

# CODECURI DE LINIE BINARE

## 1. SINTEZA LOGICA A CODEC-URILOR BINARE

Circuitele de codare/decodare sunt implementate ca automate cu număr finit de stări, folosind porți logice și bistabile. Implementarea acestor coduri poate fi realizată și în variantă „software”, cu algoritmi de codare/decodare, folosind procesoare digitale de semnal. În continuare se prezintă o metodă de implementare la nivel de circuit a principalelor codecuri de linie binare pe baza sintezei digitale cu ajutorul diagramelor Veitch-Karnaugh.

### 1.1. Elemente de sinteză digitală

Sinteza unui circuit digital se poate realiza cu ajutorul diagramelor Veitch-Karnaugh, plecându-se de la tabloul de stări al circuitului pe care dorim să-l implementăm.

O diagramă Veitch-Karnaugh este reprezentată de un pătrat (sau dreptunghi) format din  $2^n$  compartimente, unde  $n$  reprezintă numărul variabilelor de intrare. În continuare vom nota cu  $p$  și  $r$  numărul de linii, respectiv numărul de coloane al diagramei. Evident, are loc o relație de forma  $2^n = 2^p \cdot 2^r$ . În fiecare compartiment al diagramei Veitch-Karnaugh se reprezintă valoarea vectorului funcției pentru termenul reprezentat de respectivul compartiment. În Fig. 1 se reprezintă câteva forme constructive ale acestor diagrame.

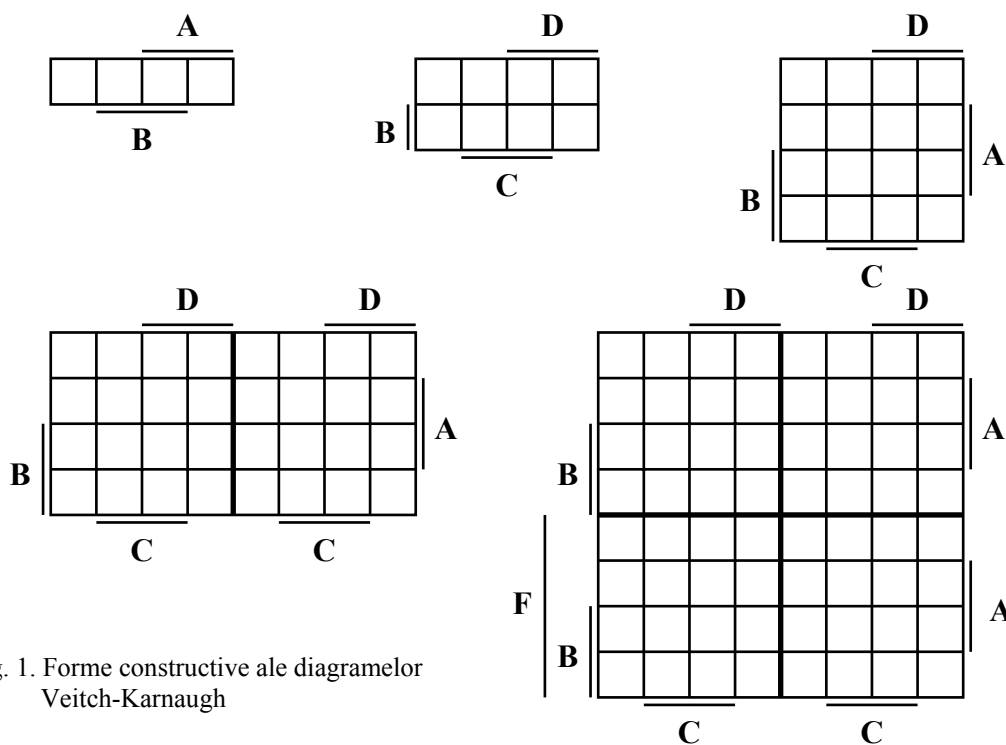


Fig. 1. Forme constructive ale diagramelor Veitch-Karnaugh

O diagramă Veitch-Karnaugh are ca principală proprietate faptul că de la un compartiment la altul, în forma expresiei termenilor canonici corespunzători, se modifică doar o singură variabilă. Un termen este *canonic* atunci când conține toate variabilele. Doi termeni care diferă prin valoarea unei singure variabile se numesc *adiacenți*.

Într-o diagramă Veitch-Karnaugh de 1, 2, 3 sau 4 variabile, sunt adiacenți termenii plasați în compartimente vecine și cei aflați la capetele unei linii sau a unei coloane.

Minimizarea funcțiilor folosind diagrame Veitch-Karnaugh se bazează pe proprietatea de adiacență a termenilor canonici din diagramă. Aceasta constă în principal din formarea de grupuri de **2**, **4**, **8**, ...,  **$2^r$**  termeni adiacenți, grupuri care se înlocuiesc cu un termen echivalent numit implicanț prim.

Expresia termenului echivalent a unui grup de  **$2^r$**  termeni canonici este formată din  **$n - r$**  variabile.

Minimizarea unei funcții cu ajutorul diagramei Veitch-Karnaugh se realizează astfel:

- 1) Se notează într-o diagramă Veitch-Karnaugh toate valorile de „1” în cazul funcțiilor sub formă disjunctivă (funcția este complet definită);
- 2) Se formează grupuri de forma  **$2^r$**  termeni adiacenți cu  **$r$**  cât mai mare;
- 3) Se stabilesc expresiile implicanților primi;
- 4) Se stabilesc implicanții primi esențiali;
- 5) Se scrie o acoperire sau mai multe pentru funcție.

### **Observații:**

*O acoperire se construiește luând implicanții primi esențiali și o parte (sau toți) din restul implicanților primi.*

*Un termen nenul al funcției, dacă este conținut într-un singur implicanț prim, îl transformă pe acesta într-un implicanț prim esențial.*

*Un termen al cărui coeficient este nenul poate fi conținut în mai mulți implicanți primi, iar la acoperirea funcției el trebuie să fie conținut cel puțin odată.*

*În cazul formei canonice disjunctive, acoperirea se referă la considerarea tuturor valorilor nenule ale funcției.*

## 1.2. Legile de codare ale principalelor codecuri de linie binare

În Tabelul 1 se prezintă legile matematice ale principalelor codecuri de linie binare, pe baza cărora rezultă implementarea acestora, reprezentată în Fig. 2.

Tabel 1

Cod de linie binar	Lege matematică codor	Lege matematică decodor
NRZ-M	$y_k = x_k \oplus y_{k-1}$	$x_k = y_k \oplus y_{k-1}$
NRZ-S	$y_k = \overline{x_k \oplus y_{k-1}}$	$x_k = \overline{y_k \oplus y_{k-1}}$
RZ	$y_k = c_k \cdot x_k$	$x_k = \overline{\overline{y_k \cdot x_{k-1/2} \cdot c_k}}$
BIF-L	$y_k = \overline{x_k \oplus c_k}$	$x_k = \overline{y_k \oplus c_k}$
BIF-M	$y_k = \overline{x_k \oplus z_{k-1} \oplus c_k}$	$x_k = \overline{y_k \oplus y_{k-1}}$
BIF-S	$y_k = x_k \oplus z_{k-1} \oplus c_k$	$x_k = y_k \oplus y_{k-1}$

În Tabelul 1,  $x_k$  reprezintă secvența de date de intrare,  $y_k$  reprezintă secvența de ieșire codată, iar  $c_k$  este clock-ul sistemului. Frecvența clock-ului este egală cu  $1/T_b$ , unde  $T_b$  reprezintă durata unui bit „1” sau „0” în secvența de date de intrare.  $y_{k-1}$  reprezintă secvența de ieșire codată, întârziată cu un interval de bit ( $T_b$ ), iar  $x_{k-1/2}$  reprezintă secvența de date de intrare întârziată cu jumătate de interval de bit ( $T_b/2$ ).

În continuare se prezintă un exemplu de sinteză digitală cu ajutorul diagramei Veitch-Karnaugh a codecului RZ.

Legea matematică care descrie codorul RZ este foarte simplă, așa cum rezultă din Tabelul 1. Implementarea decodului RZ, care nu rezultă imediat din ecuația codorului RZ prezentată în Tabelul 1, se poate realiza pe baza sintezei digitale cu diagrame Veitch-Karnaugh. Funcționarea decodului RZ este descrisă de tabloul de adevăr din Tabelul 2, iar diagrama Veitch-Karnaugh corespunzătoare este prezentată în Tabelul 3.

Tabel 2

$x_{k-1}$	$y_k$	$c_k$	$x_k$
0	0	0	0
0	0	1	0
0	1	1	1
0	1	0	∅
1	0	0	1
1	0	1	0
1	1	1	1
1	1	0	∅

Tabel 3

$y_{k-1}c_k$	00	01	11	10
$x_{k-1}$				
0	0	0	1	∅
1	1	0	1	∅

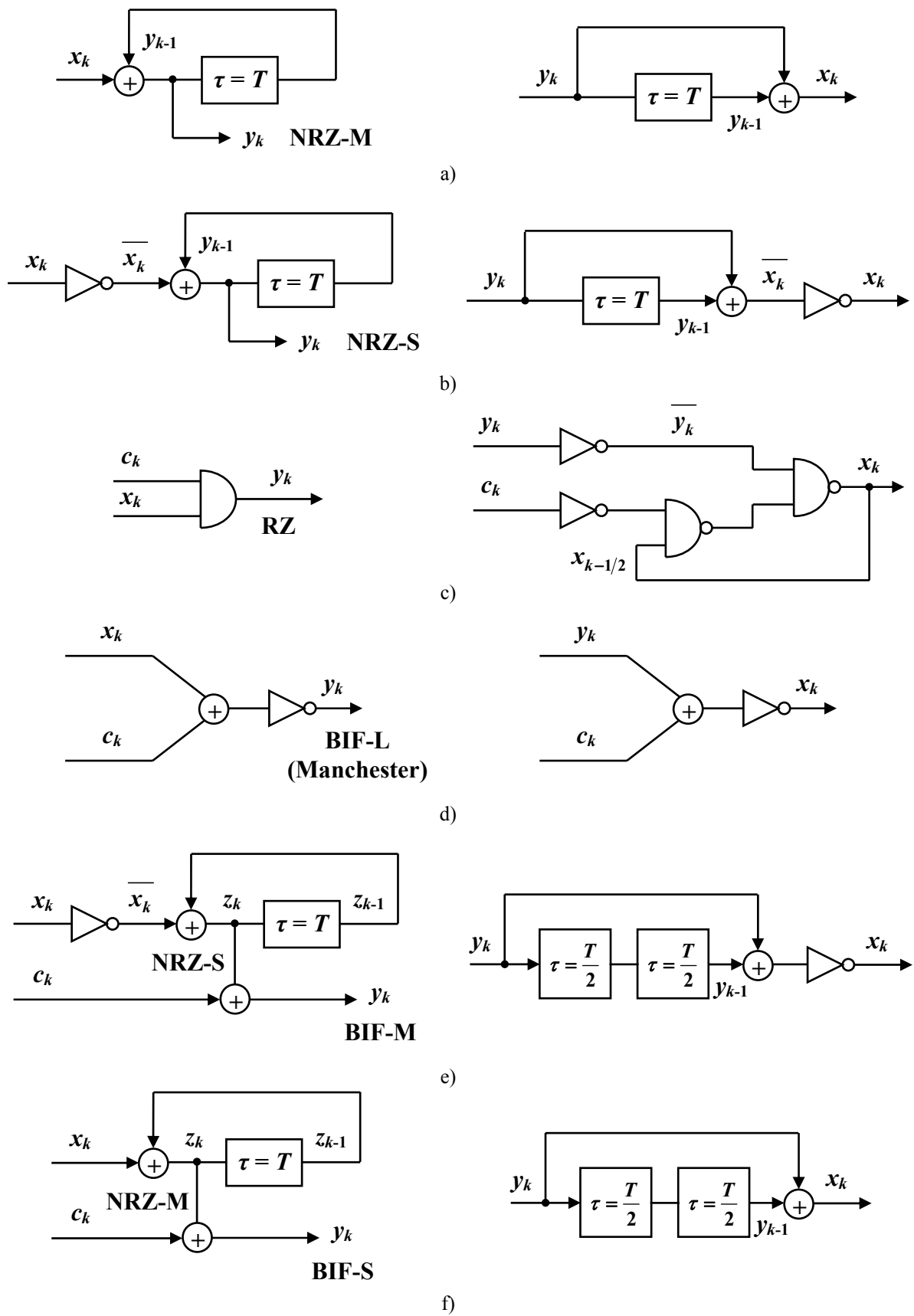


Fig. 2. Implementarea principalelor coduri de linie binare:  
a) NRZ-M; b) NRZ-S; c) RZ; d) BIF-L (Manchester); e) BIF-M; f) BIF-S

Conform diagramei Veitch-Karnaugh ilustrată în Tabelul 2, rezultă următoarea expresie matematică pentru funcția  $x_k$  sintetizată:

$$x_k = y_k + x_{k-1/2} \cdot c_k = y_k \cdot x_{k-1/2} \cdot c_k \quad (1)$$

unde  $x_{k-1/2}$  reprezintă secvența de date întârziată cu jumătate de interval de bit ( $T_b/2$ ).

În codarea NRZ („*Non-Return-to-Zero*”), succesiunile lungi de biți „1” sau „0” conduc la dispariția tranzițiilor din semnal, ceea ce face să dispară în spectrul semnalului codat componenta spectrală pe frecvența de transmisie a datelor,  $f_b = 1/T_b$ , necesară pentru funcționarea corectă a circuitelor de sincronizare de la recepție (refacerea clock-ului).

Această problemă poate fi înlăturată prin codarea RZ („*Return-to-Zero*”), în care intervalul de bit este împărțit în două jumătăți, în prima jumătate biții fiind codați conform convenției unipolare sau polare adoptate, iar în a doua jumătate prin nivelul zero. În cazul semnalului RZ simplu curent, secvențele lungi de biți „0” introduc în continuare probleme de sincronizare.

În transmisiile de date se utilizează noțiunea de *transparență la secvența de biți*, prin care se înțelege proprietatea unui sistem de transmisie de date de a transporta corect orice secvență de date.

Spre deosebire de codurile NRZ și RZ (care sunt *coduri diferențiale*), *codurile bifazice* elimină problema transparenței la secvențele lungi de biți „0” sau „1”, folosind tot o reprezentare binară. Astfel, bitul „1” este reprezentat prin semnal pozitiv pe prima jumătate a intervalului de bit, iar bitul „0” invers. Se obține astfel codul bifazic L (BIF-L) sau *Manchester*, echivalent cu o modulație PSK („*Phase Shift Keying*”) a semnalului de tact de către secvența de date. Codul BIF-L nu permite însă inversarea polarității liniei de transmisie (schimbarea firelor între ele), acest lucru fiind echivalent cu o modificare a semnalului emis, adică obținerea informației negate. Pentru eliminarea ambiguității de fază specifice sistemelor PSK, se recurge la asocierea codării Manchester cu cea diferențială, obținându-se astfel codurile BIF-M și BIF-S.

În codarea BIF-M ambele simboluri binare sunt reprezentate printr-o tranziție la începutul intervalului de bit. Bitul „1”, spre deosebire de bitul „0” prezintă în plus o a doua tranziție la mijlocul intervalului de bit. În codarea BIF-S, convenția este inversată.

## 2. CODUL MILLER

Semnalul codat Miller sau „*delay modulation*” (*bifazic întârziat*) se obține din semnalul BIF-L grupând tranzițiile în grupuri de câte două și suprimând cea de-a doua tranziție. Codarea Miller este echivalentă cu o codare diferențială pentru bitul „1”, cu tranziție la mijlocul intervalului de bit și o codare diferențială pentru bitul „0”, cu tranzițiile la sfârșitul intervalului de bit. Zerourile singulare și cele de la sfârșitul unei secvențe lungi de zerouri sunt codate prin absența tranziției.

Forma de undă codată Miller se poate obține cu un bistabil de tip „T” atacat de semnalul codat „Manchester”, așa cum rezultă din Fig. 3.a). Acesta se aseamănă cu un codor diferențial, de care diferă prin faptul că în cazul codorului diferențial, bistabilul de tip „T” este precedat de un codor RZ negat.

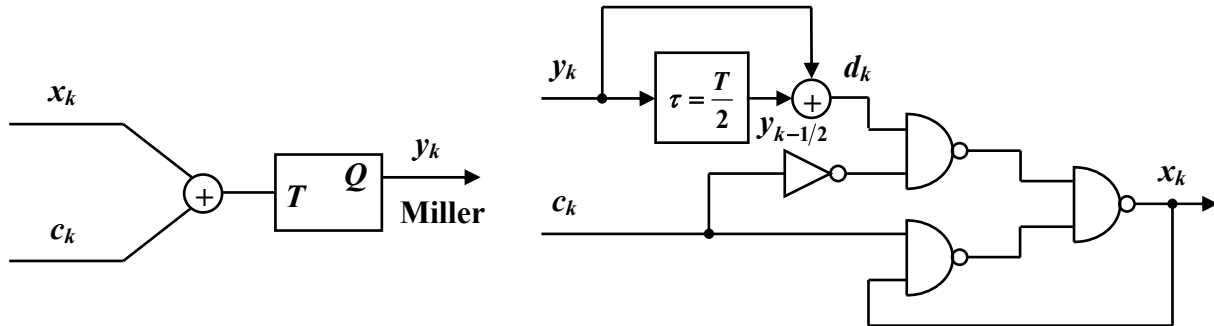


Fig. 3. Implementarea codecului Miller

În codarea Miller se obțin intervale semnificative cu lungimea  $T_b$ ,  $1,5T_b$  sau  $2T_b$ , ceea ce permite obținerea unei viteze de transmisie egală cu cea din cazul codului NRZ, în condițiile asigurării transparenței la orice secvență de biți. În plus, zerourile singulare fiind codate în mod unic, codul permite și detecția erorilor provocate de ambiguitatea de fază.

Decodorul Miller este ilustrat în Fig. 3.b). Prin sumarea modulo 2 a secvenței codată Miller ( $y_k$ ) cu cea întârziată cu jumătate de interval de bit ( $y_{k-1/2}$ ) se obține o detecție diferențială a semnalului, dar care nu este suficientă pentru decodarea semnalului Miller. În urma detecției diferențiale rezultă un semnal RZ pentru biții „1” și un semnal RZ negat pentru biții „0”. Astfel, în urma decodării diferențiale, bitul „1” apare codat „LH” („low-high”), bitul „0” singular și primul bit „0” dintr-o succesiune de zerouri apare codat „LL” („low-low”), iar ceilalți biți „0” apar codati „HL” („high-low”). În continuare, decodarea acestui semnal (notat  $d_k$  în Fig. 3.b)) se realizează cu ajutorul unui decodor RZ.

## 2.1. Eliminarea ambiguității de fază la decodorul Miller

Deoarece tranzițiile din semnalul codat Miller, pentru secvențe lungi de biți „0” sau „1”, apar la sfârșitul bitului, respectiv la mijlocul acestuia, circuitul de sincronizare trebuie să funcționeze pe frecvența  $2f_b$ , care se obține prin divizare din  $f_b$ . Prin urmare, există o ambiguitate de fază potențială a sistemului, introdusă de divizarea de frecvență, care poate conduce la obținerea tactului defazat cu  $180^\circ$  față de situația reală. Pentru decodarea corectă a semnalului Miller este necesar însă tactul cu faza corectă.

Codarea Miller permite înlăturarea ambiguității de fază prin detectarea secvențelor codate unic (secvențe de date de forma 1 0 1) (zerouri singulare), care conduc la apariția în forma de undă codată Miller a intervalelor de lungime  $2T_b$ , cu același nivel (1 sau 0). Dacă intervalele fără tranziții de lungime  $2T_b$  din semnalul codat corespund unei secvențe de date de intrare de forma 0 1 0, atunci secvența de date de intrare este codată prin intermediul codului Miller negat.

### 3. UTILIZAREA PROGRAMULUI ORCAD PENTRU STUDIUL CIRCUITELOR DIGITALE

În continuare, codurile de linie prezentate se vor analiza folosind programul OrCAD, versiunea 10.3. Pentru studiul circuitelor care vor fi ilustrate în continuare se vor utiliza circuite digitale din librăria CD4000 a acestuia.

#### 3.1. Realizarea unui stimul digital

Pentru analiza codurilor de linie se vor utiliza diferite secvențe de date, construite cu ajutorul opțiunii „Stimulus Editor” a programului OrCAD.

După lansarea în execuție a programului „Stimulus Editor” a pachetului OrCAD se solicită un stimul digital de tip „Signal”, care va fi denumit în continuare „date\_cod\_lin”. Din meniul „Plot”/”Axis settings” se setează durata acestui stimul (**30ms**), precum și rezoluția de reprezentare egală cu durata unui bit, de exemplu  $T_b=1\text{ms}$ . Pentru realizarea efectivă a stimulului de date, tranzițiile acestuia se vor alege în conformitate cu următoarea secvență binară:

$$\text{date\_cod\_lin} = 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 1 \quad (2)$$

După realizarea stimulului de mai sus acesta se salvează într-un fișier având aceeași denumire, „date\_cod\_lin.stl”.

În continuare se prezintă procedura de import a stimulului de date digital în simulator, precum și modul de lucru cu acesta.

În primul rând trebuie setată calea spre fișierul care conține stimulul descris mai sus. Acest lucru se realizează în meniul „Configuration Files”/”Stimuls” al profilului de simulare.

Apoi, în „Schematic” se aduce o sursă de tip „DigStim1” din librăria „SOURCSTM” a programului OrCAD, care în prealabil trebuie configurată. În interiorul acestei surse trebuie creat un nou parametru cu ajutorul opțiunii „New Column”. Acest nou parametru se va denumi „STIMULUS” și va avea ca valoare chiar numele stimulului creat mai sus, adică „date\_cod\_lin”.

Un stimul în programul OrCAD poate fi preluat și dintr-un fișier ASCII extern (în care semnalul este reprezentat de doi vectori, de timp și date).

Pentru aceasta se utilizează sursa „VPWL\_FILE” din librăria „SOURCE”, în care se specifică calea spre fișierul de date utilizat. În cele ce urmează, un astfel de stimul se poate obține cu ajutorul programului (realizat în C) „stimul.exe”. Ca date de intrare se utilizează secvența binară care se analizează (de forma (2)). Programul furnizează la ieșire un fișier ASCII ce reprezintă semnalul asociat secvenței binare introduse, în format unipolar sau polar.

Tactul sistemului se implementează cu o sursă de tip „DigClock” din librăria „SOURCE”, la care trebuie setate frecvența și faza. Frecvența clock-ului se alege  $f_b = 1\text{kHz}$ , iar faza dorită se setează prin intermediul parametrilor „STARTVAL” și „OPVAL”, care pentru început, se aleg „1”, respectiv „0”.

În cazul schemelor care utilizează bistabile, acestea trebuie inițializate. Inițializarea bistabilelor se face în profilul de simulare („Simulation settings”) din care se alege meniul „Options”/”Gate-level Simulation”, în care se utilizează opțiunea „Initialize all flip-flops to”. Aici se specifică valoarea dorită pentru inițializarea tuturor bistabilelor din schemă, care poate fi „0”, „1” sau „X” (indiferent). Un bistabil de tip „T” se poate obține cu ajutorul unui bistabil JK (CD 4027) la care intrările „J” și „K” sunt conectate la tensiune de alimentare de 5V.

În Fig. 4, 5 și 6 se prezintă schemele electrice ale principalelor coduri de linie binare, implementate cu circuite digitale din librăria CD4000 a programului OrCAD. În Fig. 7 sunt ilustrate formele de undă ale codurilor de linie binare analizate.

#### 4. MODUL DE LUCRU

1. Se desenează schemele din Fig. 4, 5 și 6, care reprezintă principalele coduri de linie binare, utilizând trei pagini diferite în cadrul aceluiași proiect. Se vor utiliza „conexiuni prin nume” între aceste pagini (din meniul „Place” se alege „Net Alias” pentru denumirea unei conexiuni). În Fig. 4 se prezintă două variante de implementare ale codurilor diferențiale NRZ. Prima variantă de implementare utilizează un bistabil de tip „D” (CD 4013) și rezultă direct din schemele principiale ilustrate în Fig. 2.a) și b). A doua variantă de implementare utilizează un bistabil „T”, care este precedat de un codor RZ negat. În acest caz tranziția apare pe mijlocul intervalului de bit, adică se produce o întârziere cu  $T/2$  a semnalului codat față de varianta anterioară, iar codul este cunoscut sub denumirea de NRZI („Non-Return-to-Zero Inverted”). În Fig. 5 se prezintă codorul diferențial RZ, precum și codurile bifazice: BIF-L (Manchester), BIF-M și BIF-S, iar în Fig. 6, codorul Miller.

2. Se construiește stimulul prezentat în relația (2), care se utilizează pentru analiza circuitelor de codare/decodare ilustrate în Fig. 4, 5 și 6. Se verifică corectitudinea obținerii formelor de undă reprezentate în Fig. 7.



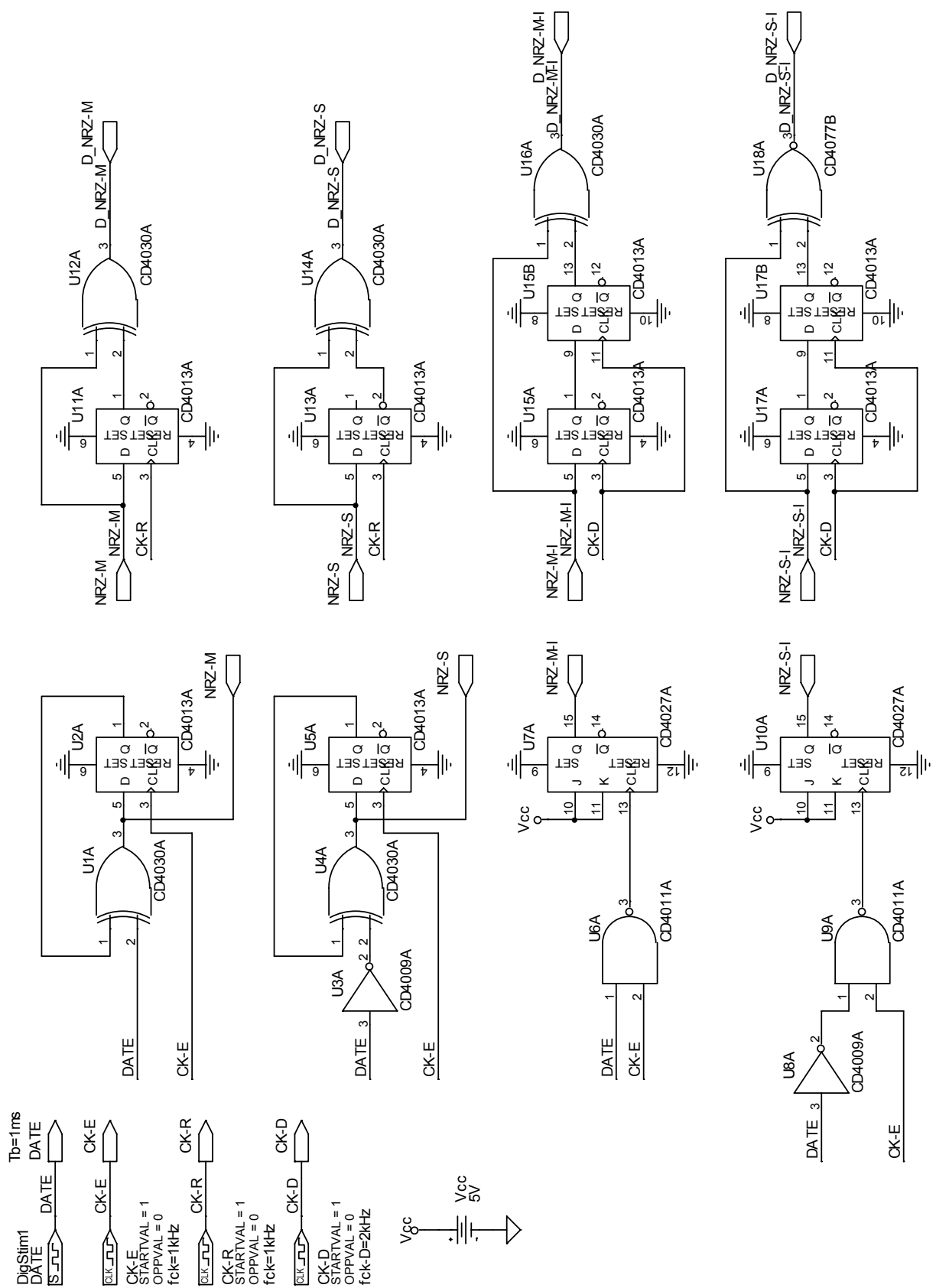


Fig. 4. Codecări de linie binară diferențială: NRZ-M, NRZ-S, NRZ-M-I, NRZ-S-I

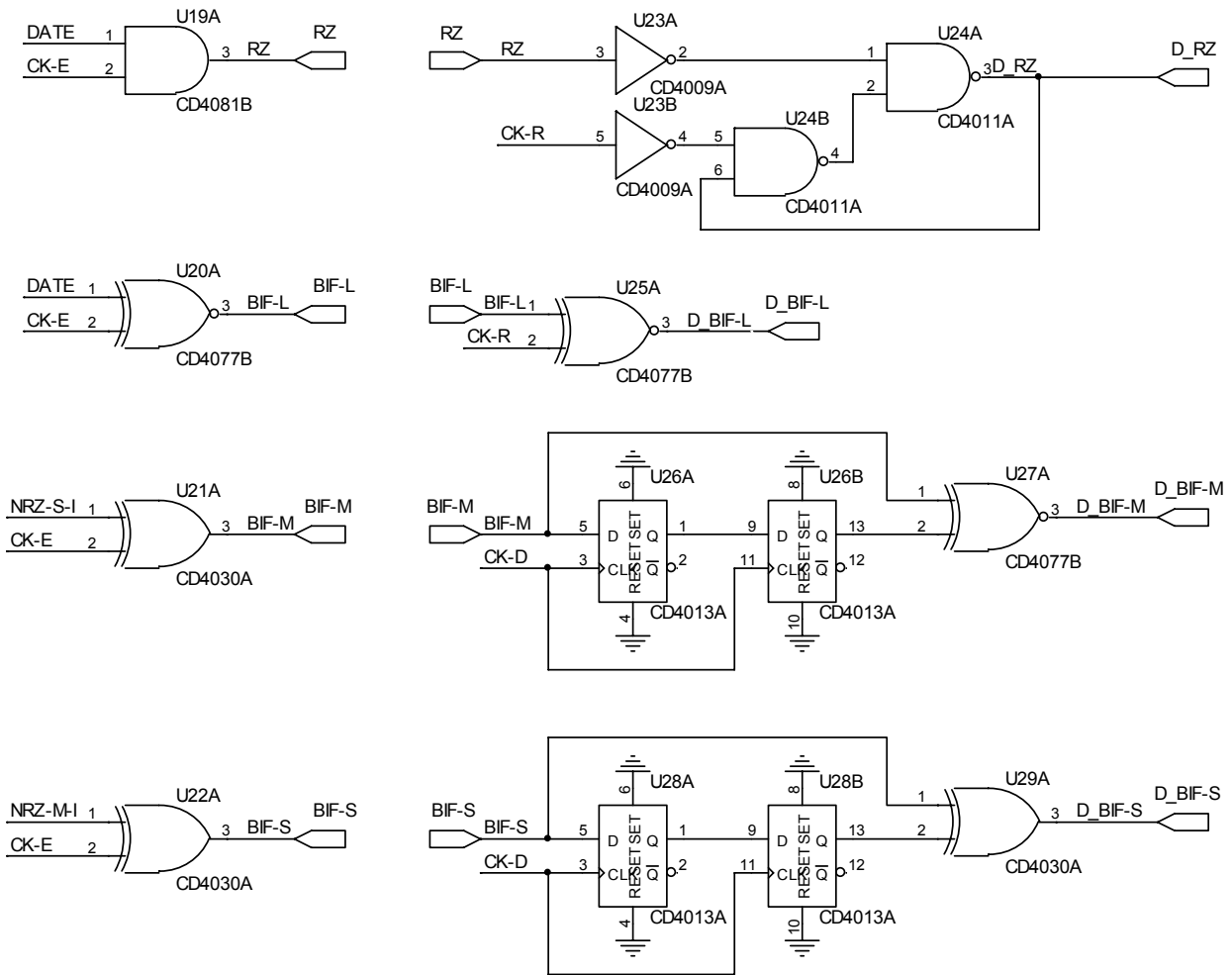


Fig. 5. Codecuri de linie binare: diferențial - RZ; bifazice - BIF-L (Manchester), BIF-M, BIF-S

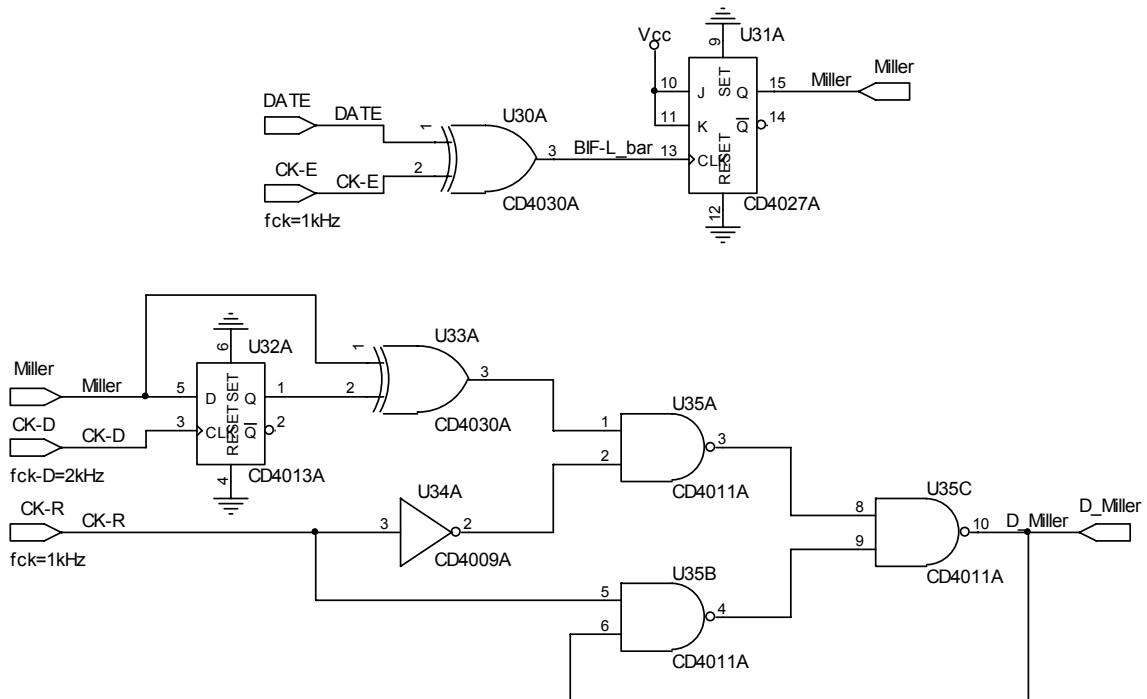


Fig. 6. Codecul Miller

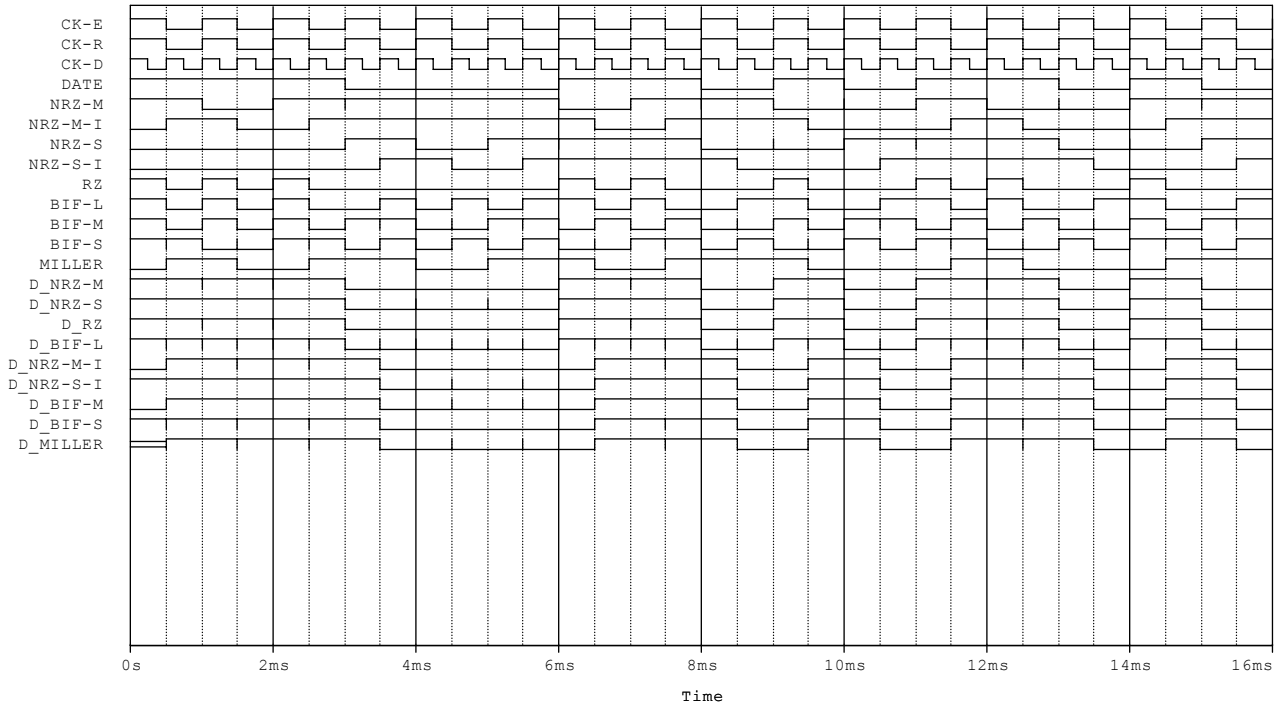


Fig. 7. Formele de undă ale principalelor coduri de linie binare

3. Să se verifice legile de codare (din Tabelul 1) pentru fiecare cod prezentat și să se construiască diagramele de stări de tranziții corespunzătoare.

4. Să se traseze spectrele de frecvențe ale codurilor analizate și să se compare performanțele acestora.

5. Explicați în ce constă ambiguitatea de fază la un cod de linie și precizați la care dintre codurile studiate se manifestă.

6. Explicați cum rezolvă codurile bifazice problema transparenței la secvențele lungi de biți.

7. Explicați avantajele codării Miller față de celelalte coduri studiate.

8. Explicați de ce în cazul codurilor NRZ-M-I, NRZ-S-I, BIF-M, BIF-S și Miller, sincronizarea la recepție trebuie făcută pe frecvența  $2f_b$ .

9. Se realizează schema din Fig. 8, în care se prezintă un decodor Miller, împreună cu o logică digitală suplimentară, care are rolul de a înlătura ambiguitatea de fază a acestui semnal. Circuitul realizează mai întâi o detecție diferențială a semnalului codat Miller, obținându-se semnalul notat „DIF” în Fig. 8. Urmăriți forma de undă a semnalului „DIF”, care codează secvența de date, astfel:

- bitul „1” este codat „LH”;
- bitul „0” singular și primul bit „0” dintr-o succesiune de zerouri apar codati „LL”;
- ceilalți biți „0” sunt codati „HL”.

Metoda de eliminare a ambiguității de fază a semnalului codat Miller realizată de circuitul reprezentat în Fig. 8 se bazează pe observația că zerourile singulare din secvența de date de intrare sunt codate în mod unic (în semnalul codat Miller acestea apar ca intervale fără tranziții de lungime  $2T_b$ ), fapt ce permite deosebirea lor. Aceste zerouri singulare din secvența de date de intrare sunt detectate cu ajutorul porții U<sub>8</sub> a circuitului din Fig. 8. Pentru exemplificarea funcționării, construiți un stimul de date, identic cu „date\_cod\_lin”, dar care să prezinte la început secvența „1 0 1”. Simulați schema din Fig. 8, utilizând această secvență de date de intrare, dar considerând faze opuse pentru tactul decodorului, „CK-R”. Urmărind formele de undă obținute în cazul acestor două simulări, explicați care trebuie să fie faza corectă a tactului care se aplică la intrarea decodorului „Miller”, notat în schemă „CK-M”, pentru decodarea corectă a secvenței de date de la intrare. Explicați, de asemenea, cum intervine logica digitală pentru corectarea fazei tactului refăcut de la recepție, astfel încât faza semnalului de clock „CK-M” să fie întotdeauna cea corectă pentru funcționarea optimă a decodorului Miller. Ce se întâmplă cu secvența decodată de la receptor, dacă se inversează faza tactului de la emițător, „CK-E”?

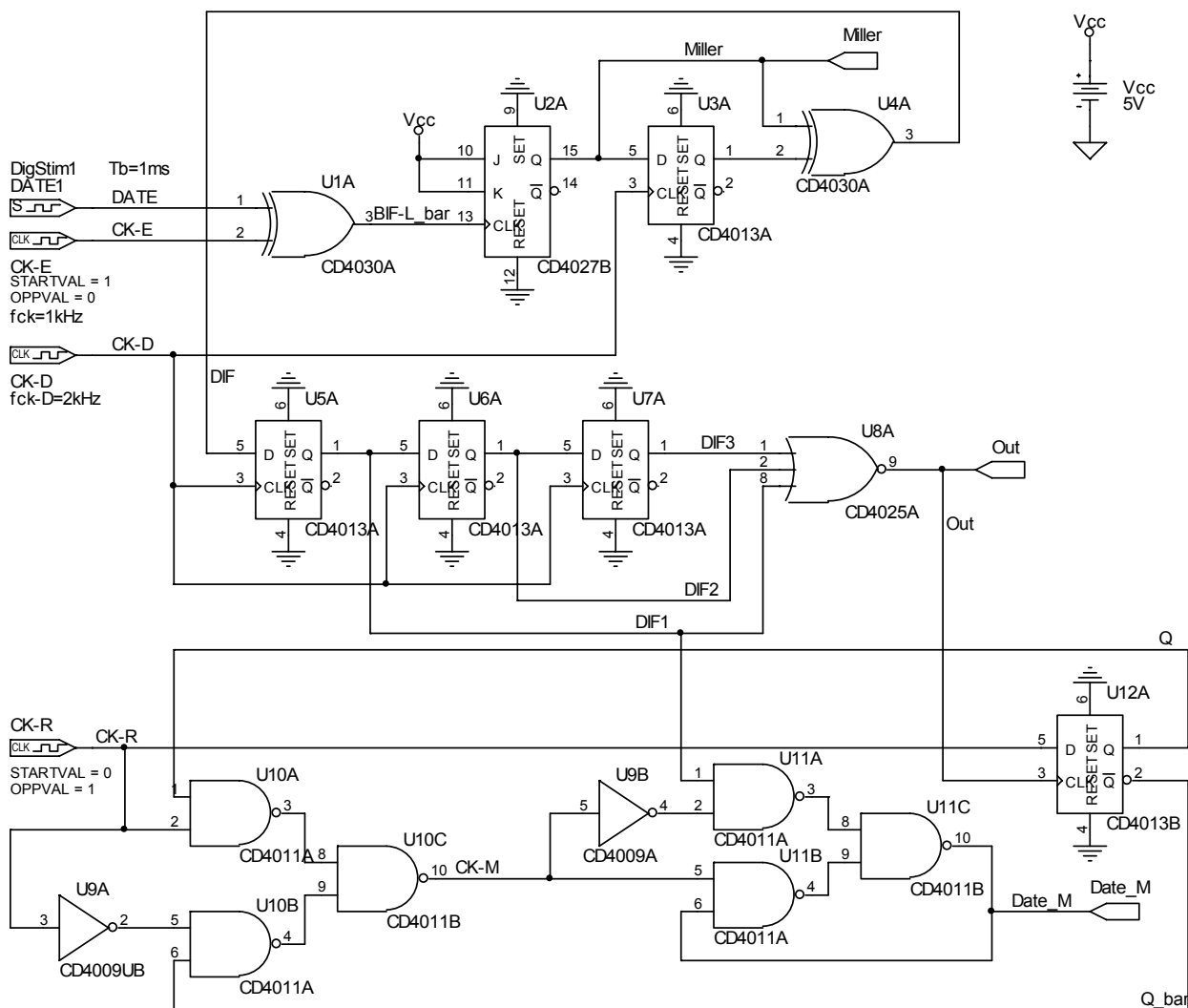


Fig. 8. Circuit pentru eliminarea ambiguității de fază la decodorul Miller

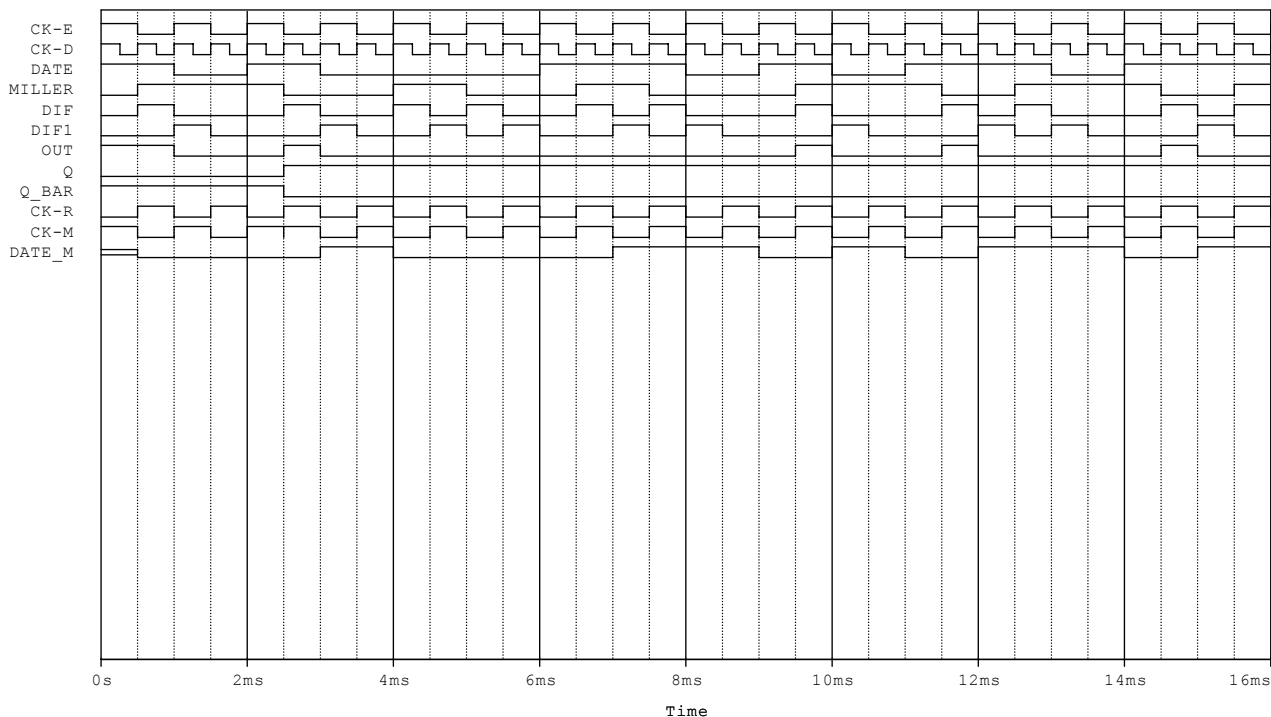


Fig. 9. Formele de undă ale decodului Miller care elimină ambiguitatea de fază

10. Se consideră circuitul ilustrat în Fig. 10. Reprezentați formele de undă CK, DATE, m,  $M_i$ ,  $M_q$ , W, Z și P pentru o secvență de date binară de forma 1 1 1 1 0 0 1 0 1 0 1 1 0 0 0 0 1. Să se deducă:

- În ce cod se obține semnalul de ieșire P;
- Ce relație există între semnalele m,  $M_i$  și  $M_q$ ;
- Cum se poate obține semnalul m dacă se cunosc  $M_i$  și  $M_q$ ;
- Ce valoare are produsul W·Z și bazat pe această observație să se explice cum se pot defini semnalele W și Z;
- Diagramele de stări de tranziții ale semnalelor W și Z, considerând toate stările logice posibile.

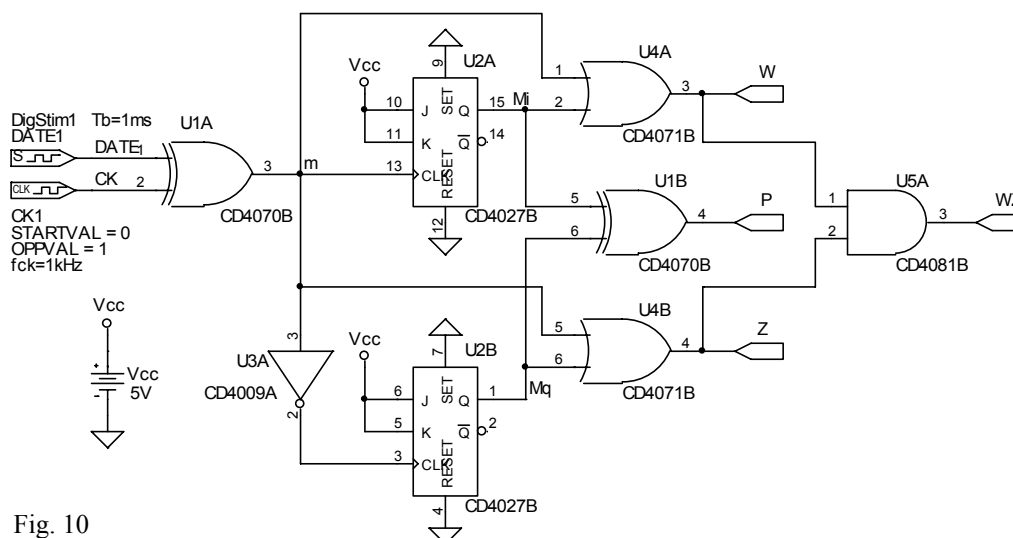


Fig. 10