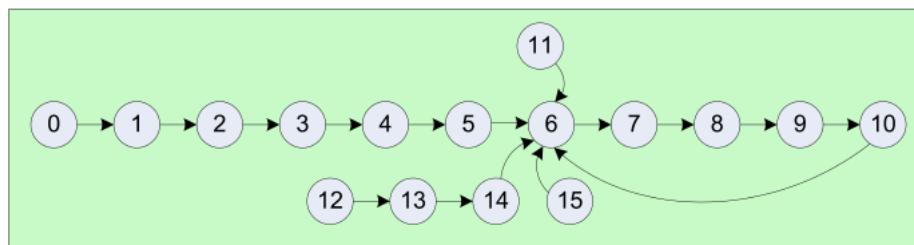
$$D_0 = \overline{Q_3} \cdot \overline{Q_1}$$

Stare prezentă $Q[3 : 0]$	Încărcare LD	Intrări $D[3 : 0]$	Acțiune	Stare viitoare $Q^+[3 : 0]$
0 0000	0	1 0001	Incrementare	1 0001
1 0001	0	1 0001	Incrementare	2 0010
2 0010	0	1 0001	Incrementare	3 0011
3 0011	0	1 0001	Incrementare	4 0100
4 0100	0	1 0001	Incrementare	5 0101
5 0101	0	1 0001	Incrementare	6 0110
6 0110	0	1 0001	Incrementare	7 0111
7 0111	0	1 0001	Incrementare	8 1000
8 1000	0	1 0001	Incrementare	9 1001
9 1001	0	1 0001	Incrementare	10 1010
10 1010	1	6 0110	Încărcare	6 0110
Se observă mai sus ciclul de stări 6, 7, 8, 9, 10, 6, ...				
Se analizează comportamentul din stările neprezente în ciclu				
11 1011	1	6 0110	Încărcare	6 0110
12 1100	0	1 0001	Incrementare	13 1101
13 1101	0	1 0001	Incrementare	14 1110
14 1110	1	6 0110	Încărcare	6 0110
15 1111	1	6 0110	Încărcare	6 0110

Numărătorul parurge în sens crescător stările: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

La apariția stării $Q[3 : 0] = 1010|_2 = 10|_{10}$ se activează $LD = 1$ și intrările devin $D[3 : 0] = 0110|_2 = 6|_{10}$. Din stările 10, 11, 14, 15 se produce o încărcare și starea devine 6. Din stările 12, 13 circuitul numără în sens crescător.

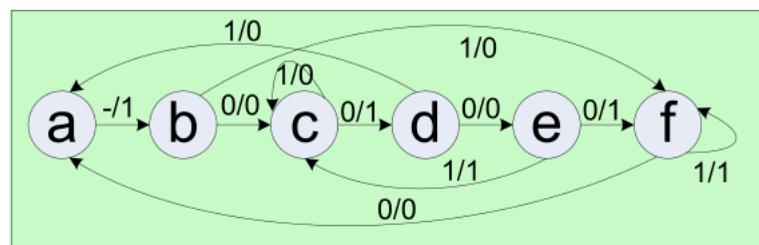
Rezultă graful de tranziții:



Condensatorul din schemă, împreună cu rezistența, formează un circuit de generare a unui puls de reset asincron, activ 0, la punerea sub tensiune a circuitului. Din punct de vedere digital, semnalul *reset* va avea inițial valoarea logică 0 (fiind activ) iar după un timp determinat de încărcarea condensatorului va ajunge la valoarea logică 1, deci se va dezactiva. Valorile RC determină constanta de timp a circuitului de integrare și determină lățimea pulsului de reset asincron.

4. Sinteză circuitelor logice secvențiale [2pct]

Se consideră graful de tranziții:



- [2.0] Implementați automatul cu bistabile T și porti NAND.

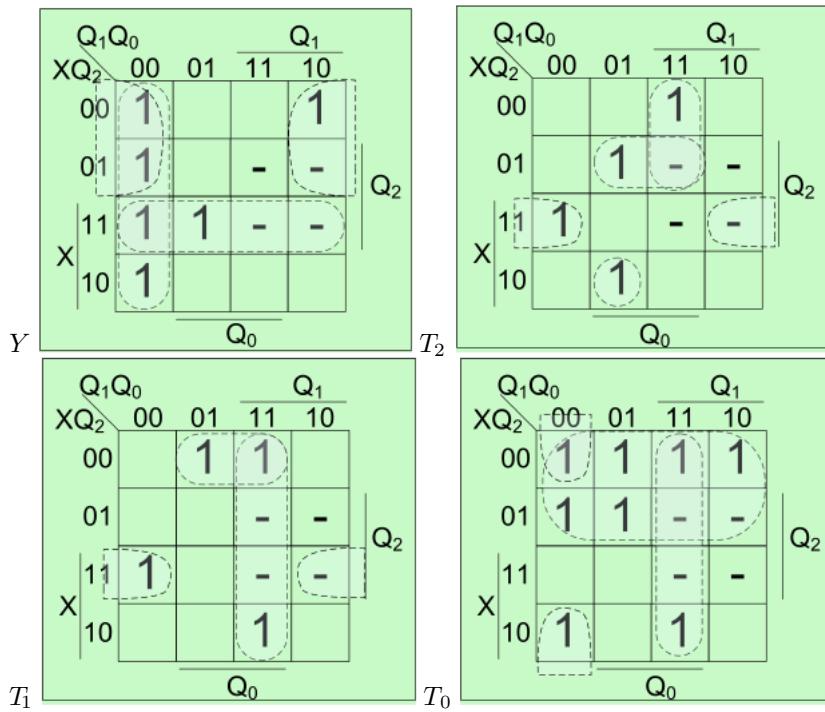
Soluție

Automat Mealy, o intrare X , o ieșire Y , 6 stări, codificate cu minim 3 biți ($Q_2Q_1Q_0$).

Codurile stărilor $a = 000$, $b = 001$, $c = 010$, $d = 011$, $e = 100$, $f = 101$. Tabelul de tranziții este:

Intrare X	Stare prezentă			Stare viitoare			Ieșire Y	Stimuli T		
	Q_2^+	Q_1^+	Q_0^+	D_2	D_1	D_0		T_2	T_1	T_0
-	a	0	0	0	b	0	0	1	0	0
0	b	0	0	1	c	0	1	0	0	1
1	b	0	0	1	f	1	0	1	1	0
0	c	0	1	0	d	0	1	1	0	0
1	c	0	1	0	c	0	1	0	0	0
0	d	0	1	1	e	1	0	0	1	1
1	d	0	1	1	a	0	0	0	0	1
0	e	1	0	0	f	1	0	1	0	0
1	e	1	0	0	c	0	1	0	1	1
0	f	1	0	1	a	0	0	0	0	1
1	f	1	0	1	f	1	0	1	0	0
-	-	1	1	0	-	-	-	-	-	-
-	-	1	1	1	-	-	-	-	-	-

Diagramele V-K:



Expresii minimizează:

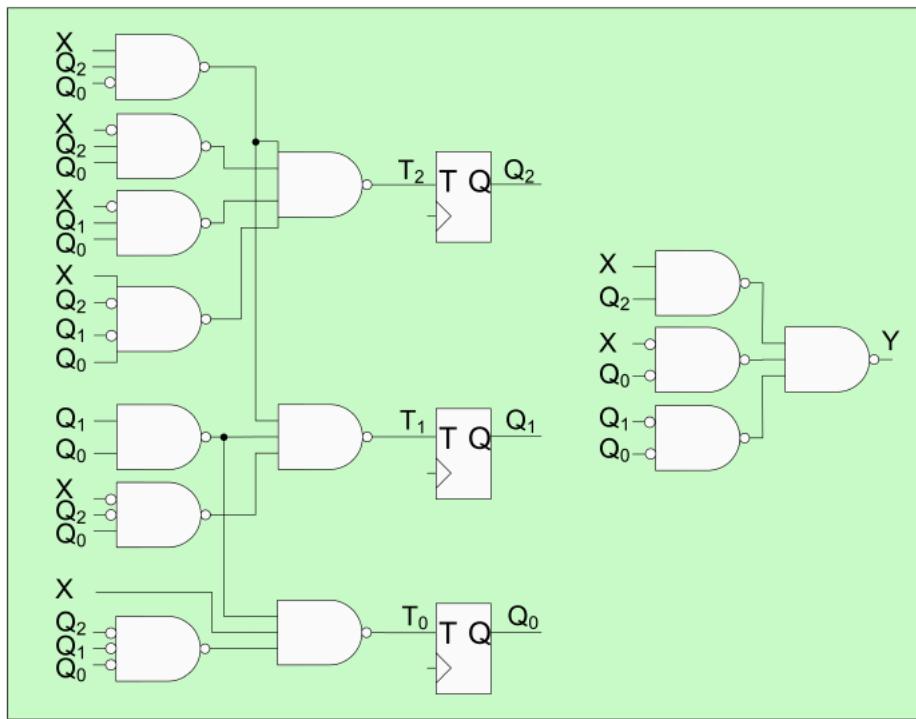
$$Y(X, Q_2, Q_1, Q_0) = X \cdot Q_2 + \overline{X} \cdot \overline{Q_0} + \overline{Q_1} \cdot \overline{Q_0}$$

$$T_2(X, Q_2, Q_1, Q_0) = X \cdot Q_2 \cdot \overline{Q_0} + \overline{X} \cdot Q_2 \cdot Q_0 + \overline{X} \cdot Q_1 \cdot Q_0 + X \cdot \overline{Q_2} \cdot \overline{Q_1} \cdot Q_0$$

$$T_1(X, Q_2, Q_1, Q_0) = X \cdot Q_2 \cdot \overline{Q_0} + Q_1 \cdot Q_0 + \overline{X} \cdot \overline{Q_2} \cdot Q_0$$

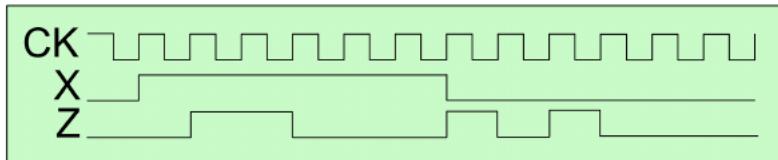
$$T_0(X, Q_2, Q_1, Q_0) = Q_1 \cdot Q_0 + \overline{X} + \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0}$$

Implementare:



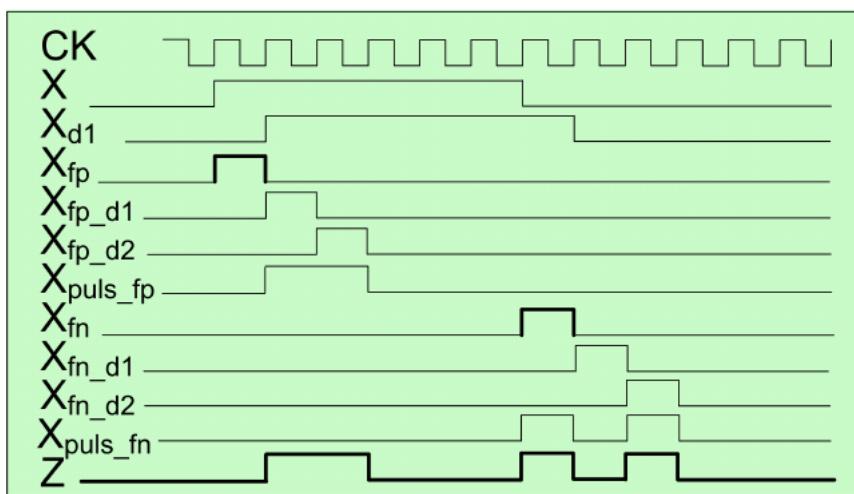
5. Proiectare sistem digital [1pct]

- [1.0] Proiectați un sistem secvențial sincron cu o intrare X și o ieșire Z. La fiecare front pozitiv al semnalului X apare un puls pe ieșirea Z. La fiecare front negativ al semnalului X apare o secvență de două pulsuri pe ieșirea Z. Formele de undă și temporizarea sunt prezentate în figură.



Soluție

Rezolvarea problemei constă în detectarea celor două fronturi (pozitiv și negativ) ale semnalului X și prelucrarea pulsurilor pentru a genera secvența cu forma și temporizarea dorite.



X_{d1} este X întârziat cu o perioadă, printr-un bistabil.

X_{fp} este un puls care marchează frontul pozitiv al semnalului X : $X_{fp} = X \cdot \overline{X_{d1}}$.

X_{fp_d1} și X_{fp_d2} sunt versiuni întârziate cu 1 și 2 perioade față de X_{fp} , utilizate pentru a formata secvența

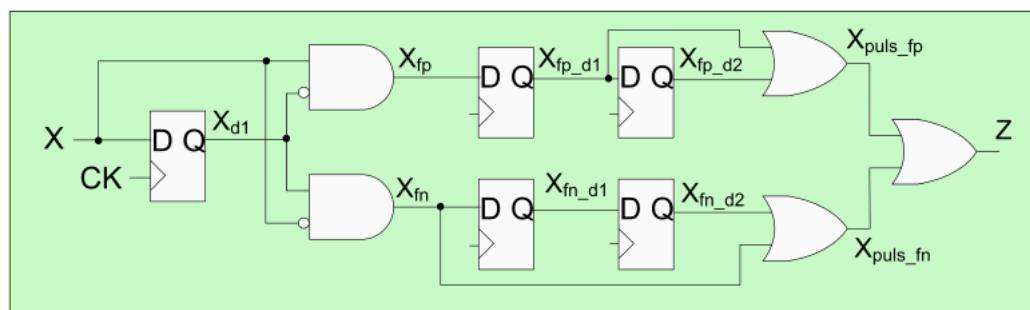
necesară a fi generată la frontul pozitiv: $X_{puls_fp} = X_{fp_d1} + X_{fp_d2}$.

X_{fn} este un puls care marchează frontul negativ al semnalului X : $X_{fn} = \overline{X} \cdot X_{d1}$.

X_{fn_d1} și X_{fn_d2} sunt versiuni întârziate cu 1 și 2 perioade față de X_{fn} , utilizate pentru a forma secvența necesară a fi generată la frontul negativ: $X_{puls_fn} = X_{fn} + X_{fn_d2}$.

În final $Z = X_{puls_fp} + X_{puls_fn}$.

Circuitul rezultat este:



6. Bonus [2pct]

Un aparat de radio are un buton prin apăsare pentru selecția digitală a frecvenței.

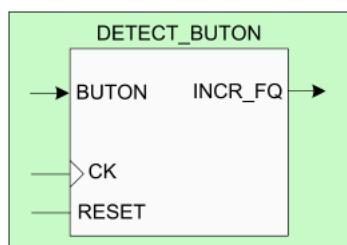


La o apăsare, frecvența se incrementează cu o unitate. Dacă butonul se ține apăsat mai mult de 2 secunde, frecvența se incrementează automat cu 10 unități pe secundă.

Să se proiecteze un sistem secvențial sincron cu frecvența de 100 MHz care să genereze **pulsurile de incrementare** pe baza semnalului provenit de la **buton**.

Soluție

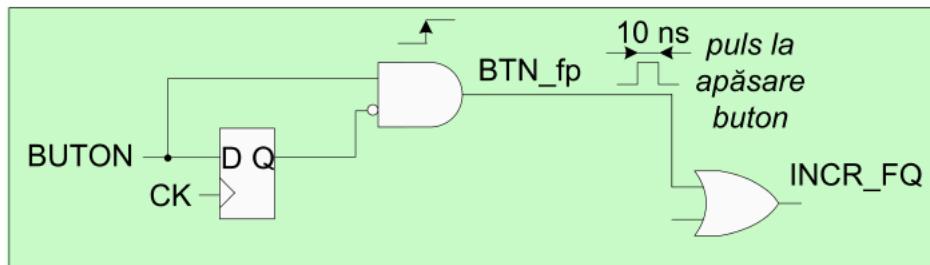
Modulul proiectat este un sistem sincron cu o intrare denumită *BUTON* și o ieșire denumită *INCR_FQ*. Perioada semnalului de ceas este $T_{CK} = 1/f_{CK} = 1/100 \text{ MHz} = 10 \text{ ns}$.



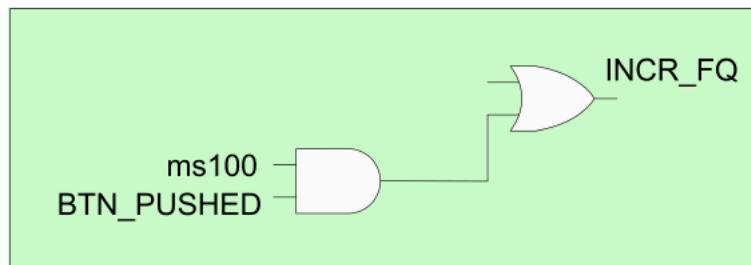
Ieșirea *INCR_FQ* se activează din două motive (SAU logic):

- la detectarea apăsării butonului (front pozitiv la intrările *BUTON*) și
- periodic, la fiecare 100 ms, dacă butonul este ținut apăsat (*BUTON* = 1).

Prima condiție se monitorizează cu un circuit detector de front pozitiv:



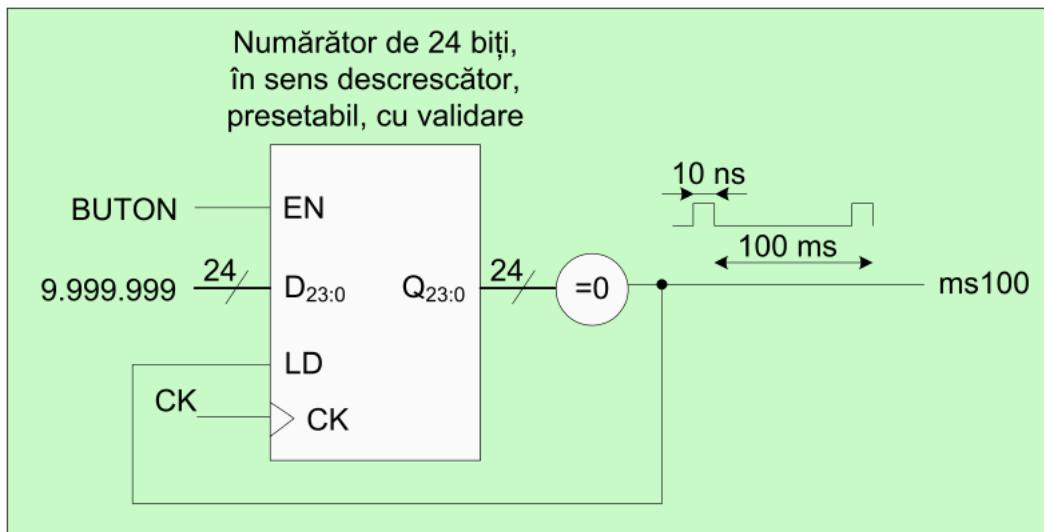
A doua condiție se obține prin funcția AND între un semnal activ dacă butonul este ținut apăsat mai mult de 2 secunde (denumit *BTN_PUSHED*) și un semnal care prezintă un puls cu lățime de o perioadă de ceas la fiecare 100 ms (denumit *ms100*).



Semnalul *BTN_PUSHED* se obține la ieșirea unui bistabil RS care se setează dacă *BUTON* = 1 mai mult de 2 secunde și se resetează la revenirea butonului (*BUTON* comută din 1 în 0, *BTN_fn* = 1). Cât timp *BTN_PUSHED* = 1 la ieșire se propagă un set de pulsuri de la un numărător cu perioadă de 100 ms.

Numărătorul cu perioadă de 100 ms contorizează un număr de perioade egal cu raportul între intervalul de timp și perioada de ceas: $100 \text{ ms} / 10 \text{ ns} = 10^7 = 10.000.000$. Numărul de biți necesari este 24 deoarece $2^{24} = 16M > 10.000.000$. Numărătorul trebuie să aibă următoarele caracteristici:

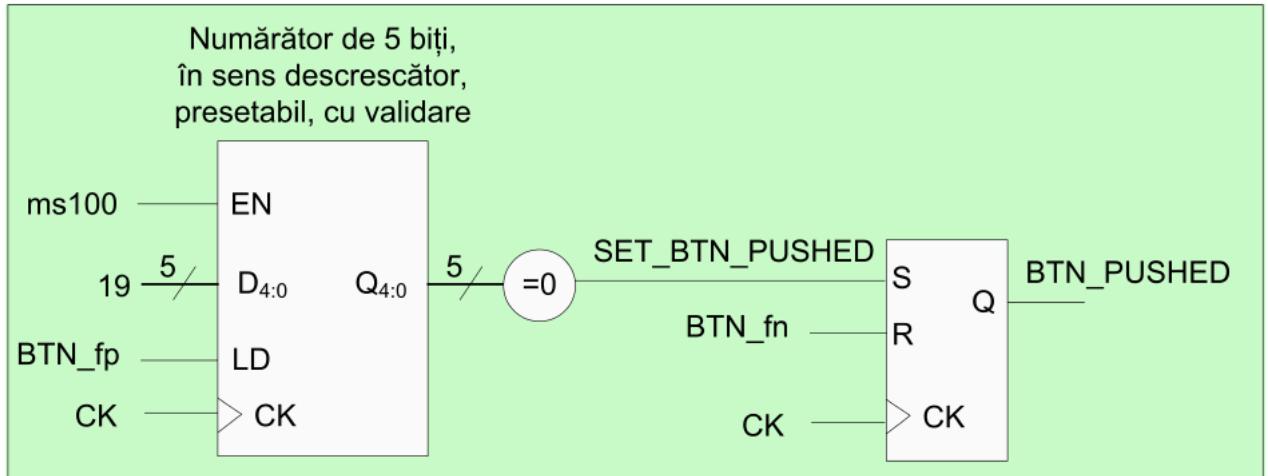
- facilitate de presetare la valoarea 9.999.999 (numărarea se face între 9.999.999 și 0);
- semnalare printr-un puls la ajungerii la zero (acest puls va preseta numărătorul, astfel încât numărătoarea să fie reluată);
- facilitate de validare doar dacă butonul este apăsat (*BUTON* = 1).



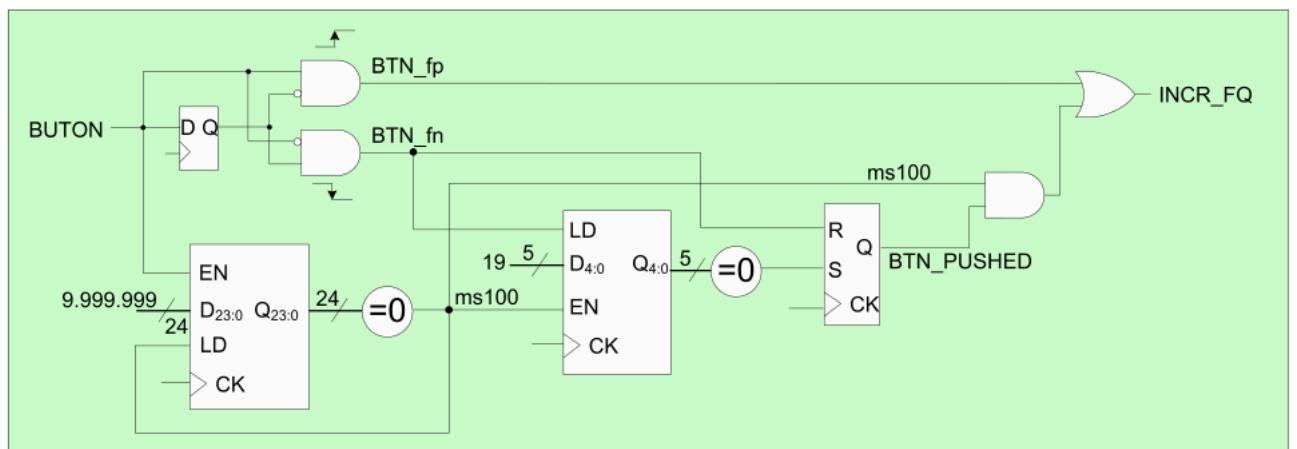
Durata de 2 secunde se măsoară (cu un alt numărător) ca fiind 20 de pulsuri ale semnalului *ms100*. Număratorul are 5 biți ($2^5 > 20$) și următoarele caracteristici:

- numără în sens descrescător la fiecare puls cu perioada de 100 ms ($ms100 = 1$);
- se initializează cu 19 la apăsarea pe buton (numărarea va fi între 19 și 0);
- dacă ajunge la zero, activează un indicator (pe baza căruia se setează *BTN_PUSHED* = 1, semnificând apăsarea pe buton menținută mai mult de 2 secunde).





Circuitul care implementează modulul cerut este:



20.3 Subiect 2012 - 3

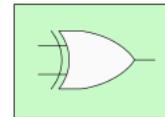
1. Întrebări [2pct]

- [0.2] Definiți noțiunea de "poartă logică XOR".

Soluție

Poarta logică XOR (sau exclusiv) are tabelul de funcționare și simbolul:

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



Poarta XOR poate fi folosită ca:

- circuit sumator de 2 biți;
- circuit pentru determinarea inegalității dintre doi biți (sau imparitatea numărului de intrări egale cu 1);
- inversor comandat.

- [0.2] Definiți noțiunea de "latch D".

Soluție

Latch-ul D este un circuit care memorează starea, activ pe palierul semnalului de ceas. Dacă $CK = 1$ latch-ul este transparent și preia starea intrării D. Dacă $CK = 0$ latch-ul memorează starea.

- [0.2] Definiți noțiunea de "fan-out".

Soluție

Fan-out-ul unei porți logice reprezintă numărul de porți logice de același tip care pot fi comandate de la ieșirea unei porți logice. Se definește ca raportul între curentul de ieșire și curentul de intrare al porții logice.

Fan-out în starea H (High = 1 logic): $FO_H = I_{OH} - I_{IH}$

Fan-out în starea L (Low = 0 logic): $FO_L = I_{OL} - I_{IL}$

Fan-out: $FO = \min(FO_H, FO_L)$

- [0.2] Definiți proprietatea fundamentală a codului Gray.

Soluție

Codul Gray are proprietatea că două coduri alăturate diferă printr-un singur bit.

- [0.2] Definiți noțiunea de "structură pipeline".

Soluție

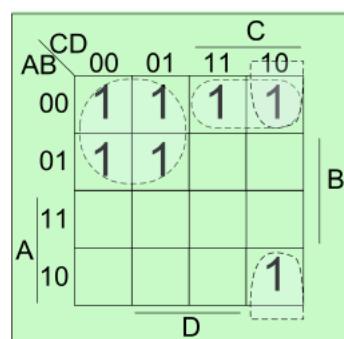
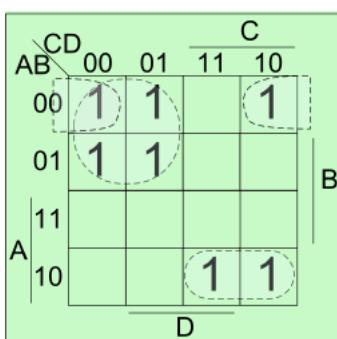
Structură pipeline este o modalitate de organizare a unui sistem digital similar cu o "linie de asamblare". Structura pipeline este formată din succesiuni de circuite combinaționale și registre. Scopul acestui tip de organizare constă în creșterea frecvenței de operare a sistemului prin micșorarea numărului de nivale logice dintre două registre.

- [0.5] Justificați valoarea de adevar a expresiei:

$$\overline{A} \cdot \overline{C} + A \cdot \overline{B} \cdot C + \overline{A} \cdot \overline{B} \cdot \overline{D} = \overline{A} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C + \overline{B} \cdot C \cdot \overline{D}$$

Soluție

Se construiesc diagramele V-K asociate expresiilor din stânga și din dreapta.



Se observă că expresia este falsă:

$$\sum(0, 1, 2, 4, 5, 10, 11) \neq \sum(0, 1, 2, 3, 4, 5, 10)$$

- [0.5] Reprezentați operația prezentată cu numere în baza zece, cu operanzi reprezentați în complement față de 2, pe 16 biți.

$$1004 - (-1008) = 2012|_{10}$$

Soluție

$$1004|_{10} = 512 + 256 + 128 + 64 + 32 + 8 + 4 = 11\text{.}1110\text{.}1100|_2 = 0000\text{.}0011\text{.}1110\text{.}1100|_2$$

$$1008|_{10} = 512 + 256 + 128 + 64 + 32 + 16 = 11\text{.}1111\text{.}0000|_2 = 0000\text{.}0011\text{.}1111\text{.}0000|_2$$

$$-1008|_{10} = 1111\text{.}1100\text{.}0001\text{.}0000|_2$$

$$2012|_{10} = 0000\text{.}0111\text{.}1101\text{.}1100|_2$$

$$\begin{array}{r} 0000\text{.}0011\text{.}1110\text{.}1100+ \\ 0000\text{.}0011\text{.}1111\text{.}0000 \\ \hline 0000\text{.}0111\text{.}1101\text{.}1100 \end{array}$$

$$\begin{array}{r} 0000\text{.}0011\text{.}1110\text{.}1100- \\ 1111\text{.}1100\text{.}0001\text{.}0000 \\ \hline 0000\text{.}0111\text{.}1101\text{.}1100 \end{array}$$

2. Circuite logice combinaționale [2pct]

Se consideră funcția:

$$F = \sum(3, 6, 7, 8, 12, 14, 15) + d(0, 1, 4, 5)$$

- [0.5] Minimizați funcția logică și implementați-o cu porți NAND.
- [0.5] Evaluați "costul" implementării.
- [0.5] Propuneți o soluție de eliminare a hazardului combinațional și evaluați costul suplimentar al acesteia.
- [0.5] Implementați funcția cu MUX 8:1 și un număr minim de porți logice suplimentare.

Soluție

Diagrama V-K:

		CD		C		B
		00		01	11	
AB	00	-	-	1		B
	01	-	-	1	1	
A	11	1		1	1	B
	10	1				
		D				

$$\text{Expresia minimizată: } F(A, B, C, D) = B \cdot C + \overline{C} \cdot \overline{D} + \overline{A} \cdot D$$

Necesar de porți logice:

3 NAND x 2 = 6 intrări

1 NAND x 3 = 3 intrări

3 NOT x 1 = 3 intrări

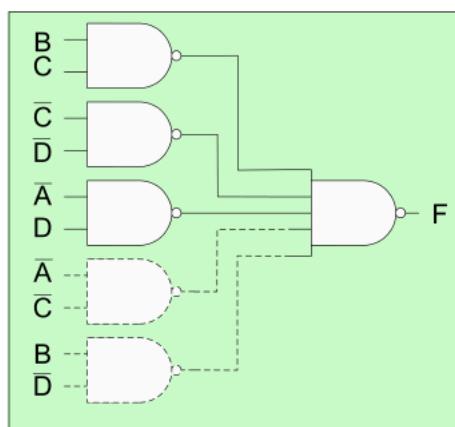
Total: 7 porți (12 intrări)

Implementarea funcției prezintă hazard combinațional ce se poate elibera prin considerarea a două suprafete suplimentare: $\overline{A} \cdot \overline{C}$ și $B \cdot \overline{D}$, necesitând suplimentar pentru implementare 2 porți NAND x 2 și încă două intrări. Cost total al implementării, fără hazard combinațional: 9 porți (18 intrări).



	CD	AB	00	01	11	10	C
		00	-	-	1		
		01	-	-	1	1	
		11	1		1	1	
		10	1				

Implementarea cu porti NAND este:

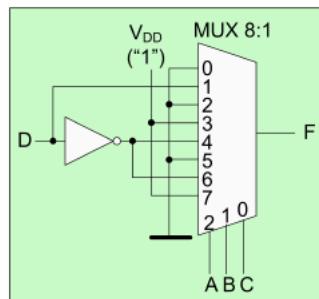


Din diagrama V-K următoare se deduc funcțiile reziduu D :

	CD	AB	00	01	11	10	C
		00	-	-	1		
		01	-	-	1	1	
		11	1		1	1	
		10	1				

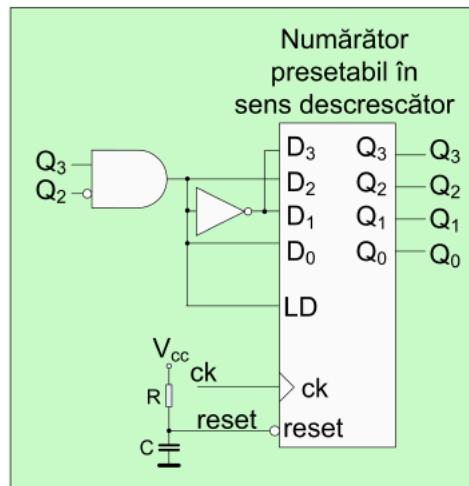
	C	AB	0	1	C
		00	D		
		01		1	
		11	D-bar	1	
		10	D		

Implementarea este:



3. Analiza circuitelor logice secvențiale [2pct]

Se consideră circuitul:



Numărătorul presetabil comută sincron cu frontul pozitiv al semnalului de ceas. Dacă $LD = 1$, atunci numărătorul se încarcă cu valoarea afaltă pe intrarea D . Dacă $LD = 0$, atunci numărătorul se decrementează, starea viitoare fiind $Q^+ = Q - 1$. Funcționarea numărătorul presetabil este descrisă în Verilog astfel:

```
input [3:0]      D;                      // intrarea de date presetate (4 biti)
output [3:0]     Q;                     // iesirea numaratorului (4 biti)
always @ (posedge ck or negedge reset)
if (!reset) Q <= 4'b0000; else           // initializare (reset activ "0")
if (LD)          Q <= D; else             // se incarca noua stare
                                Q <= Q - 1;                // se decrementeaza starea
```

- [0.5] Explicați în limbaj natural funcționarea circuitului.
- [1.0] Reprezentați grafic comportamentului circuitului.
- [0.5] Ce rol are condensatorul prezent în schemă?

Soluție

Circuitul prezintă un numărător de 4 biți, în sens descrescător, presetabil. Intrarea de încărcare are ecuația: $LD = Q_3 \cdot \overline{Q_2}$, activă când $Q[3 : 0] = 10XX$ (adică 8, 9, 10, 11₁₀).

Din circuit, se deduc ecuațiile intrărilor de presetare ale numărătorului:

$$D_3 = Q_3 \cdot \overline{Q_2}$$

$$D_2 = Q_3 \cdot \overline{Q_2}$$

$$D_1 = Q_3 \cdot \overline{Q_2}$$

$$D_0 = Q_3 \cdot \overline{Q_2}$$

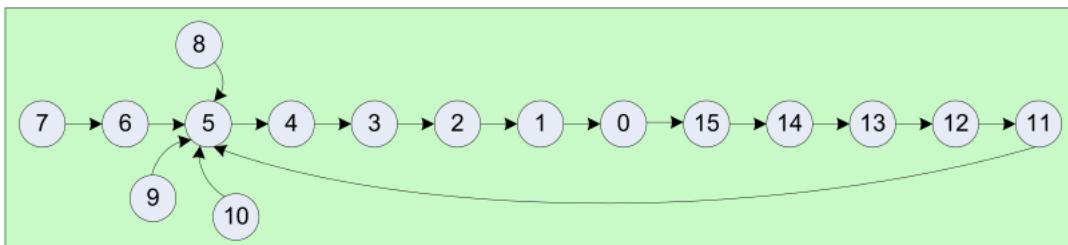
Numărătorul va parurge în sens descrescător stările: 7, 6, 5, 4, 3, 2, 1, 0, 15, 14, 13, 12, 11.

La apariția stării $Q[3 : 0] = 1011_2 = 11_{10}$ (prima din setul de stări care respectă patternu-ul care determină presetarea numărătorului, se activează $LD = 1$ și intrările devin $D[3 : 0] = 0101_2 = 5_{10}$

Datele pot fi centralizate într-un tabel:

Stare prezentă $Q[3 : 0]$	Încărcare LD	Intrări $D[3 : 0]$	Acțiune	Stare viitoare $Q^+[3 : 0]$
0 0000	0	10 1010	Decrementare	15 1111
15 1111	0	10 1010	Decrementare	14 1110
14 1110	0	10 1010	Decrementare	13 1101
13 1101	0	10 1010	Decrementare	12 1100
12 1100	0	10 1010	Decrementare	11 1011
11 1011	1	5 0101	Încărcare	5 0101
5 0101	0	10 1010	Decrementare	4 0100
4 0100	0	10 1010	Decrementare	3 0011
3 0011	0	10 1010	Decrementare	2 0010
2 0010	0	10 1010	Decrementare	1 0001
1 0001	0	10 1010	Decrementare	0 0000
Se observă mai sus ciclul de stări 0, 15, 14, 13, 12, 11, 5, 4, 3, 2, 1, 0, ...				
Se analizează comportamentul din stările neprezente în ciclu				
6 0110	0	10 1010	Decrementare	5 0101
7 0111	0	10 1010	Decrementare	6 0110
8 1000	1	5 0101	Încărcare	5 0101
9 1001	1	5 0101	Încărcare	5 0101
10 1010	1	5 0101	Încărcare	5 0101

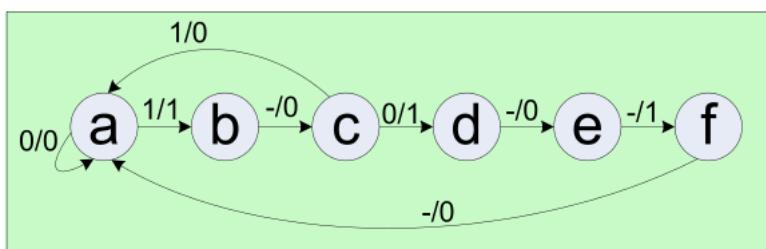
Rezultă graful de tranziții:



Condensatorul din schemă, împreună cu rezistența, formează un circuit de generare a unui puls de reset asincron, activ 0, la punerea sub tensiune a circuitului. Din punct de vedere digital, semnalul *reset* va avea inițial valoarea logică 0 (fiind activ) iar după un timp determinat de încărcarea condensatorului va ajunge la valoarea logică 1, deci se va dezactiva. Valorile RC determină constanta de timp a circuitului de integrare și determină lățimea pulsului de reset asincron.

4. Sinteză circuitelor logice secvențiale [2pct]

Se consideră graful de tranziții:



- [2.0] Implementați automatul cu bistabile T și porti NAND.

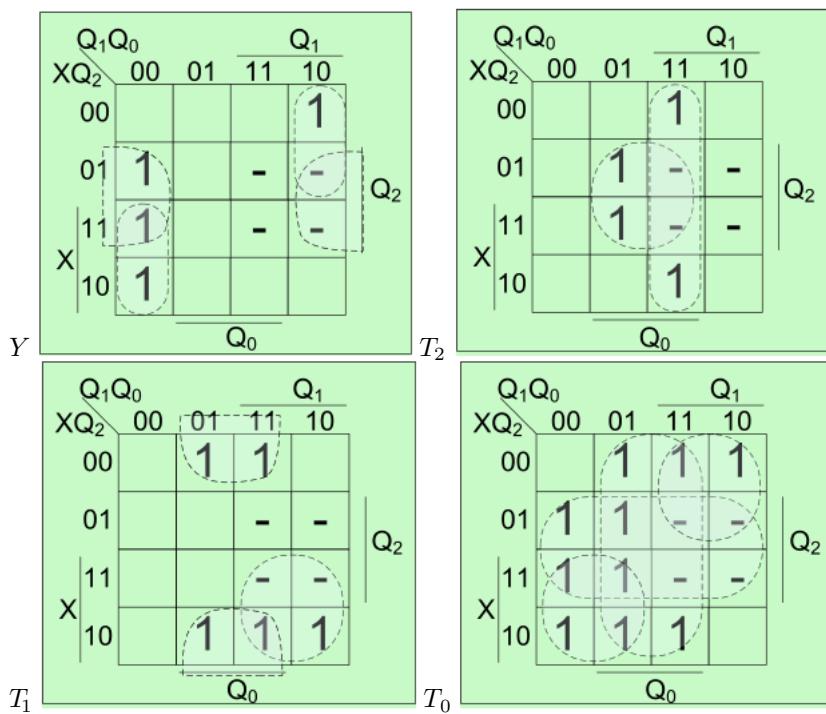
Soluție

Automat Mealy, o intrare X , o ieșire Y , 6 stări, codificate cu minim 3 biți ($Q_2Q_1Q_0$).

Codurile stărilor $a = 000$, $b = 001$, $c = 010$, $d = 011$, $e = 100$, $f = 101$. Tabelul de tranziții este:

Intrare X	Stare prezentă			Stare viitoare			Ieșire Y	Stimuli T		
	Q_2^+	Q_1^+	Q_0^+	D_2	D_1	D_0		T_2	T_1	T_0
0	a	0	0	0	a	0	0	0	0	0
1	a	0	0	0	b	0	0	1	0	0
-	b	0	0	1	c	0	1	0	0	1
0	c	0	1	0	d	0	1	1	0	0
1	c	0	1	0	a	0	0	0	0	1
-	d	0	1	1	e	1	0	0	1	1
-	e	1	0	0	f	1	0	1	0	0
-	f	1	0	1	a	0	0	0	1	0
-	-	1	1	0	-	-	-	-	-	-
-	-	1	1	1	-	-	-	-	-	-

Diagramele V-K:



Expresii minimizează:

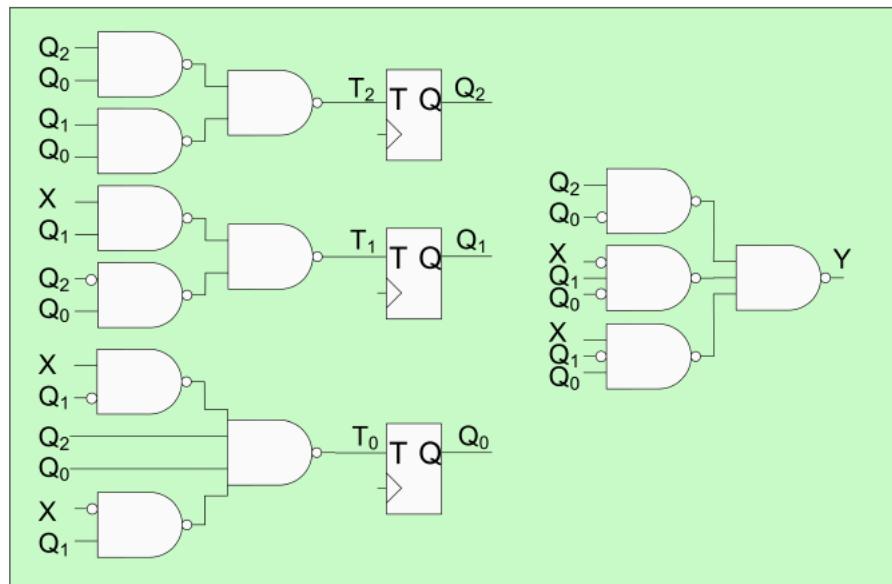
$$Y(X, Q_2, Q_1, Q_0) = Q_2 \cdot \overline{Q_0} + \overline{X} \cdot Q_1 \cdot \overline{Q_0} + X \cdot \overline{Q_1} \cdot Q_0$$

$$T_2(X, Q_2, Q_1, Q_0) = Q_2 \cdot Q_0 + Q_1 \cdot Q_0$$

$$T_1(X, Q_2, Q_1, Q_0) = X \cdot Q_1 + \overline{Q_2} \cdot Q_0$$

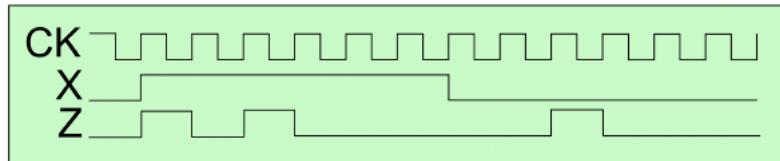
$$T_0(X, Q_2, Q_1, Q_0) = Q_2 + Q_0 + X \cdot \overline{Q_1} + \overline{X} \cdot Q_1$$

Implementare:



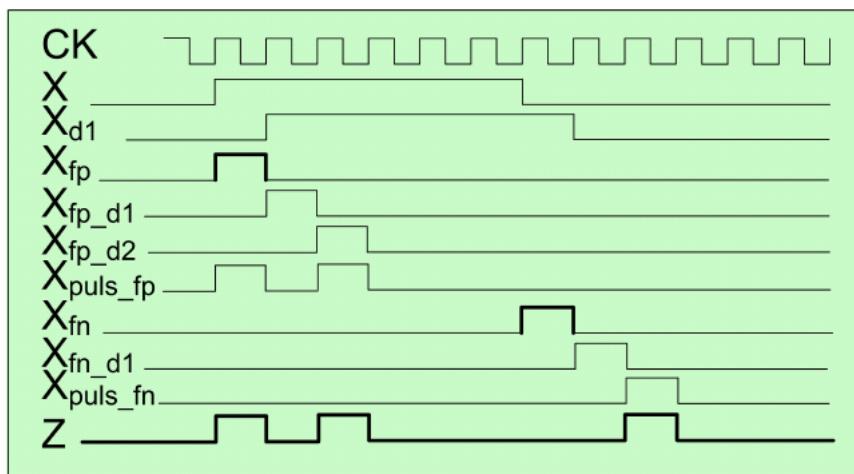
5. Proiectare sistem digital [1pct]

- [1.0] Proiectați un sistem secvențial sincron cu o intrare X și o ieșire Z. La fiecare front pozitiv al semnalului X apar două pulsuri pe ieșirea Z. La fiecare front negativ al semnalului X apare un puls pe ieșirea Z. Formele de undă și temporizarea sunt prezentate în figură.



Soluție

Rezolvarea problemei constă în detectarea celor două fronturi (pozitiv și negativ) ale semnalului X și prelucrarea pulsurilor pentru a genera secvența cu forma și temporizarea dorite.



X_{d1} este X întârziat cu o perioadă, printr-un bistabil.

X_{fp} este un puls care marchează frontul pozitiv al semnalului X : $X_{fp} = X \cdot \overline{X_{d1}}$.

X_{fp_d1} și X_{fp_d2} sunt versiuni întârziate cu 1 și 2 perioade față de X_{fp} , utilizate pentru a forma secvența necesară a fi generată la frontul pozitiv: $X_{puls_fp} = X_{fp} + X_{fp_d2}$.

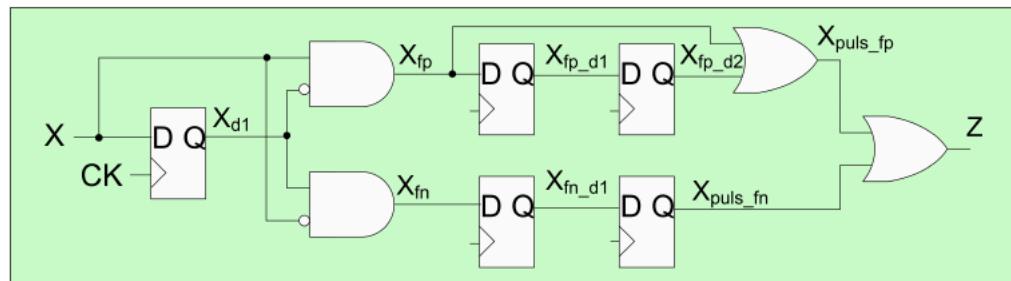
X_{fn} este un puls care marchează frontul negativ al semnalului X : $X_{fn} = \overline{X} \cdot X_{d1}$.

X_{fn_d1} și X_{puls_fn} sunt versiuni întârziate cu 1 și 2 perioade față de X_{fn} .

În final $Z = X_{puls_fp} + X_{puls_fn}$.

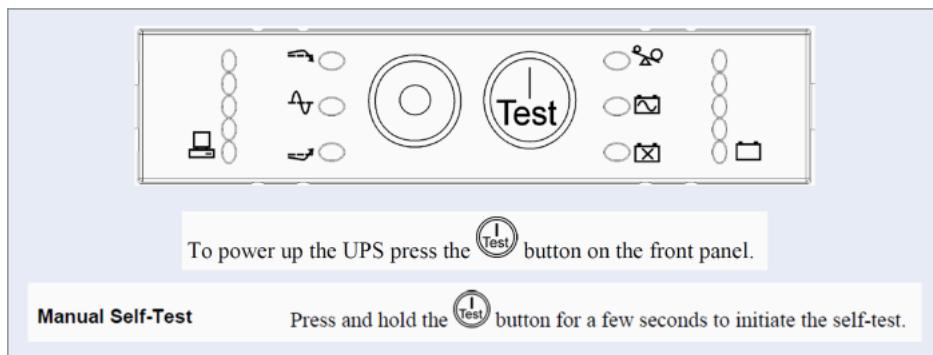
Circuitul rezultat este:





6. Bonus [2pct]

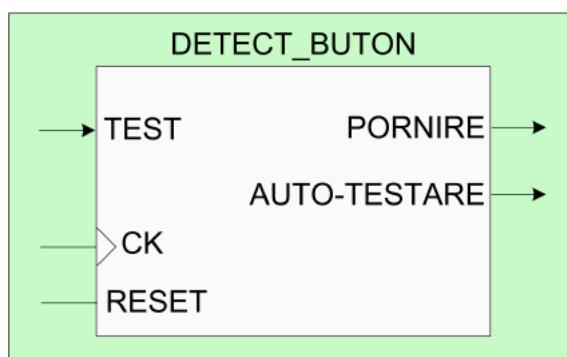
Manualul de utilizare a sursei de tensiune neîntreruptibile APC Smart-UPS prezintă panoul frontal și modalitatea de pornire și testare prin acționarea aceluiași buton.



Proiectați un sistem digital secvențial sincron, având frecvența de ceas de 100 MHz, care preia intrarea de la butonul **Test** și generează **două impulsuri** (cu lățime o perioadă de ceas) pentru începerea celor două funcții: *pornire* și *auto-testare*.

Soluție

Modulul proiectat este un sistem sincron cu o intrare denumită *BUTON* și două ieșiri denumite *PORNIRE* și *AUTO – TESTARE*. Perioada semnalului de ceas este $T_{CK} = 1/f_{CK} = 1/100 \text{ MHz} = 10 \text{ ns}$.



Cele două ieșiri prezintă câte un puls cu lățimea egală cu o perioadă a semnalului de ceas în următoarele condiții:

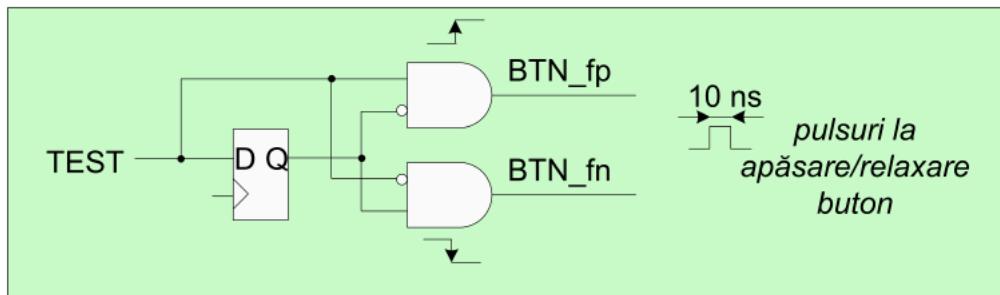
- *PORNIRE*, în momentul relaxării butonului (*TEST* comută din 1 în 0) dacă intervalul de timp de la apăsarea butonului a fost mai mic decât "câteva secunde";

- *AUTO – TESTARE*, în momentul relaxării butonului (*TEST* comută din 1 în 0) dacă intervalul de timp de la apăsarea butonului a fost mai mare decât "câteva secunde".

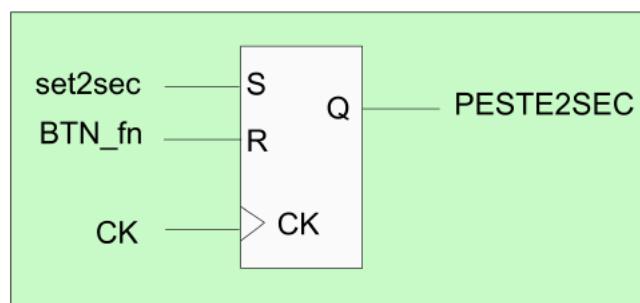
Se consideră intervalul de timp a fi egal cu 2 secunde.

Cele 2 secunde trebuie măsurate din momentul apăsării butonului. După scurgerea intervalului impus de 2 secunde, se setează un indicator *PESTE2SEC*. Dacă depistarea relaxării butonului se face când *PESTE2SEC* = 0, atunci se activează *PORNIRE*. Altfel, dacă depistarea relaxării butonului se face când *PESTE2SEC* = 1, atunci se activează *AUTO – TESTARE*. Este necesar un circuit demultiplexor 1:2, care să aibă conectat pe selecție semnalul *PESTE2SEC* și la intrarea de date pulsul care marchează frontul negativ al intrării *TEST*.

Circuitele care generează pulsuri la fronturile semnalului de intrare *TEST*:

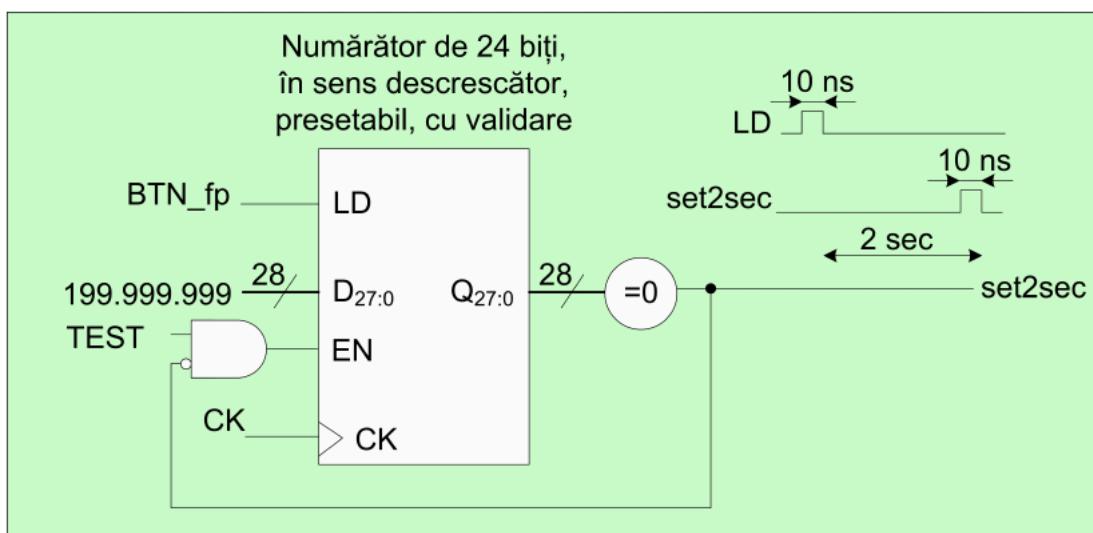


Semnalul *PESTE2SEC* se obține la ieșirea unui bistabil RS care se setează dacă *TEST* = 1 mai mult de 2 secunde și se resetează la revenirea butonului (*TEST* comută din 1 în 0, *BTN_fn* = 1).



Numărătorul pentru măsurarea intervalului de 2 secunde contorizează un număr de perioade egal cu raportul între intervalul de timp și perioada de ceas: $2 \text{ sec}/10 \text{ ns} = 2 \cdot 10^8 = 200.000.000$. Numărul de biți necesari este 28 deoarece $2^{28} = 256M > 200.000.000$. Numărătorul trebuie să aibă următoarele caracteristici:

- facilitate de presetare la valoarea 199.999.999 (numărarea se face între 199.999.999 și 0);
- semnalare printr-un puls a ajungerii la zero (acest puls va opri număratorul);
- facilitate de validare doar dacă butonul este apăsat (*TEST* = 1).



Circuitul care implementează modulul cerut este:



