

Dan NICULA

ELECTRONICĂ DIGITALĂ

Carte de învățatură 2.0



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

2015

Lecția 8

Codificatoare și decodificatoare

8.1 Noțiuni teoretice

Decodicatorul $N : 2^N$ are N intrări de date pe care primește codul binar al unui număr în domeniul între 0 și $2^N - 1$. La un moment dat, o singură ieșire a decodicatorului este activă: cea cu indexul egal cu numărul prezentat la intrare. Starea activă a ieșirii poate fi în 0 sau în 1. Se poate considera că decodicatorul generează pe cele 2^N ieșiri toți mintermii celor N variabile prezentate la intrare.

Un decodicator poate avea o intrare de validare. Dacă intrarea de validare este activată, decodicatorul funcționează și prezintă la ieșire valoarea decodificată a intrării (o singură ieșire activată). Dacă intrarea de validare nu este activată, toate ieșirile decodicatorului sunt în starea inactivă. Intrarea de validare oferă suport pentru extinderea capacității de decodificare.

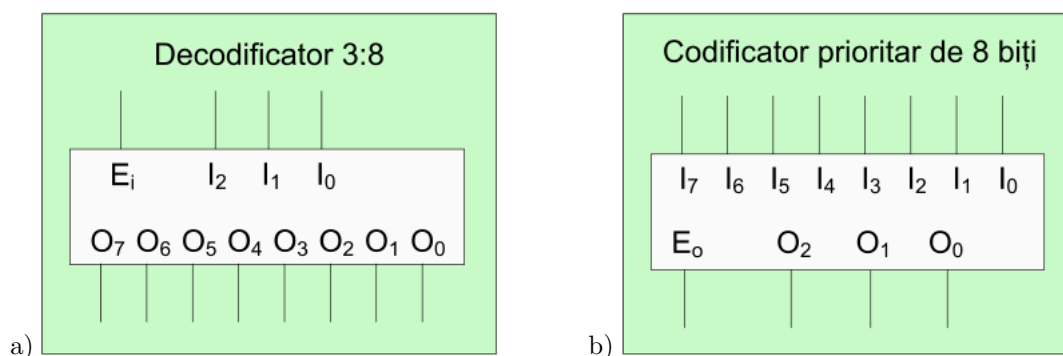


Figura 8.1 a) Simbol de decodicator 3 : 8, b) Simbol de codificator prioritar de 8 biți.

Codificatorul prioritar de 2^N biți are 2^N intrări și N ieșiri. La ieșire se prezintă codul binar al celei mai prioritare intrări activate. Codificatorul poate avea o ieșire care semnalează apariția pe intrare a unei configurații valide (cu cel puțin un bit egal cu 1).

Orice funcție logică cu N intrări poate fi implementată cu următoarele resurse: decodicator $N : 2^N$ și o poartă logică de maximum 2^{N-1} intrări.

- Pentru implementarea unei funcții de N variabile cu un DCD $N : 2^N$ funcția nu trebuie minimizată deoarece în nivelul AND sunt produși toți mintermii de N variabile.
- Circuitul DCD $N : 2^N$ produce la ieșire toți cei 2^N mintermi de N variabile. Orice funcție poate fi scrisă sub forma canonică disjunctivă (FCND) ca o "sumă de mintermi". Existând generați toți cei 2^N mintermi la ieșirile decodicatorului, mintermii necesari funcției sunt selectați într-un nivel de logică cu o poartă OR.



- În cazul în care decodificatorul are ieșirile negare (ieșiri active în stare 0), mintermi sunt generați negați. Din acest motiv, pentru implementarea unei funcții exprimate prin FCND în exteriorul decodificatorului trebuie adăugată o poartă NAND.
- Pentru cazul în care funcția însumează un număr de mintermi mai mare decât jumătatea celor existenți 2^{N-1} , se poate face sinteza funcției negare \overline{F} , deoarece aceasta se poate exprima cu mai puțini mintermi, și ulterior se neagă ieșirea.
- Pentru cazul când decodificatorul are intrările active în 1, se poate exprima funcția astfel încât să necesite în exterior o poartă NOR.

În concluzie, implementarea optimă a unei funcții exprimate în FCND cu circuit decodicator se realizează astfel:

- Dacă decodificatorul are ieșirile active în 1, iar funcția un număr de mintermi mai mic decât jumătate, colectarea într-o poartă OR a mintermilor existenți.
- Dacă decodificatorul are ieșirile active în 0, iar funcția un număr de mintermi mai mic decât jumătate, colectarea într-o poartă NAND a mintermilor existenți.
- Dacă decodificatorul are ieșirile active în 1, iar funcția un număr de mintermi mai mare decât jumătate, colectarea într-o poartă NOR a mintermilor inexistenți.
- Dacă decodificatorul are ieșirile active în 0, iar funcția un număr de mintermi mai mare decât jumătate, colectarea într-o poartă AND a mintermilor inexistenți.

8.2 Pentru cei ce vor doar să promoveze examenul

1. Să se implementeze cu decodicator 3:8 funcțiile:
 - a) $F_a(A, B, C) = \sum(0, 1, 7)$
 - b) $F_b(A, B, C) = \sum(0, 1, 2, 5, 7)$
2. Să se implementeze cu decodicator 4:16 funcțiile:
 - a) $F_a(A, B, C, D) = \sum(0, 1, 11)$
 - b) $F_b(A, B, C, D) = \sum(3, 4, 12, 14)$
3. Să se implementeze cu un singur circuit DCD 3:8 (și porți logice adiționale) un circuit logic combinațional cu ieșiri multiple definit de funcțiile:

$$F_a = m_0 + m_3 + m_5 + m_7$$

$$F_b = m_1 + m_2 + m_4 + m_5$$

$$F_c = m_3 + m_4 + m_6$$

$$F_d = m_1 + m_2 + m_5 + m_6 + m_7$$

8.3 Pentru cei ce vor să învețe

1. Să se implementeze cu decodicator 3:8 funcția: $F(A, B, C) = \sum(0, 1, 3, 5, 7)$.

Soluție

Circuitul DCD 3 : 8 produce la ieșire toți cei $2^3 = 8$ mintermi de 3 variabile. Orice funcție poate fi scrisă sub forma canonică disjunctivă (FCND) ca o "sumă de mintermi". Existând generați toți cei 8 mintermi la ieșirile decodificatorului, mintermi necesari funcției sunt selectați într-un nivel de logică cu o poartă OR:

$$F(A, B, C) = \sum(0, 1, 3, 5, 7) = m_0 + m_1 + m_3 + m_5 + m_7$$

În cazul în care decodificatorul are ieșirile negare (ieșiri active în stare 0), mintermi sunt generați negați. Din acest motiv, pentru implementarea unei funcții exprimate prin FCND în exteriorul decodificatorului trebuie adăugată o poartă NAND:



$$F(A, B, C) = \sum(0, 1, 3, 5, 7) = \overline{m_0 + m_1 + m_3 + m_5 + m_7} = \overline{m_0 \cdot \overline{m_1} \cdot \overline{m_3} \cdot \overline{m_5} \cdot \overline{m_7}}$$

Pentru cazul în care funcția însumează un număr de mintermi mai mare decât jumătatea celor existenți $2^{3-1} = 2^2 = 4$, se poate face sinteza funcției negate \overline{F} , deoarece aceasta se poate exprima cu mai puțini mintermi, și ulterior se neagă ieșirea.

$\overline{F}(A, B, C) = \overline{m_0 + m_1 + m_3 + m_5 + m_7} = m_2 + m_4 + m_6 = \overline{m_2 + m_4 + m_6} = \overline{m_2 \cdot \overline{m_4} \cdot \overline{m_6}}$. Rezultă, că în exterior trebuie adăugată o poartă AND cu trei intrări:

$$F(A, B, C) = \overline{m_2 \cdot \overline{m_4} \cdot \overline{m_6}}$$

Pentru cazul când decodificatorul are intrările active în 1, se poate exprima funcția astfel încât să necesite în exterior o poartă NOR:

$$F(A, B, C) = \overline{m_2 + m_4 + m_6}$$

Figura 8.2 prezintă variantele de implementare cu decodificator a funcției $F(A, B, C) = \sum(0, 1, 3, 5, 7)$.

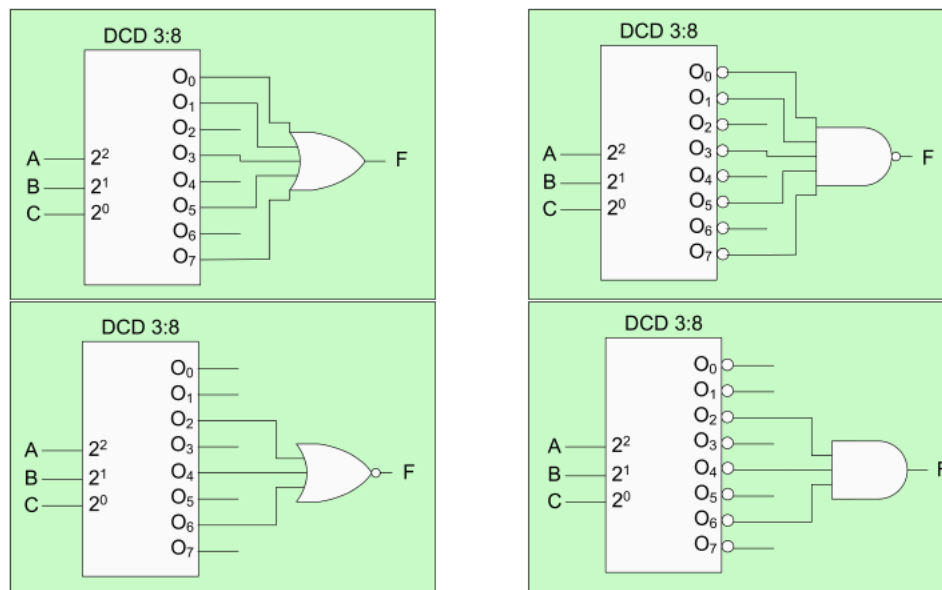


Figura 8.2 Implementarea funcției $F(A, B, C) = \sum(0, 1, 3, 5, 7)$ cu circuit decodificator, pentru problema 1.

2. Care din următoarele funcții este implementată pe circuitul din figura 8.3?

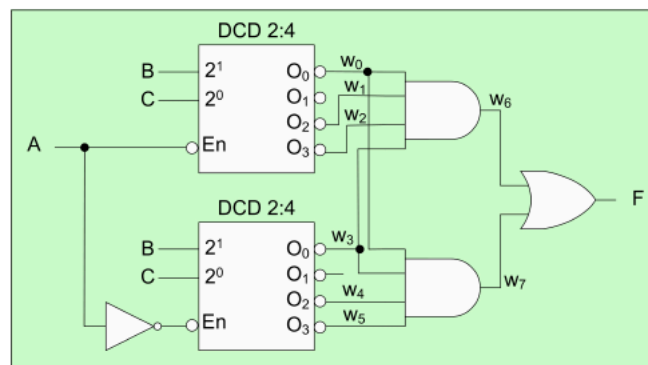


Figura 8.3 Circuit cu decodificatoare, pentru problema 2.

- a) $F(A, B, C) = \sum_0^7(0, 1, 3, 5, 6, 7)$
- b) $F(A, B, C) = \prod_0^7(1, 2, 3, 5, 6, 7)$
- c) $F(A, B, C) = \sum_0^7(1, 2, 3, 5, 6, 7)$
- d) $F(A, B, C) = \sum_0^7(0, 1, 2, 4, 6, 7)$
- e) $F(A, B, C) = \sum_0^7(1, 5)$



Soluție

Tabelul de adevăr al funcției este:

| A | B | C | w_0 | w_1 | w_2 | w_3 | w_4 | w_5 | w_6 | w_7 | F |
|-----|-----|-----|-------|-------|-------|-------|-------|-------|-------|-------|-----|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |

$$w_0 = \overline{A \cdot B \cdot C} = \overline{m_0} = M_0$$

$$w_1 = \overline{A \cdot B \cdot C} = \overline{m_2} = M_2$$

$$w_2 = \overline{A \cdot B \cdot C} = \overline{m_3} = M_3$$

$$w_3 = \overline{A \cdot B \cdot C} = \overline{m_4} = M_4$$

$$w_4 = \overline{A \cdot B \cdot C} = \overline{m_6} = M_6$$

$$w_5 = \overline{A \cdot B \cdot C} = \overline{m_7} = M_7$$

$$w_6 = w_0 \cdot w_1 \cdot w_2 \cdot w_3 = M_0 \cdot M_2 \cdot M_3 \cdot M_4$$

$$w_7 = w_0 \cdot w_3 \cdot w_4 \cdot w_5 = M_0 \cdot M_4 \cdot M_6 \cdot M_7$$

$$F = w_6 + w_7 = M_0 \cdot M_2 \cdot M_3 \cdot M_4 + M_0 \cdot M_4 \cdot M_6 \cdot M_7 = M_0 \cdot M_4 \cdot (M_2 \cdot M_3 + M_6 \cdot M_7) = M_0 \cdot M_4 \cdot ((M_2 + M_6) \cdot (M_2 + M_7) \cdot (M_3 + M_6) \cdot (M_3 + M_7)) = M_0 \cdot M_4$$

În prelucrările analitice s-a aplicat duala axiomei de distributivitate și s-a ținut cont că $M_i + M_j = 1, \forall i \neq j$. S-au notat $m(A, B, C)$ și $M(A, B, C)$ mintermii și maxtermii celor 3 variabile de intrare.

Din tabelul de adevăr și din calcule analitice rezultă că funcția implementată de circuit este:

$$F(A, B, C) = \prod(0, 4) = \sum(1, 2, 3, 5, 6, 7)$$

Deci, răspunsul corect este c).

3. Să se implementeze cu circuite DCD 4:16 un circuit logic combinațional cu ieșiri multiple definit de funcțiile:

$$F_1 = m_0 + m_3 + m_5 + m_7$$

$$F_2 = m_1 + m_2 + m_4 + m_5 + m_8 + m_{11} + m_{12} + m_{14} + m_{15}$$

$$F_3 = m_3 + m_4 + m_6 + m_{12} + m_{14} + m_{15}$$

$$F_4 = m_1 + m_2 + m_5 + m_6 + m_7 + m_8 + m_9 + m_{15}$$

Soluție

Pentru implementarea unui CLC cu ieșiri multiple se poate folosi un singur decodificator. Pentru fiecare ieșire se utilizează câte o poartă OR care unește mintermii prezenți în expresia unei funcții.

4. Să se implementeze următoarele funcții cu circuit decodificator de 4 biți. Implementați aceleași funcții utilizând un decodificator de 3 biți și un multiplexor 2:1.

a) $F_a(A, B, C) = \sum(0, 2, 5, 7)$

b) $F_b(A, B, C, D) = \prod(2, 3, 4, 7)$

c) $F_c(A, B, C, D) = \sum(0, 3, 5, 6, 9, 10, 11, 12, 13)$

d) $F_d(A, B, C, D) = \prod(2, 3, 6, 7, 8, 9, 13, 14, 15)$

e) $F_e(A, B, C, D) = \sum(2, 5, 7)$

f) $F_f(A, B, C, D) = \sum(3, 6, 9, 12, 14, 15)$

g) $F_g(A, B, C, D) = \sum(0, 1, 2, 3, 4, 15)$

h) $F_h(A, B, C, D) = \sum(0, 1, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)$

Soluție

Implementarea cu decodificator de 4 biți a unei funcții de 4 intrări se face fără minimizare, prin conectarea intrărilor pe intrările decodicatorului și conectarea într-o poartă OR a ieșirilor al căror index apare în expresia FCD a funcției. În funcție de numărul de mintermi care apar în reprezentarea FCD, se poate opta pentru implementarea negatei funcției și negarea adițională la final, ca în figura 8.2.

Dacă funcția are doar 3 intrări, se poate conecta o constantă pe cea mai prioritară intrare a decodicatorului.

O funcție cu 4 intrări se poate rescrie sub forma:

$$F(A, B, C, D) = A \cdot H(B, C, D) + \overline{A} \cdot G(B, C, D)$$

Funcțiile $H(B, C, D)$ și $G(B, C, D)$ se pot obține cu același decodificator (și porți OR individuale). Ieșirile porților OR se vor conecta pe intrările de date ale unui multiplexor 2:1 a cărui selecție este conectată la intrarea A.



5. Implementați următorul sistem de funcții cu un decodificator și un număr minim de porți logice suplimentare.

a)

$$F_1(A, B, C) = \sum(2, 4, 7),$$

$$F_2(A, B, C) = \sum(0, 3),$$

$$F_3(A, B, C) = \sum(0, 2, 3, 4, 7)$$

b)

$$F_1(A, B, C) = \overline{A} \cdot \overline{B} \cdot \overline{C} + A \cdot C,$$

$$F_2(A, B, C) = A \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot B,$$

$$F_3(A, B, C) = \overline{A} \cdot \overline{B} \cdot C + A \cdot B$$

6. Să se implementeze următoarele funcții cu circuit decodificator și porți OR suplimentare:

$$F_1 = \overline{A+B} + A \cdot B \cdot \overline{C}$$

$$F_2 = \overline{A+B} + \overline{A} \cdot B \cdot C$$

$$F_3 = A \cdot B \cdot C + \overline{A+B}$$

7. Proiectați structura internă a unui decodificator de 2 biți cu intrare de validare, implementată cu porți NOR și NOT.

8. Proiectați un decodificator de 2 biți cu intrare de validare implementat exclusiv cu porți NOR.

9. Implementați un circuit codificator cu prioritate de 4 biți cu ieșire de validare cu circuit decodificator cu intrare de validare și un număr minim de porți logice adiționale. Implementați același circuit cu porți logice NAND pe două nivele.

Soluție

| D_3 | D_2 | D_1 | D_0 | C_1 | C_0 | V |
|-------|-------|-------|-------|-------|-------|-----|
| 1 | X | X | X | 1 | 1 | 1 |
| 0 | 1 | X | X | 1 | 0 | 1 |
| 0 | 0 | 1 | X | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | X | X | 0 |

10. Proiectați circuitul care comandă un afișaj cu 7 segmente pe baza unui cod BCD. Implementați circuitul cu decodificator.

Soluție

Se conectează variabilele de intrare pe intrările de selecție ale decodicatorului. Pentru fiecare din cele 7 funcții de ieșire (asociate segmentelor) va fi necesară câte o poartă OR.

11. Proiectați un convertor din cod BCD în cod Gray utilizând un circuit decodificator.

Soluție

Circuitul are 4 intrări și 4 ieșiri. Se utilizează un decodificator 4:16 cu care se obțin toți mintermii variabilelor de intrare în cod BCD. Cele 4 ieșiri se obțin din conectare prin porți OR a mintermilor corespunzători funcțiilor de transfer.

12. Proiectați un decodificator 4:16 implementat cu un inversor pe post de decodificator 1:2 și două decodificatoare 3:8 cu validare.

Soluție

Cei mai puțin semnificativi 3 biți se conectează pe intrările de selecție ale celor două decodificatoare 3:8. Cele 16 ieșiri ale celor două decodificatoare vor reprezenta ieșirile decodicatorului 4:16. Cel mai semnificativ bit al intrării decodicatorului se "decodifică" (bitul însuși și bitul negat) și se aplică pe intrările de validare ale celor două decodificatoare.

13. Proiectați cu scheme bloc un decodificator de 5 biți utilizând 4 decodificatoare de 3 biți cu intrare de validare și un decodificator de 2 biți.

Soluție

Cei mai semnificativi 2 biți se decodifică cu decodicatorul de 2 biți și se obțin 4 semnale de validare folosite pentru validarea a 4 decodificatoare de 3 biți, cu intrările de selecție conectate împreună la cei mai puțin semnificativi biți ai intrării. Se obțin $8 \times 4 = 32$ ieșiri de la cele 4 decodificatoare de 3 biți. Circuitul este prezentat în figura 8.4.



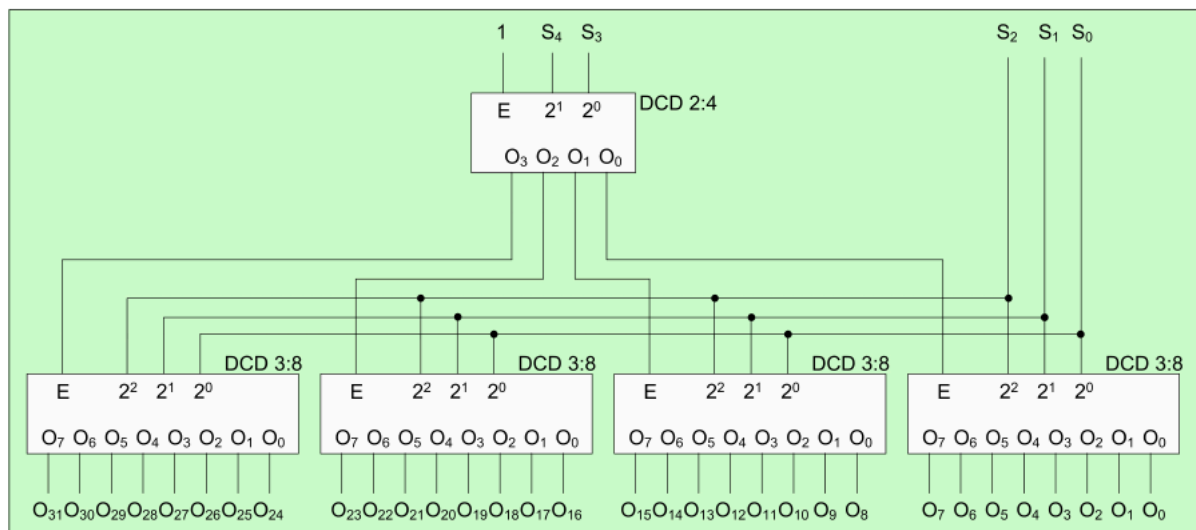


Figura 8.4 Structură de decodificator de 5 biți.

14. Utilizând circuite decodificator DCD 3:8 cu validare să se realizeze o structură care implementează un circuit decodificator DCD 5:32.

Soluție

Circuitul are 5 intrări de selecție. Cei mai semnificativi 2 biți se decodifică pentru a se obține $2^2 = 4$ semnale de validare. Pe al doilea nivel logic, 4 decodificatoare primesc câte un semnal de validare de la decodicatorul de pe primul nivel și generează fiecare 8 ieșiri. Concatenarea celor 8 ieșiri de la cele 4 decodificatoare vor reprezenta cele 32 de ieșiri ale decodicatorului. Circuitul este prezentat în figura 8.5.

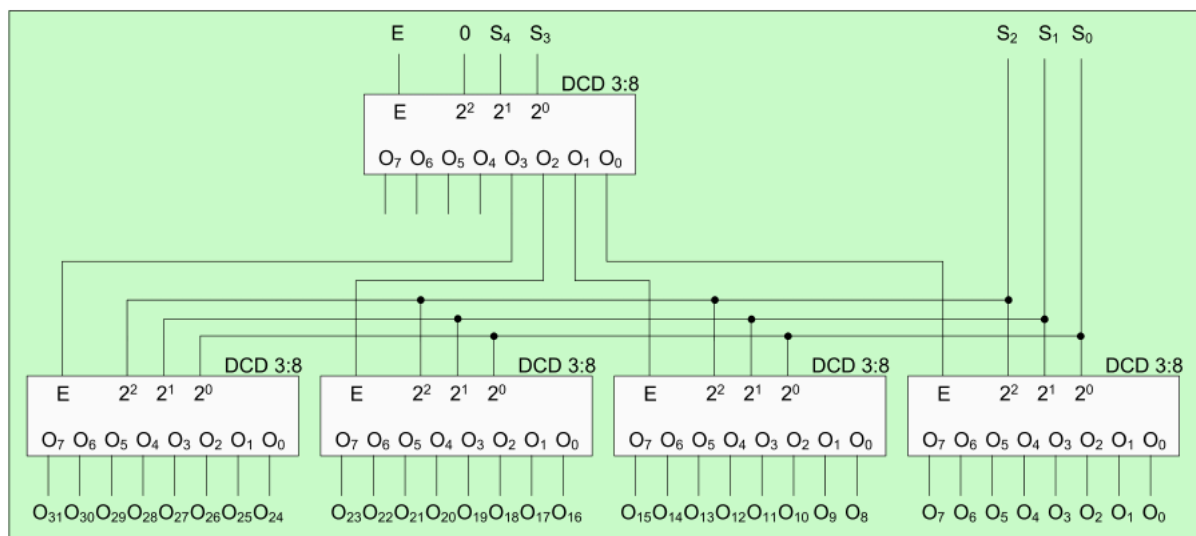


Figura 8.5 Decodificator 5:32 realizat cu decodificatoare 3:8 (problema 14).

15. Să se realizeze structurile următoarelor decodificatoare utilizând decodificatoare de dimensiuni mai mici, astfel:
- decodificator 6:64 cu decodificatoare 4:16;
 - decodificator 4:16 cu decodificatoare 3:8;
 - decodificator 2:4 cu decodificatoare 1:2.

Soluție

Circuitele cerute la **b)** și **c)** sunt prezentate în figura 8.6.

16. Construiți un decodificator de 4 biți utilizând 5 decodificatoare de 2 biți cu intrare de validare.



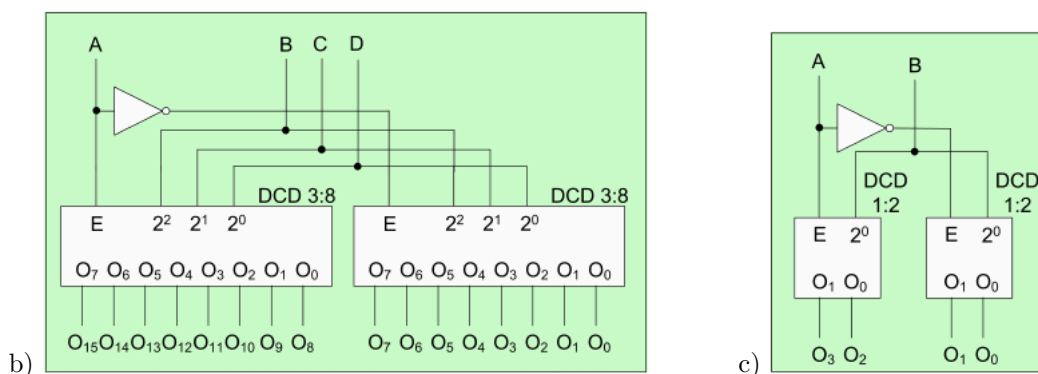


Figura 8.6 Structuri de decodificare, (problema 15-b, c).

17. Să se implementeze operația: $Y = 2^X$ utilizând un decodificator de 3 biți. Variabila de intrare $X \in [0, 7]$.

Soluție

Decodificatorul implementează funcția $Y_{[7:0]} = 2^{X_{[2:0]}}$. Biții de intrare $X_{[2:0]}$ se conectează pe intrările de selecție iar ieșirile $Y_{[7:0]}$ se obțin de la ieșirile decodificatorului având același index.

8.4 Pentru cei ce vor să devină profesioniști

Decodificator cu validare (cod Verilog)

```
parameter N;           // dimensiunea generica a decodificatorului
input  [N-1:0]    I;    // intrare de date
output [(1<<N)-1:0] O;  // iesire
input             Ei;   // intrare de validare

assign O = Ei << I;
```

Codificator prioritar de 4 biți (cod Verilog)

```
input  [3:0]    I; // intrare de date
output [1:0]    O; // iesire
output          Eo; // iesire de validare

always @ (I) begin
    if (I[3]) O = 'd3; else
    if (I[2]) O = 'd2; else
    if (I[1]) O = 'd1; else
    if (I[0]) O = 'd0; else
        O = 'bx;
end

assign Eo = |I; // OR pe toate intrarile
```

1. Să se proiecteze un circuit codificator care determină indexul intrării activate cu prioritatea cea mai scăzută.

Soluție

Dacă la intrările unui codificator se conectează biții în ordinea ponderilor, atunci circuitul codificator prioritar va prezenta pe ieșire indexul intrării celei mai semnificative activate (figura 8.7-a).

Dacă la intrările unui codificator se conectează biții în ordinea inversă a ponderilor, atunci circuitul codificator prioritar va prezenta pe ieșire indexul intrării celei mai puțin semnificative activate, în cod complementar. Soluția constă în conectarea biților de date în ordine inversă și negarea ieșirii, așa ca în figura 8.7-b.



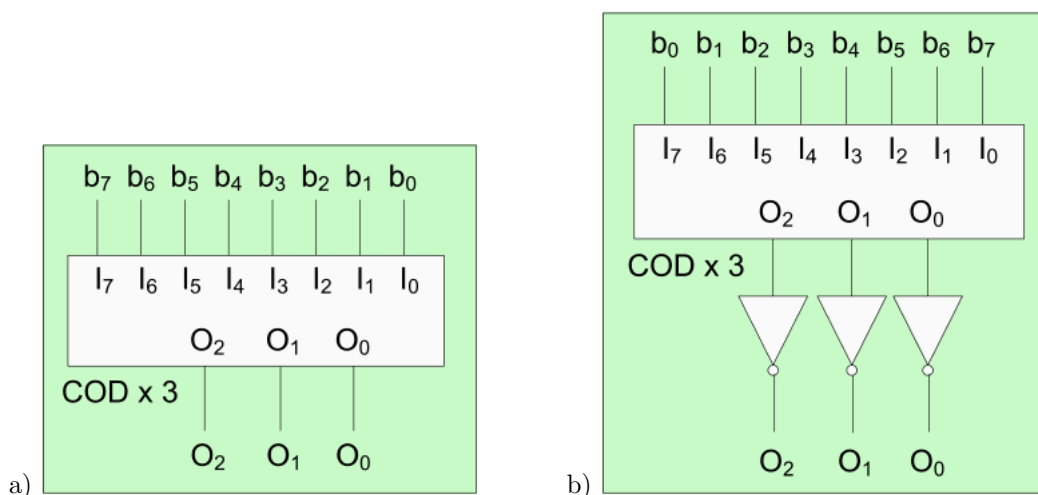


Figura 8.7 Circuite cu codificator: **a)** determinarea indexului intrării cu prioritate cea mai mare și **b)** determinarea indexului intrării cu prioritate cea mai mică.

De exemplu, dacă $b_2 = b_3 = 1$, restul biților fiind 0, intrarea cea mai prioritară a codificatorului este I_5 , ceea ce va genera pe ieșire valoarea 101, negată $010 = 2|_{10}$.

Dacă $b_3 = b_4 = b_7 = 1$, restul biților fiind 0, intrarea cea mai prioritară a codificatorului este I_4 , ceea ce va genera pe ieșire valoarea 100, negată $011 = 3|_{10}$.

- Utilizând două circuite codificatoare prioritare de 8 biți să se realizeze o structură de codificator prioritar cu 16 intrări. Circuitul codificator prioritar de 8 biți are 8 intrări de date $I[7:0]$ și 3 ieșiri $O[2:0]$. Codificatorul are o intrare de validare E_i (dacă $E_i = 0$, codificatorul este blocat și prezintă toate ieșirile în starea 0). În plus, codificatorul are și o ieșire de validare E_o (activă în 1 dacă pe intrare este o combinație validă, cu cel puțin un 1). Tabelul de funcționare al codificatorului este:

| E_i | I_7 | I_6 | I_5 | I_4 | I_3 | I_2 | I_1 | I_0 | O_2 | O_1 | O_0 | E_o |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0 | x | x | x | x | x | x | x | x | 0 | 0 | 0 | 0 |
| 1 | 1 | x | x | x | x | x | x | x | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | x | x | x | x | x | x | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | x | x | x | x | x | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | x | x | x | x | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | x | x | x | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | x | x | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | x | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Soluție

Circuitul prezintă două instanțe de codificator prioritar de 8 biți. Jumătate din intrări (cele mai semnificative) se leagă la intrările de date ale unui codificator de 8 biți cu intrarea validată $E_i = 1$. Jumătate din intrări (cele mai puțin semnificative) se leagă la intrările de date ale celui de-al doilea codificator de 8 biți cu intrarea de validare (negată) conectată la ieșirea de validare a celui alt codificator.

Dacă se activează ieșirea $E_o = 1$ a celui mai semnificativ codificator, înseamnă că s-a găsit cel puțin o intrare activată, deci numărul de la ieșire trebuie să fie mai mare decât jumătate (cel puțin egal cu 8). Deci, bitul cel mai semnificativ al rezultatului poate fi luat de la ieșirea E_o a celui mai semnificativ codificator.

Dacă $O_3 = 0$, înseamnă că cea mai prioritară intrare activată este între 7 și 0. Cel mai puțin semnificativ codificator va produce la ieșire codul intrării celei mai prioritare activate. Ieșirile codificatorului mai semnificativ vor fi toate egale cu 0.

Dacă $O_3 = 1$, înseamnă că cea mai prioritară intrare activată este între 15 și 8. Cel mai semnificativ codificator va produce la ieșire codul intrării celei mai prioritare activate (minus 8). Ieșirile codificatorului mai puțin semnificativ vor fi toate egale cu 0 deoarece acest codificator va avea $E_i = 0$.

Realizând funcția OR între intrările cu același index ale celor două codificatoare se obțin ieșirile circuitului proiectat.



Circuitul este prezentat în figura 8.8.

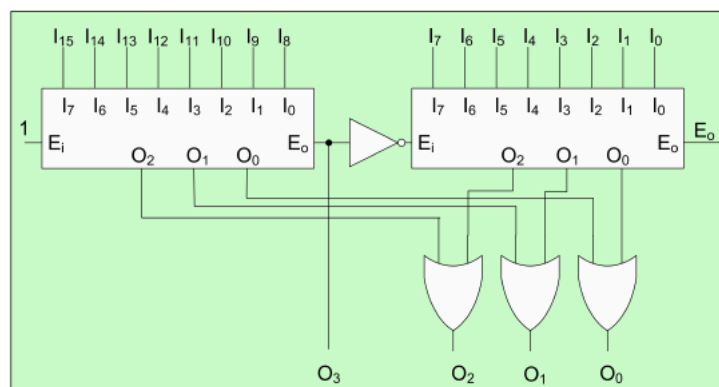


Figura 8.8 Codificator prioritar cu 16 intrări realizat cu două codificatoare prioritare de 8 biți (problema 2).

3. Să se realizeze un circuit logic combinațional care implementează funcția $Y = 2^{2^X}$, unde $X \in [0, 7]$.

Soluție

Circuitul decodificator $N : 2^N$ implementează funcția aritmetică $Y = 2^X$. Deoarece $Y = 2^{2^X}$ poate fi scrisă sub forma $2^{(2^X)}$, implementarea se reduce la conectarea celor 8 ieșiri ale unui DCD 3:8 la intrările unui DCD 8:256. (figura 8.9-a). Se observă că intrările DCD 8:256 provin de la ieșirile DCD 3:8. Deci, nu toate combinațiile de intrare vor fi prezentate DCD 8:256. De fapt, doar 8 combinații din cele $2^8 = 256$ combinații sunt posibile. În fiecare din cele 8 combinații de intrare, o singură ieșire a DCD 8:256 este activată. Cu această observație, se poate înlocui DCD 8:256 cu un set de conexiuni care leagă un bit de intrare cu ieșirea corespunzătoare pe baza formulei $out = 2^{in}$. Celelalte $256 - 8 = 248$ ieșiri sunt întotdeauna egale cu 0. Vor fi activate doar următoarele ieșiri:

$$\begin{aligned} X = 000|_2 = 0|_{10} &\rightarrow Y[2^0] = Y[1] \\ X = 001|_2 = 1|_{10} &\rightarrow Y[2^1] = Y[2] \\ X = 010|_2 = 2|_{10} &\rightarrow Y[2^2] = Y[4] \\ X = 011|_2 = 3|_{10} &\rightarrow Y[2^3] = Y[8] \\ X = 100|_2 = 4|_{10} &\rightarrow Y[2^4] = Y[16] \\ X = 101|_2 = 5|_{10} &\rightarrow Y[2^5] = Y[32] \\ X = 110|_2 = 6|_{10} &\rightarrow Y[2^6] = Y[64] \\ X = 111|_2 = 7|_{10} &\rightarrow Y[2^7] = Y[128]. \end{aligned}$$

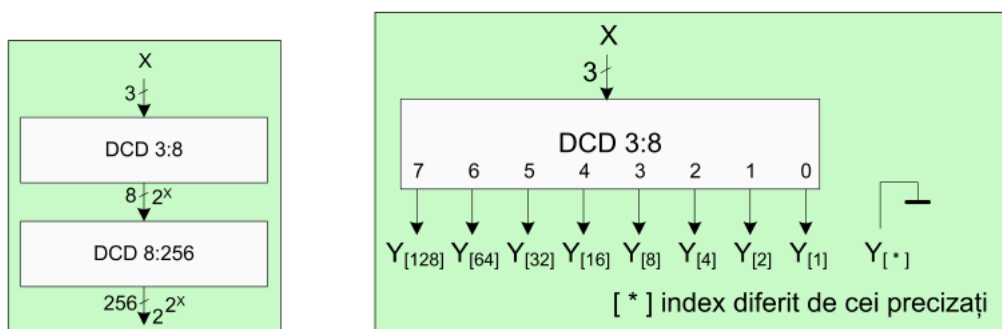


Figura 8.9 Circuit pentru calculul expresiei $y = 2^{2^x}$ (problema 3).

4. Să se realizeze un circuit combinațional care calculează funcția matematică $F_{[N-1:0]} = [\log_2 M_{[2^N-1:0]}]$. S-a notat cu $[]$ partea întregă a numărului.

Soluție

Codificatorul implementează funcția matematică menționată, adică generează indexul celui mai semnificativ 1 din reprezentarea binară a lui $M_{[2^N-1:0]}$.



5. Să se implementeze un circuit combinațional la intrarea căruia se aplică un cuvânt M de 16 biți și un cuvânt N de 2 biți. Ieșirea F a circuitului este activă când M este un multiplu de 2^{2^N} .
6. Minimizați funcțiile logice și implementați-le cu porți NAND. Implementați funcțiile cu DCD 3:8 și un număr minim de porți logice suplimentare.
- a) $F_a = \sum(4, 5, 7, 8, 12) + d(2, 13)$
 b) $F_b = \sum(0, 1, 3, 5, 7, 8, 12, 13, 15) + d(4, 10)$
 c) $F_c = \sum(2, 3, 6, 7, 13, 15) + d(8, 9, 11)$

Soluție

a) Minimizarea funcției, conform diagramei V-K prezentate în figura 8.10, conduce la expresia:

$$F_a = I + II + III = B \cdot \bar{C} + A \cdot \bar{C} \cdot \bar{D} + \bar{A} \cdot B \cdot D$$

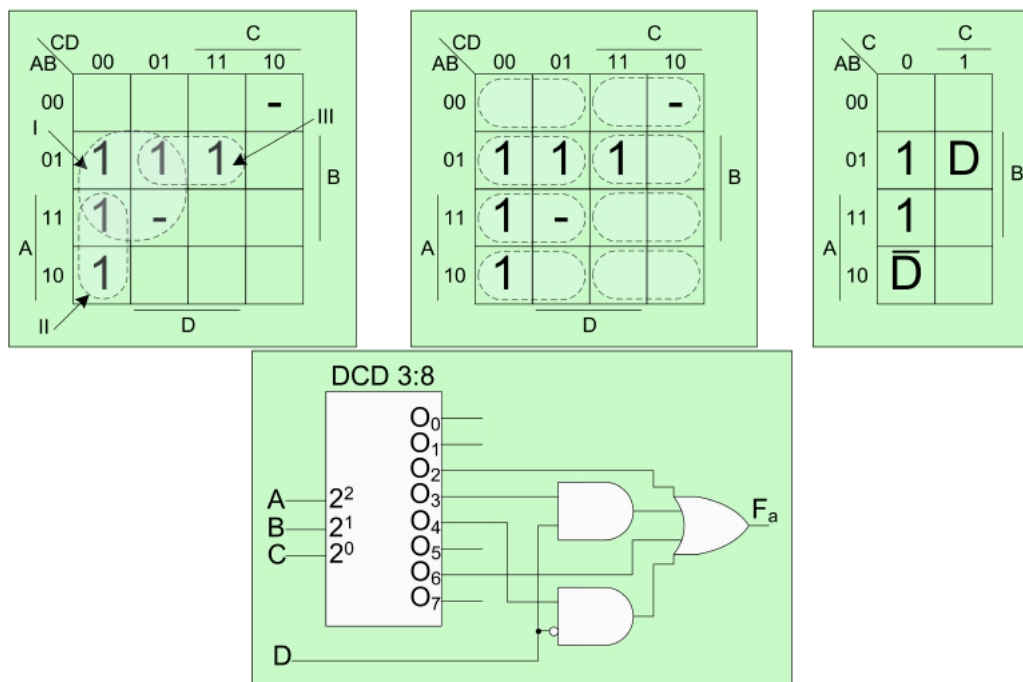


Figura 8.10 F_a , problema 6-a: diagrama V-K, diagrama V-K redusă și implementare cu decodificator și porți logice.

Prin introducerea unei variabile reziduu în diagrama V-K (în acest caz intrarea D), se înjumătățește dimensiunea diagramei și se poate rescrie funcția ca:

$$F_a = m_2 + m_3 \cdot D + m_4 \cdot \bar{D} + m_6, \text{ unde mintermii } m_i(A, B, C).$$

Sub această formă, funcția se poate implementa cu un circuit decodificator 3:8 (care generează toți mintermii intrărilor A, B, C) și o poartă OR. În cazul apariției variabilei reziduu D , mintermul corespunzător intră în poarta OR în conjuncție cu variabila reziduu D sau \bar{D} . Circuitul este prezentat în figura 8.10.