

Dan NICULA

ELECTRONICĂ DIGITALĂ
Carte de învățătură 2.0



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

Lecția 9

Multiplexoare și demultiplexoare

9.1 Noțiuni teoretice

Multiplexorul MUX $2^N : 1$ selectează pe baza unui cod de N biți una din cele 2^N intrări de date pe care o transmite la unica ieșire.

Funcționarea multiplexorului 2:1 (o intrare de selecție și două intrări de date) se poate descrie astfel: "dacă selecția este egală cu 0, ieșirea ia valoarea intrării I_0 , iar dacă selecția este egală cu 1, ieșirea ia valoarea intrării I_1 ".

Multiplexorul implementează în hardware funcția de decizie, similară cu specificația software "if ... then ... else" (în cazul MUX 2:1) sau "case" (în cazul MUX de dimensiuni mai mari).

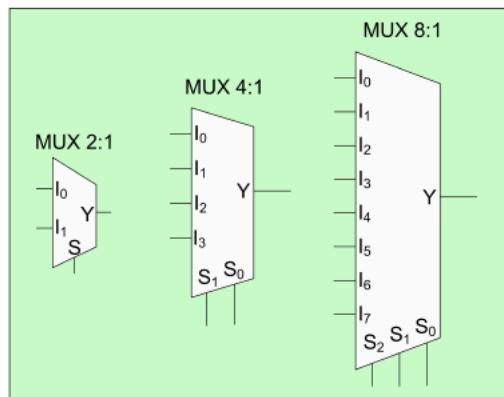


Figura 9.1 Simboluri de multiplexoare.

Ecuațiile funcțiilor logice implementate de MUX sunt:

$$\text{MUX 2:1} \quad Y = S \cdot I_1 + \overline{S} \cdot I_0$$

$$\text{MUX 4:1} \quad Y = S_1 \cdot S_0 \cdot I_3 + S_1 \cdot \overline{S_0} \cdot I_2 + \overline{S_1} \cdot S_0 \cdot I_1 + \overline{S_1} \cdot \overline{S_0} \cdot I_0$$

$$\begin{aligned} \text{MUX 8:1} \quad Y = & S_2 \cdot S_1 \cdot S_0 \cdot I_7 + S_2 \cdot S_1 \cdot \overline{S_0} \cdot I_6 + S_2 \cdot \overline{S_1} \cdot S_0 \cdot I_5 + S_2 \cdot \overline{S_1} \cdot \overline{S_0} \cdot I_4 + \\ & + \overline{S_2} \cdot S_1 \cdot S_0 \cdot I_3 + \overline{S_2} \cdot S_1 \cdot \overline{S_0} \cdot I_2 + \overline{S_2} \cdot \overline{S_1} \cdot S_0 \cdot I_1 + \overline{S_2} \cdot \overline{S_1} \cdot \overline{S_0} \cdot I_0 \end{aligned}$$

Circuitele MUX pot avea o intrare de validare. Funcționarea este neschimbată dacă intrarea de validare este activată. Dacă intrarea de validare este inactivă ieșirea multiplexorului este egală cu 0 indiferent de valorile intrărilor de date (multiplexorul nu transmite la ieșire nicio valoare primită la intrare). În acest caz, ecuațiile se modifică astfel:



$$\text{MUX 2:1} \quad Y = E \cdot S \cdot I_1 + E \cdot \bar{S} \cdot I_0$$

$$\text{MUX 4:1} \quad Y = E \cdot S_1 \cdot S_0 \cdot I_3 + E \cdot S_1 \cdot \bar{S}_0 \cdot I_2 + E \cdot \bar{S}_1 \cdot S_0 \cdot I_1 + E \cdot \bar{S}_1 \cdot \bar{S}_0 \cdot I_0$$

$$\begin{aligned} \text{MUX 8:1} \quad Y = & E \cdot S_2 \cdot S_1 \cdot S_0 \cdot I_7 + E \cdot S_2 \cdot S_1 \cdot \bar{S}_0 \cdot I_6 + E \cdot S_2 \cdot \bar{S}_1 \cdot S_0 \cdot I_5 + E \cdot S_2 \cdot \bar{S}_1 \cdot \bar{S}_0 \cdot I_4 + \\ & + E \cdot \bar{S}_2 \cdot S_1 \cdot S_0 \cdot I_3 + E \cdot \bar{S}_2 \cdot S_1 \cdot \bar{S}_0 \cdot I_2 + E \cdot \bar{S}_2 \cdot \bar{S}_1 \cdot S_0 \cdot I_1 + E \cdot \bar{S}_2 \cdot \bar{S}_1 \cdot \bar{S}_0 \cdot I_0 \end{aligned}$$

Structurile logice ale multiplexoarelor se obțin pe baza ecuațiilor.

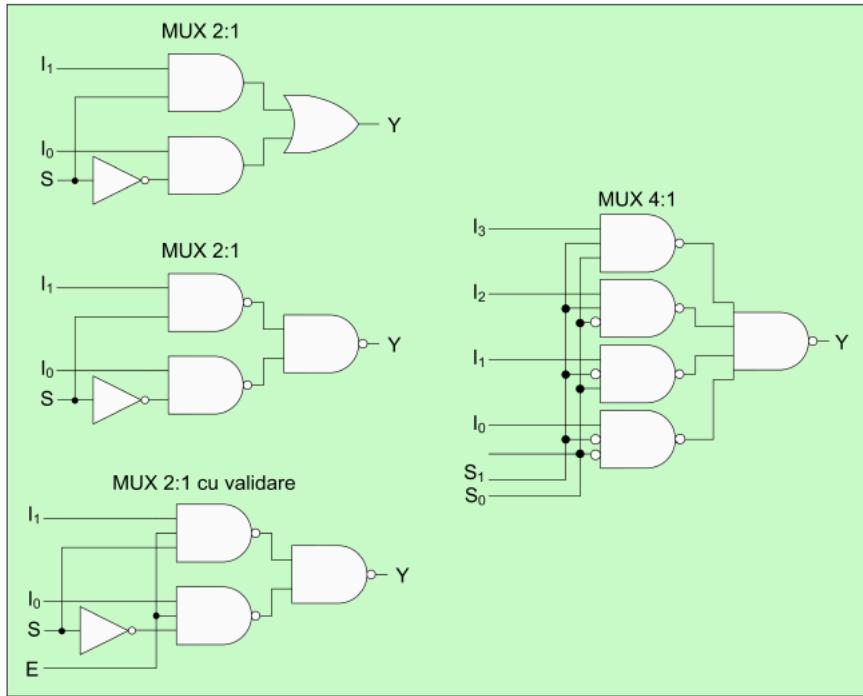


Figura 9.2 Structuri de multiplexoare.

Orice funcție logică de N variabile poate fi implementată utilizând un MUX $2^N : 1$, fără minimizare, astfel:

- Pe cele N intrări de selecție ale multiplexorului se conectează cele N intrări.
- Pe cele 2^N intrări de date ale multiplexorului se conectează valorile logice 1 și 0, conform coloanei funcției din tabelul de adevăr. Dacă funcția este exprimată FCND, se poate utiliza regula: intrările de date ale multiplexorului ce își regăsesc indexul în FCND se conectează la 1, cele ce nu apar în FCND se conectează la 0.

Costul maxim total al implementării oricărei funcții de N intrări este un MUX $2^{N-1} : 1$.

Orice funcție logică de N variabile poate fi implementată utilizând un MUX $2^{N-1} : 1$, fără minimizare, astfel:

- Pe cele $N - 1$ intrări de selecție ale multiplexorului se conectează cele mai semnificative $N - 1$ intrări.
- Pe cele 2^{N-1} intrări de date ale multiplexorului se conectează funcții reziduu ale celei mai puțin semnificative intrări. Funcțiile reziduu de o variabilă pot fi implementate cu maximum un inversor deoarece acestea sunt: 0, 1, X , \bar{X} . Funcțiile reziduu se pot determina din tabelul de adevăr astfel:
 - Se delimitizează în tabelul de adevăr grupuri de câte două rânduri.
 - Se compară valorile de pe cele două rânduri, coloana variabilei reziduu (cea mai puțin semnificativă, aflată în partea dreaptă) și coloana funcției.
 - Pot exista 4 cazuri:

Intrare reziduu X	Funcție F	Funcție reziduu R[i]	Observații
0	0		
1	0	0	$F = 0$ pe toate rândurile grupului
0	0		
1	1	X	$F = X$ pe toate rândurile grupului
0	1		
1	0	\bar{X}	$F = \bar{X}$ pe toate rândurile grupului
0	1		
1	1	1	$F = 1$ pe toate rândurile grupului

Costul maxim total al implementării oricărei funcții de N intrări este un MUX $2^{N-1} : 1$ și un inversor.

Demultiplexorul DMUX $1 : 2^N$ selectează pe baza unui cod de N biți una din cele 2^N ieșiri de date pe care transmite intrarea de date.

Funcționarea demultiplexorului 1:2 (o intrare de selecție și două ieșiri de date) se poate descrie astfel: "dacă selecția este egală cu 0, intrarea este transmisă la ieșirea Y_0 , iar dacă selecția este egală cu 1, intrarea este transmisă la ieșirea Y_1 ".

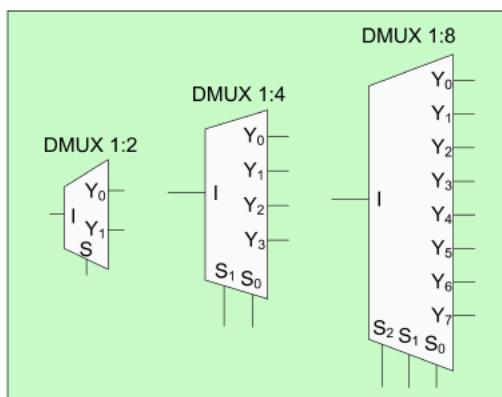


Figura 9.3 Simboluri de demultiplexoare.

Ecuațiile funcțiilor logice implementate de DMUX sunt:

$$\text{DMUX 1:2} \quad Y_1 = S \cdot I \quad Y_0 = \bar{S} \cdot I$$

$$\text{DMUX 1:4} \quad Y_3 = S_1 \cdot S_0 \cdot I \quad Y_2 = S_1 \cdot \bar{S}_0 \cdot I \quad Y_1 = \bar{S}_1 \cdot S_0 \cdot I \quad Y_0 = \bar{S}_1 \cdot \bar{S}_0 \cdot I$$

$$\text{DMUX 1:8} \quad Y_7 = S_2 \cdot S_1 \cdot S_0 \cdot I \quad Y_6 = S_2 \cdot S_1 \cdot \bar{S}_0 \cdot I \quad Y_5 = S_2 \cdot \bar{S}_1 \cdot S_0 \cdot I \quad Y_4 = S_2 \cdot \bar{S}_1 \cdot \bar{S}_0 \cdot I \\ Y_3 = \bar{S}_2 \cdot S_1 \cdot S_0 \cdot I \quad Y_2 = \bar{S}_2 \cdot S_1 \cdot \bar{S}_0 \cdot I \quad Y_1 = \bar{S}_2 \cdot \bar{S}_1 \cdot S_0 \cdot I \quad Y_0 = \bar{S}_2 \cdot \bar{S}_1 \cdot \bar{S}_0 \cdot I$$

Circuitele DMUX $1 : 2^N$ au un comportament similar cu cel al unui decodificator de N biți cu intrare de validare. Dacă la un DMUX $1 : 2^N$ se conectează intrarea $I = 1$, circuitul este echivalent cu un DCD $N : 2^N$.

Structurile logice ale demultiplexoarelor se obțin pe baza ecuațiilor.

Orice funcție logică de N variabile poate fi implementată utilizând un DMUX $1 : 2^N$, fără minimizare, astfel:

- Pe cele N intrări de selecție ale demultiplexorului se conectează cele N intrări.
- Pe intrarea de date se va conecta 1.
- Conform modului de conectare, demultiplexorul generează pe cele 2^N ieșiri funcțiile logice ale tuturor celor 2^N minterminilor ai variabilelor de intrare. Implementarea necesită conectarea la intrările unei porți OR a ieșirilor demultiplexorului cu indexul egal cu indexul rândurilor cu valoare 1 în coloana de adevăr a funcției. Dacă funcția este exprimată FCND, se poate utiliza regula: ieșirile demultiplexorului ce își regăsesc indexul în FCND se conectează la intrarea unei porți OR.



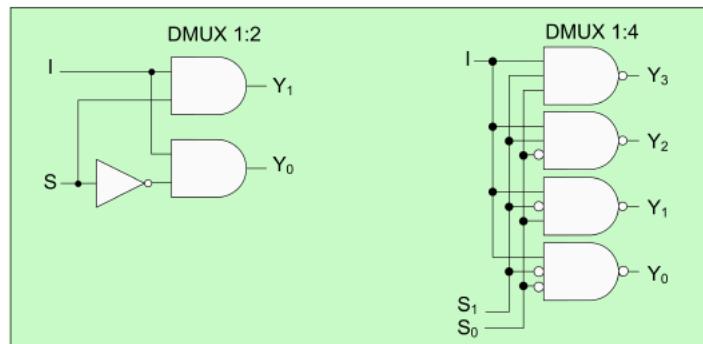


Figura 9.4 Structuri de demultiplexoare.

- Dacă în expresia analitică a funcției exprimată FCND sunt prezenți mai mult de jumătate de mintermi (2^{N-1}), se pot uni într-o poartă NOR ieșirile ale căror index nu apar în FCND.

Costul maxim total al implementării oricărei funcții de N intrări este un DMUX $1 : 2^N$ și o poartă OR/NOR de maximum 2^{N-1} intrări.

O funcție logică de N variabile poate fi implementată pe un DMUX $1 : 2^{N-1}$, fără minimizare, dacă în expresia analitică o variabilă de intrare poate fi dată în factor (negată sau ne-negată).

De exemplu, $F(A, B, C, D) = A \cdot G(B, C, D)$ sau $F(A, B, C, D) = \bar{A} \cdot G(B, C, D)$.

Implementarea constă în conectarea intrării factorizate pe intrarea de date a demultiplexorului. Celelalte intrări se conectează pe intrările de selecție ale demultiplexorului $1 : 2^{N-1}$. Cu o poartă OR se vor conecta ieșirile necesare pentru realizarea funcției cu variabilele de pe selecție.

Costul maxim total al implementării oricărei funcții de N intrări ce îndeplinește condiția menționată este un DMUX $1 : 2^{N-1}$ și o poartă OR/NOR de maximum 2^{N-2} intrări (eventual un inversor pe intrarea de date).

9.2 Pentru cei ce vor doar să promoveze examenul

- Implementați următoarele funcții conform cerințelor:
 - $F_a(A, B) = \sum(0, 3)$, cu MUX 4:1
 - $F_b(A, B, C) = \sum(0, 2, 5, 7)$, cu MUX 8:1
 - $F_c(A, B, C, D) = \sum(3, 4, 5, 9, 10, 12, 14, 15)$, cu MUX 16:1
 - $F_d(A, B) = \sum(0, 3)$, cu MUX 2:1 și un inversor
 - $F_e(A, B, C) = \sum(0, 2, 5, 7)$, cu MUX 4:1 și un inversor
 - $F_f(A, B, C, D) = \sum(3, 4, 5, 9, 10, 12, 14, 15)$, cu MUX 8:1 și un inversor
 - $F_g(A, B, C) = \sum(3, 4, 6, 7)$, cu MUX 8:1
 - $F_h(A, B, C) = \sum(3, 4, 6, 7)$, cu MUX 4:1 și un inversor

Soluție

g) Funcția $F_g(A, B, C) = \sum(3, 4, 6, 7)$ are 3 intrări. Implementarea cu MUX 8:1 necesită conectarea celor 3 intrări pe intrările de selecție și a valorilor din coloana de adevăr a funcției pe intrările de date ale multiplexorului. Altfel spus, intrările de date ale multiplexorului care au indexul prezent în forma FCND a funcției se conectează la 1. Fără minimizare, rezultă implementarea din figura 9.5.

h) Funcția $F_h(A, B, C) = \sum(3, 4, 6, 7)$ are 3 intrări. Implementarea cu MUX 4:1 necesită conectarea a două intrări (cele mai semnificative, A și B) pe cele 2 intrări pe intrări de selecție ale multiplexorului și a funcțiilor reziduu ale celei mai puțin semnificative intrări (C) pe intrările de date. Funcțiile reziduu se deduc din tabelul următor:

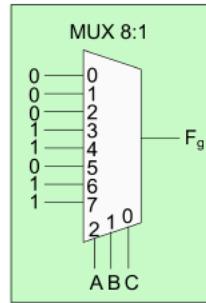


Figura 9.5 Implementarea cu MUX 8:1 a funcției de 3 intrări $F_g(A, B, C) = \sum(3, 4, 6, 7)$, problema 1-g.

A	B	C	F_h	Funcții reziduu
0	0	0	0	$I_0 = 0$
0	0	1	0	
0	1	0	0	$I_1 = C$
0	1	1	1	
1	0	0	1	$I_2 = \overline{C}$
1	0	1	0	
1	1	0	1	$I_3 = 1$
1	1	1	1	

Fără minimizare, rezultă implementarea din figura 9.6.

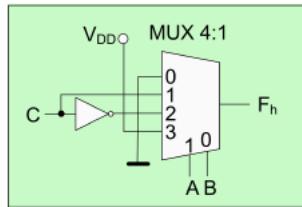


Figura 9.6 Implementarea cu MUX 4:1 a funcției de 3 intrări $F_h(A, B, C) = \sum(3, 4, 6, 7)$, problema 1-h.

2. Determinați funcția logică implementată de multiplexoare în cazul realizării următoarelor conexiuni:
- MUX 4:1 cu conexiunile: $S_1 = A$, $S_0 = B$, $I_0 = I_3 = 0$, $I_1 = I_2 = 1$
 - MUX 4:1 cu conexiunile: $S_1 = A$, $S_0 = B$, $I_0 = I_3 = 0$, $I_1 = C$, $I_2 = 1$
 - MUX 8:1 cu conexiunile: $S_2 = A$, $S_1 = B$, $S_0 = C$, $I_0 = I_3 = 0$ și $I_1 = I_2 = I_4 = I_5 = I_6 = 1$
 - MUX 8:1 cu conexiunile: $S_2 = A$, $S_1 = B$, $S_0 = C$, $I_0 = I_3 = 0$, $I_7 = D$, $I_1 = I_2 = \overline{D}$ și $I_4 = I_5 = I_6 = 1$
 - MUX 8:1 cu conexiunile: $S_2 = A$, $S_1 = B$, $S_0 = C$, $I_0 = I_3 = 0$, $I_1 = I_2 = D + E$, $I_4 = A \cdot B$, $I_5 = I_6 = E$ și $I_7 = 1$

Soluție

- a) Se particularizează funcția multiplexorului 4:1 cu valorile impuse.
 $Y = S_1 \cdot S_0 \cdot I_3 + S_1 \cdot \overline{S_0} \cdot I_2 + \overline{S_1} \cdot S_0 \cdot I_1 + \overline{S_1} \cdot \overline{S_0} \cdot I_0$

Rezultă:

$$F_a(A, B) = A \cdot B \cdot 0 + A \cdot \overline{B} \cdot 1 + \overline{A} \cdot B \cdot 1 + \overline{A} \cdot \overline{B} \cdot 0 = A \cdot \overline{B} + \overline{A} \cdot B = \sum(2, 1) = \sum(1, 2) = A \oplus B$$

- b) Se particularizează funcția multiplexorului 4:1 cu valorile impuse.

$$Y = S_1 \cdot S_0 \cdot I_3 + S_1 \cdot \overline{S_0} \cdot I_2 + \overline{S_1} \cdot S_0 \cdot I_1 + \overline{S_1} \cdot \overline{S_0} \cdot I_0$$

Rezultă:

$$F_a(A, B, C) = A \cdot B \cdot 0 + A \cdot \overline{B} \cdot 1 + \overline{A} \cdot B \cdot C + \overline{A} \cdot \overline{B} \cdot 0 = A \cdot \overline{B} + \overline{A} \cdot B \cdot C = A \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot B \cdot C = \sum(5, 4, 3) = \sum(3, 4, 5)$$

3. Să se implementeze funcția $F(A, B, C, D) = \sum(3, 4, 6, 7, 10, 11, 12, 14, 15)$ astfel:
- cu multiplexor 16 : 1;

- b) cu multiplexor 8 : 1, cu variabila reziduu D, pe selecție $S_2S_1S_0 = ABC$;
 c) cu multiplexor 8 : 1, cu variabila reziduu C, pe selecție $S_2S_1S_0 = ABD$;
 d) cu multiplexor 4 : 1, cu variabile reziduu C și D, pe selecție $S_1S_0 = AB$;
 e) cu multiplexor 4 : 1, cu variabile reziduu B și C, pe selecție $S_1S_0 = AD$.

Soluție

a) Funcția are 4 variabile de intrare. Implementarea cu MUX 16:1 presupune:

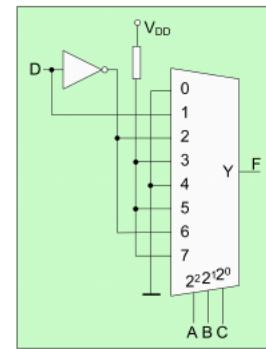
- conectarea pe intrările de selecție a celor 4 variabile de intrare;

- conectarea pe intrările de date a valorilor logice corespunzătoare definirii funcției, conform tabelului de adevăr:

b) Implementarea cu MUX 8:1 presupune introducerea funcțiilor dependente de o variabilă reziduu pe intrările de date. În tabelul de adevăr se grupează rândurile două câte două, prezentând aceleași valori pentru variabilele introduse pe intrările de selecție (A, B, C).

AB\CD	00	01	11	10	C
00	0	0	1	1	
01	1	1	1	1	
11	1	1	1	1	
10	0	1	1	1	D

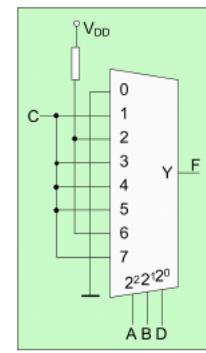
Nr.	A	B	C	D	F	I_x	ABC
0	0	0	0	0	0	$I_0 = 0$	000
1	0	0	0	1	0		
2	0	0	1	0	0	$I_1 = D$	001
3	0	0	1	1	1		
4	0	1	0	0	1	$I_2 = \overline{D}$	010
5	0	1	0	1	0		
6	0	1	1	0	1	$I_3 = 1$	011
7	0	1	1	1	1		
8	1	0	0	0	0	$I_4 = 0$	100
9	1	0	0	1	0		
10	1	0	1	0	1	$I_5 = 1$	101
11	1	0	1	1	1		
12	1	1	0	0	1	$I_6 = \overline{D}$	110
13	1	1	0	1	0		
14	1	1	1	0	1	$I_7 = 1$	111
15	1	1	1	1	1		



c) Tabelul de adevăr se prelucrează similar cu cazul b) însă în tabel s-au inversat coloanele C și D pentru a deveni adiacente rândurile având aceleași valori pentru variabilele introduse pe intrările de selecție (A, B, D).

AB\CD	00	01	11	10	C
00	0	0	1	1	
01	1	1	1	1	
11	1	1	1	1	
10	0	1	1	1	D

Nr.	A	B	D	C	F	I_x	ABD
0	0	0	0	0	0	$I_0 = 0$	000
2	0	0	0	1	0		
1	0	0	1	0	0	$I_1 = C$	001
3	0	0	1	1	1		
4	0	1	0	0	1	$I_2 = 1$	010
6	0	1	0	1	1		
5	0	1	1	0	0	$I_3 = C$	011
7	0	1	1	1	1		
8	1	0	0	0	0	$I_4 = C$	100
10	1	0	0	1	1		
9	1	0	1	0	0	$I_5 = C$	101
11	1	0	1	1	1		
12	1	1	0	0	1	$I_6 = 1$	110
14	1	1	0	1	1		
13	1	1	1	0	0	$I_7 = C$	111
15	1	1	1	1	1		

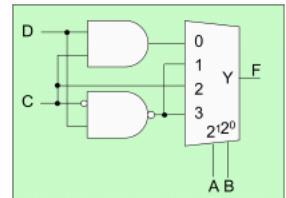


d) Se grupează câte 4 rânduri și se deduc cele 4 funcții reziduu dependente de C și D .



A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

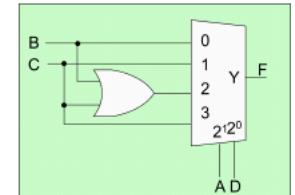
A	B	C	D	F	I_x	$S_1 S_0 = AB$
0	0	0	0	0		
0	0	0	1	0	$I_0 = C \cdot D$	00
0	0	1	0	0		
0	0	1	1	1		
0	1	0	0	1		
0	1	0	1	0	$I_1 = \overline{C} \cdot D$	01
0	1	1	0	1		
0	1	1	1	1		
1	0	0	0	0		
1	0	0	1	0	$I_2 = C$	10
1	0	1	0	1		
1	0	1	1	1		
1	1	0	0	1		
1	1	0	1	0	$I_3 = \overline{C} \cdot D$	11
1	1	1	0	1		
1	1	1	1	1		



e) Similar cu punctul d), cu coloanele inversate astfel încât să fie adiacente coloanele apartinând variabilelor introduse pe intrările de selecție (A, D).

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

A	D	B	C	F	I_x	$S_1 S_0 = AD$
0	0	0	0	0		
0	0	0	1	0	$I_0 = B$	00
0	0	1	0	1		
0	0	1	1	1		
0	1	0	0	0		
0	1	0	1	1	$I_1 = C$	01
0	1	1	0	0		
0	1	1	1	1		
1	0	0	0	0		
1	0	0	1	1	$I_2 = B + C$	10
1	0	1	0	1		
1	0	1	1	1		
1	1	0	0	0		
1	1	0	1	1	$I_3 = C$	11
1	1	1	0	0		
1	1	1	1	1		



4. Găsiți portile logice echivalente ale circuitelor cu multiplexor 2:1 prezentate în figura 9.7.

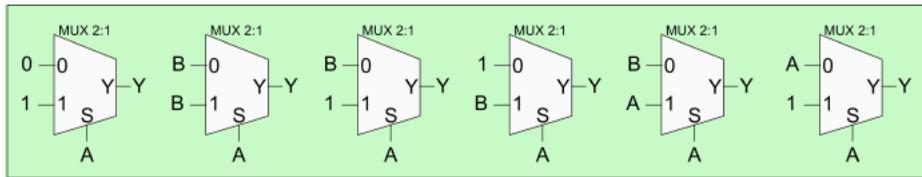


Figura 9.7 Circuite particulare implementate cu multiplexor 2:1, problema 4.

Soluție

Ecuatia implementată de MUX 2:1 este $Y = \overline{S} \cdot I_0 + S \cdot I_1$. Particularizând ecuația pentru circuitele prezentate rezultă:

- a) $Y = \overline{A} \cdot 0 + A \cdot 1 = A$
- b) $Y = \overline{A} \cdot B + A \cdot B = B$
- c) $Y = \overline{A} \cdot B + A \cdot 1 = A + B$
- d) $Y = \overline{A} \cdot 1 + A \cdot B = \overline{A} + B$



5. Implementați funcțiile cu MUX 8:1.

- a) $F_a = \sum(1, 3, 5) + d(0, 2)$
- b) $F_b = \sum(0, 3, 4, 5, 6, 7, 14, 15) + d(10, 11)$
- c) $F_c = \sum(0, 1, 5, 7, 9) + d(10, 11, 12, 13, 14, 15)$

9.3 Pentru cei ce vor să învețe

1. Care este numărul de intrări ale unui multiplexor 4:1? Aceeași întrebare pentru multiplexoarele 8:1, 16:1, 32:1 și 64:1. Generalizați răspunsul pentru multiplexor $2^N : 1$.

Soluție

Un multiplexor $2^N : 1$ are 2^N intrări de date selectate pe baza unui cod de selecție de N biți. Distincția "intrări de date" și "intrări de selecție" este făcută ținând cont de funcționalitatea circuitului. Însă, cele două tipuri de intrări sunt similare, din punct de vedere al intrărilor unui circuit combinațional. Deci, un multiplexor $2^N : 1$ are $2^N + N$ intrări și o ieșire.

2. Care din următoarele funcții este implementată pe circuitul din figura 9.8?

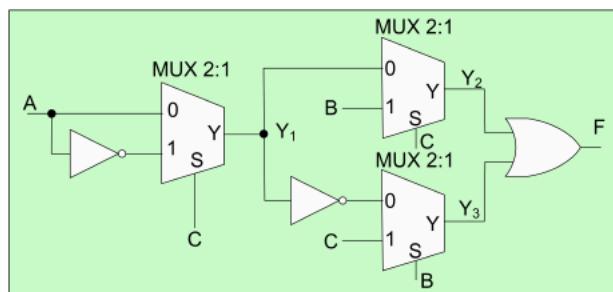


Figura 9.8 Circuitul cu multiplexoare referit la problema 2.

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

Soluție

$$\begin{aligned}
 Y_1(A, B, C) &= \overline{C} \cdot A + C \cdot \overline{A}, \\
 Y_2(A, B, C) &= \overline{C} \cdot Y_1 + C \cdot B = \overline{C} \cdot (\overline{C} \cdot A + C \cdot \overline{A}) + C \cdot B = A \cdot \overline{C} + B \cdot C = A \cdot B \cdot \overline{C} + A \cdot \overline{B} \cdot \overline{C} + A \cdot B \cdot C + \overline{A} \cdot B \cdot C = \\
 &\sum(6, 4, 7, 3) = \sum(3, 4, 6, 7), \\
 Y_3(A, B, C) &= \overline{B} \cdot \overline{Y_1} + B \cdot C = \overline{B} \cdot (A \cdot C + \overline{A} \cdot \overline{C}) + B \cdot C = A \cdot \overline{B} \cdot C + \overline{A} \cdot \overline{B} \cdot \overline{C} + A \cdot B \cdot C + \overline{A} \cdot B \cdot C = \\
 &\sum(5, 0, 7, 3) = \sum(0, 3, 5, 7), \\
 F(A, B, C) &= Y_3 + Y_2 = \sum(3, 4, 6, 7) + \sum(0, 3, 5, 7) = \sum(0, 3, 4, 5, 6, 7)
 \end{aligned}$$

$$F(A, B, C) = \sum(0, 3, 4, 5, 6, 7), \text{ deci răspunsul corect este d).}$$

3. Care din următoarele funcții este implementată de circuitul din figura 9.9?

- a) $F(A, B) = \overline{A \cdot B}$
- b) $F(A, B) = \overline{A + B}$
- c) $F(A, B) = \overline{A \oplus B}$
- d) $F(A, B) = A \oplus B$
- e) $F(A, B) = A \cdot B$

Soluție

$$F(A, B) = m_0 + m_3 = \overline{A \oplus B}, \text{ deci răspunsul corect este c).}$$

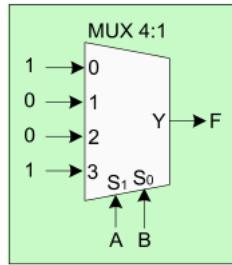


Figura 9.9 Circuitul cu multiplexor referit la problema 3.

4. Proiectați un circuit de ”vot majoritar” cu 3 intrări și implementați-l cu multiplexor 4:1, fără minimizarea funcției. Ieșirea circuitului are valoarea logică a majorității intrărilor.

Soluție

Tabelul de adevăr al circuitului ”vot majoritar” este:

V_2	V_1	V_0	VOT
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Implementarea cu MUX4:1 necesită scrierea funcției cu o variabilă reziduu (V_0):

$$\begin{aligned} VOT &= V_2 \cdot V_1 + V_2 \cdot V_0 + V_1 \cdot V_0 = V_2 \cdot V_1 \cdot V_0 + V_2 \cdot V_1 \cdot \overline{V_0} + V_2 \cdot V_1 \cdot V_0 + V_2 \cdot \overline{V_1} \cdot V_0 + V_2 \cdot V_1 \cdot V_0 + \overline{V_2} \cdot V_1 \cdot V_0 = \\ &= V_2 \cdot V_1 \cdot (V_0 + \overline{V_0}) + V_0 + V_1 \cdot (V_0 + \overline{V_0}) + \overline{V_2} \cdot V_1 \cdot (V_0 + \overline{V_0}) + \overline{V_2} \cdot \overline{V_1} \cdot (0) = m_3 \cdot 1 + m_2 \cdot V_0 + m_1 \cdot V_0 + m_0 \cdot 0, \end{aligned}$$

unde $m_i(V_2, V_1)$.

Circuitul este prezentat în figura 9.10.

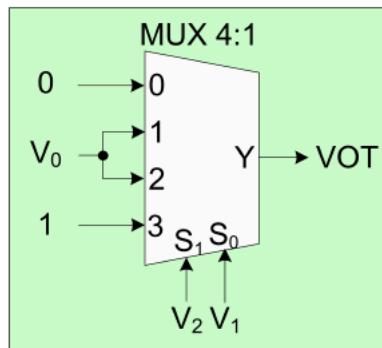


Figura 9.10 Funcția ”vot majoritar”: implementare cu multiplexor.

5. Să se implementeze următoarele circuite MUX cu circuitele MUX de dimensiuni mai reduse, precizate:
- MUX 4:1 cu MUX 2:1
 - MUX 8:1 cu MUX 2:1
 - MUX 8:1 cu MUX 4:1
 - MUX 16:1 cu MUX 2:1
6. Să se implementeze un MUX 32:1 utilizând 4 circuite MUX 8:1 cu validare și un decodificator de 2 biți.



Soluție

MUX 32:1 are 5 biți de selecție, 32 biți de date și o ieșire. Cei mai semnificativi 2 biți de selecție se decodifică pentru a se obține 4 semnale de validare. Cele 4 semnale de validare se aplică câte unul la fiecare MUX 8:1 astfel încât, la un moment dat, o singură ieșire este nenuță. Cei 32 biți de date se aplică pe cele 4×8 intrări de date ale multiplexoarelor. Cele 4 ieșiri ale multiplexoarelor se conectează la intrările unei porți OR cu 4 intrări care generează bitul de ieșire.

7. Se consideră circuitul de adunare a operanzilor de 8 biți proveniți din 4 surse posibile pentru un operand și alte 4 surse posibile pentru al doilea operand și direcționarea rezultatului spre una din cele 4 destinații posibile, cu schema bloc prezentată în figura 9.11.

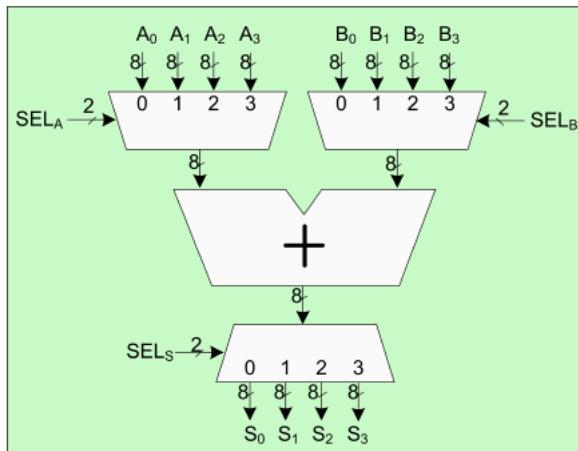


Figura 9.11 Circuitul de adunare descris la problema 7.

Să se scrie cuvintele de comandă pentru următoarele adunări:

- a) $S_1 = A_1 + B_1$
- b) $S_3 = A_2 + B_3$
- c) $S_2 = A_2 + B_1$
- d) $S_3 = A_2 + B_0$
- e) $S_0 = A_3 + B_3$

Soluție

Cuvintele de comandă sunt:

- a) $SEL_A = 01, SEL_B = 01, SEL_S = 01$
- b) $SEL_A = 10, SEL_B = 11, SEL_S = 11$

8. Să se implementeze funcțiile în variantele precizate:

$$F_1(A, B, C, D) = \sum (0, 1, 3, 4, 7, 8, 10, 11, 15)$$

$$F_2(A, B, C, D) = \sum (3, 4, 5, 6, 13, 14, 15)$$

$$F_3(A, B, C, D) = \prod (0, 1, 4, 5, 6, 9, 12, 14)$$

- a) cu MUX 2:1
- b) cu MUX 4:1
- c) cu MUX 16:1
- d) cu MUX 2:1 și MUX 4:1

Soluție

a) Implementarea funcțiilor cu MUX 2:1 se poate face, fără minimizare, pe o structură arborescentă de multiplexoare pe un număr de nivele egal cu numărul de variabile de intrare. Multiplexoarele de pe fiecare nivel au aceeași selecție. Variabilele de intrare se aplică pe intrările de selecție, în ordine, intrarea cea mai semnificativă pe ultimul nivel de multiplexare. Pe intrările de date se aplică valorile logice corespunzătoare minternilor prezenți în expresia funcției. În funcție de valorile unei anumite funcții, se poate renunța la primul nivel de multiplexoare care poate fi implementat cu maxim un inversor.

Figurile 9.12-a,b prezintă implementarea funcției F_1 pe structura arborescentă de multiplexoare 2:1 și structura redusă, ținând cont de valorile concrete ale funcției.

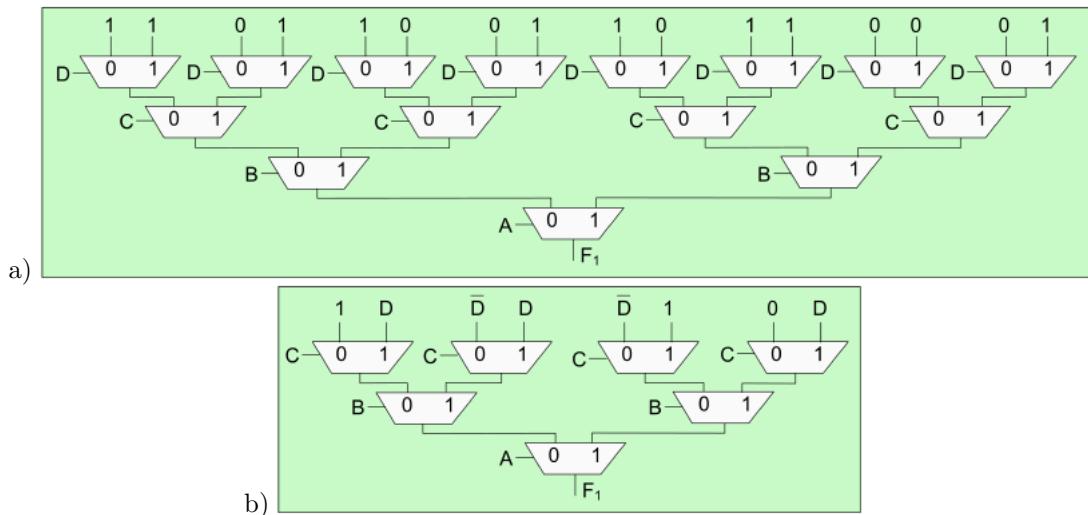


Figura 9.12 a) Implementarea funcției F_1 pe o rețea arborescentă de MUX 2:1, **b)** varianta redusă, problema 8.

b) Implementarea funcțiilor cu MUX 4:1 este similară cu implementarea cu MUX 2:1 cu deosebirea că numărul de nivele de multiplexare se înjumătățește deoarece pe fiecare nivel se utilizează câte două variabile de selecție.

9. Să se implementeze funcția $F(A, B, C, D) = \sum(3, 6, 11, 14, 15) + \sum d(2, 5, 12, 13)$, cu MUX 4:1.
10. Să se implementeze funcția $F(A, B, C, D, E) = \bar{A} \cdot B \cdot E + \bar{A} \cdot \bar{B} \cdot D \cdot E + A \cdot \bar{B} \cdot C \cdot E + A \cdot C \cdot \bar{D} \cdot E$ cu MUX 8:1 cu validare.

Soluție

Funcția având 5 variabile, pentru implementare cu MUX 8:1, este necesară exprimarea acesteia cu două variabile reziduu. Deoarece în expresia funcției se poate factoriza variabila E, aceasta poate fi utilizată pentru comanda validării circuitului. Rămâne de introdus numai o singură variabilă reziduu. Expresia funcției se transformă în:

$$\begin{aligned}
 F(A, B, C, D, E) &= E \cdot (\bar{A} \cdot B + \bar{A} \cdot \bar{B} \cdot D + A \cdot \bar{B} \cdot C + A \cdot C \cdot \bar{D}) = E \cdot [\bar{A} \cdot (B + \bar{B} \cdot D) + A \cdot (\bar{B} \cdot C + C \cdot \bar{D})] = \\
 &= E \cdot [\bar{A} \cdot (B \cdot \bar{C} \cdot \bar{D} + B \cdot \bar{C} \cdot D + B \cdot C \cdot \bar{D} + B \cdot C \cdot D) + A \cdot (\bar{B} \cdot C \cdot \bar{D} + \bar{B} \cdot C \cdot D + \bar{B} \cdot C \cdot \bar{D} + B \cdot C \cdot \bar{D})] = \\
 &= E \cdot [\bar{A} \cdot (m_4 + m_5 + m_6 + m_7 + m_1 + m_3) + A \cdot (m_2 + m_3 + m_2 + m_6)] = \\
 &= E \cdot [\bar{A} \cdot (m_1 + m_3 + m_4 + m_5 + m_6 + m_7) + A \cdot (m_2 + m_3 + m_6)] = \\
 &= E \cdot [\bar{A} \cdot (m_1 + m_4 + m_5 + m_7) + A \cdot (m_2) + (m_3 + m_6)]
 \end{aligned}$$

Mintermii sunt calculați asupra celor trei variabile m_i (B, C, D).

Funcția de ieșire a circuitului MUX 8:1 cu validare este:

$$Y = E \cdot (I_0 \cdot m_0 + I_1 \cdot m_1 + I_2 \cdot m_2 + I_3 \cdot m_3 + I_4 \cdot m_4 + I_5 \cdot m_5 + I_6 \cdot m_6 + I_7 \cdot m_7)$$

Funcția considerată se poate scrie ca:

$$F = E \cdot (0 \cdot m_0 + \bar{A} \cdot m_1 + A \cdot m_2 + 1 \cdot m_3 + \bar{A} \cdot m_4 + \bar{A} \cdot m_5 + 1 \cdot m_6 + \bar{A} \cdot m_7)$$

Implementarea acestei expresii este reprezentată în figura 9.13.

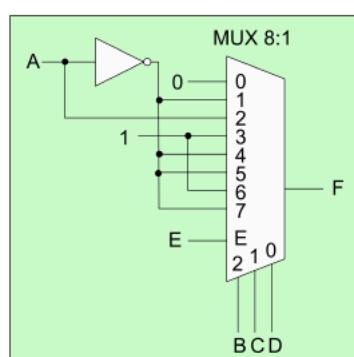


Figura 9.13 Implementarea funcției de la problema 10 cu MUX 8:1.



11. Să se implementeze funcția:

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

utilizând un circuit DMUX 1:8, un MUX 4:1 și portii logice OR/NOR.

Soluție

Ideea implementării constă în utilizarea a 2 variabile de intrare (D, E) conectate pe intrările de selecție ale multiplexorului 4:1 și celelalte 3 variabile de intrare (A, B, C) pe intrările de selecție ale demultiplexorului. Demultiplexorul generează toți mintermii variabilelor A, B, C. Multiplexorul implementează funcția:

$$Y = H_3 \cdot D \cdot E + H_2 \cdot D \cdot \overline{E} + H_1 \cdot \overline{D} \cdot E + H_0 \cdot \overline{D} \cdot \overline{E}$$

Funcțiile $H_{0,1,2,3}(A, B, C)$ pot fi obținute cu un singur DMUX 1:8 și câte o poartă OR/NOR pentru fiecare funcție, în total maximum 4 portii OR/NOR. Expresia funcției se prelucrează analitic pentru a fi adusă la forma canonica disjunctivă FCD (prin adăugarea variabilelor lipsă):

$$\begin{aligned} F(A, B, C, D, E) &= \overline{A} \cdot B \cdot E + \overline{A} \cdot \overline{B} \cdot D \cdot E + A \cdot \overline{B} \cdot C \cdot E + A \cdot C \cdot \overline{D} \cdot E + A \cdot B \cdot C \cdot D \cdot \overline{E} + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{E} = \\ &= \overline{A} \cdot B \cdot (\overline{C} + C) \cdot (\overline{D} + D) \cdot E + \overline{A} \cdot \overline{B} \cdot (\overline{C} + C) \cdot D \cdot E + A \cdot \overline{B} \cdot C \cdot (\overline{D} + D) \cdot E + A \cdot (\overline{B} + B) \cdot C \cdot \overline{D} \cdot E + A \cdot B \cdot C \cdot D \cdot \overline{E} + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{E} = \\ &= \overline{A} \cdot B \cdot \overline{C} \cdot \overline{D} \cdot E + \overline{A} \cdot B \cdot \overline{C} \cdot D \cdot E + \overline{A} \cdot B \cdot C \cdot \overline{D} \cdot E + \overline{A} \cdot B \cdot C \cdot D \cdot E + \\ &\quad + \overline{A} \cdot \overline{B} \cdot C \cdot D \cdot E + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D \cdot E + \\ &\quad + A \cdot \overline{B} \cdot C \cdot \overline{D} \cdot E + A \cdot B \cdot C \cdot \overline{D} \cdot E + \\ &\quad + A \cdot B \cdot C \cdot D \cdot \overline{E} + \\ &\quad + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{E} \end{aligned}$$

Apoi se dau în factor mintermii variabilelor D și E: $D \cdot E$, $D \cdot \overline{E}$, $\overline{D} \cdot E$, $\overline{D} \cdot \overline{E}$, rezultând:

$$\begin{aligned} F &= (\overline{A} \cdot B \cdot \overline{C} + \overline{A} \cdot B \cdot C + \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot \overline{B} \cdot \overline{C} + A \cdot \overline{B} \cdot C) \cdot D \cdot E + \\ &\quad + A \cdot B \cdot C \cdot D \cdot \overline{E} + \\ &\quad + (\overline{A} \cdot B \cdot \overline{C} + \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot \overline{C} + A \cdot B \cdot C) \cdot \overline{D} \cdot E + \\ &\quad + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{E} \\ &= (m_2 + m_3 + m_1 + m_0 + m_5) \cdot D \cdot E + \\ &\quad + m_7 \cdot D \cdot \overline{E} + \\ &\quad + (m_2 + m_3 + m_5 + m_0 + m_7) \cdot \overline{D} \cdot E + \\ &\quad + m_4 \cdot \overline{D} \cdot \overline{E} \end{aligned}$$

$$F = (m_0 + m_1 + m_2 + m_3 + m_5) \cdot D \cdot E + m_7 \cdot D \cdot \overline{E} + (m_2 + m_3 + m_5 + m_7) \cdot \overline{D} \cdot E + m_4 \cdot \overline{D} \cdot \overline{E}$$

Expresia finală obținută este implementată de circuitul din figura 9.14.

$$\begin{aligned} F &= (\overline{m_4 + m_6 + m_7}) \cdot D \cdot E + m_7 \cdot D \cdot \overline{E} + (m_2 + m_3 + m_5 + m_7) \cdot \overline{D} \cdot E + m_4 \cdot \overline{D} \cdot \overline{E} = \\ &= H_3 \cdot D \cdot E + H_2 \cdot D \cdot \overline{E} + H_1 \cdot \overline{D} \cdot E + H_0 \cdot \overline{D} \cdot \overline{E} \end{aligned}$$

În cazul H_3 s-a optat pentru complementarea funcției pentru a reduce numărul de intrări de la 5 la 3.

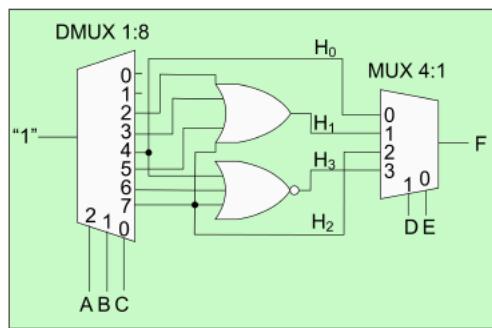


Figura 9.14 Implementarea funcției de la problema 11 cu DMUX 1:8 și MUX 4:1.



12. Să se implementeze funcția:

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

utilizând un circuit DMUX 1:8, un MUX 2:1 și porti logice OR/NOR.

Soluție

Se observă că în expresia funcției se poate da în factor variabila E , scriindu-se funcția ca:

$$F(A, B, C, D, E) = E \cdot G(A, B, C, D)$$

Apoi, funcția $G(A, B, C, D)$ se scrie sub forma:

$$G(A, B, C, D) = H_0(A, B, C) \cdot \overline{D} + H_1(A, B, C) \cdot D$$

Funcția G se va implementa cu MUX 2:1 având pe selecție variabila D , iar funcțiile $H_{0,1}(A, B, C)$ se implementează cu DMUX 1:8 și porti OR/NOR. Circuitul DMUX va avea pe intrare variabila E , realizând $F = 0$ dacă $E = 0$.

Prin prelucrări algebrice se obține:

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

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

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

Apoi se dau în factor \overline{D} și D , rezultând:

$$G(A, B, C, D) = (\overline{A} \cdot B \cdot \overline{C} + \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot C) \cdot \overline{D} + (\overline{A} \cdot B \cdot \overline{C} + \overline{A} \cdot B \cdot C + \overline{A} \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot C) \cdot D = \\ = (m_2 + m_3 + m_5 + m_7) \cdot \overline{D} + (m_2 + m_3 + m_1 + m_0 + m_5) \cdot D$$

Rezultă $H_0(A, B, C) = \sum(2, 3, 5, 7)$ și $H_1(A, B, C) = \sum(0, 1, 2, 3, 5)$

Expresia finală obținută este implementată de circuitul din figura 9.15.

$$F(A, B, C, D, E) = E \cdot [(m_2 + m_3 + m_5 + m_7) \cdot \overline{D} + (m_0 + m_1 + m_2 + m_3 + m_5) \cdot D]$$

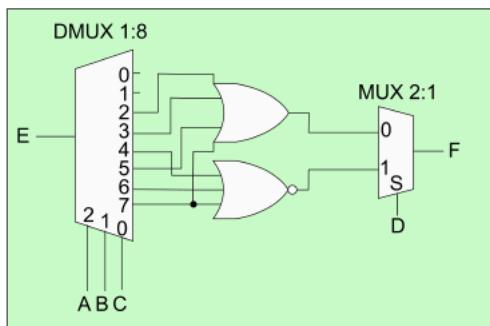


Figura 9.15 Implementarea funcției de la problema 12 cu DMUX 1:8 și MUX 2:1.

13. Să se implementeze funcția $F(A, B, C, D, E) = \sum(3, 6, 7, 10, 11, 19, 22, 26, 27, 30, 31)$ utilizând un circuit DMUX 1:8, porti logice OR și MUX 4:1.

Să se implementeze aceeași funcție cu DMUX 1:4, porti logice OR și MUX 4:1 cu validare.

14. Să se implementeze funcția $F(A, B, C, D, E) = \overline{A} \cdot B \cdot E + \overline{A} \cdot \overline{B} \cdot D \cdot E + A \cdot \overline{B} \cdot C \cdot E + A \cdot C \cdot \overline{D} \cdot E$ utilizând un circuit DMUX 1:4 urmat de un MUX 4:1 și porti logice OR, conectând intrările C și D pe intrările de selecție și E pe intrarea de date a demultiplexorului iar intrările A și B pe intrările de selecție ale demultiplexorului.

Soluție

Circuitul DMUX 1:4, cu intrarea de date activată (E), generează toți mintermii de 2 variabile (A și B).

Circuitul MUX 4:1 selectează o intrare pe baza unui cod de selecție de 2 biți (C și D).

Pentru această implementare funcția trebuie pusă sub forma:

$$F(A, B, C, D, E) = E \cdot (C \cdot D \cdot G_3(A, B) + C \cdot \overline{D} \cdot G_2(A, B) + \overline{C} \cdot D \cdot G_1(A, B) + \overline{C} \cdot \overline{D} \cdot G_0(A, B))$$

Funcțiile $G_{3,2,1,0}(A, B)$ vor fi implementate cu porti OR pe baza mintermilor generați de demultiplexor.



$$\begin{aligned}
 F(A, B, C, D, E) &= E \cdot (\overline{A} \cdot B + \overline{A} \cdot \overline{B} \cdot D + A \cdot \overline{B} \cdot C + A \cdot C \cdot \overline{D}) = E \cdot [m_1 + m_0 \cdot D + m_2 \cdot C + (m_2 + m_3) \cdot C \cdot \overline{D}], \text{ unde } m_i(A, B). \\
 F(A, B, C, D, E) &= E \cdot [m_1 \cdot (C \cdot D + C \cdot \overline{D} + \overline{C} \cdot D + \overline{C} \cdot \overline{D}) + m_0 \cdot (C \cdot D + \overline{C} \cdot D) + m_2 \cdot (C \cdot \overline{D} + C \cdot D) + (m_2 + m_3) \cdot C \cdot \overline{D}] = \\
 &= E \cdot [(m_0 + m_1 + m_2) \cdot C \cdot D + (m_1 + m_2 + m_3) \cdot C \cdot \overline{D} + (m_0 + m_1) \cdot \overline{C} \cdot D + m_1 \cdot \overline{C} \cdot \overline{D}]
 \end{aligned}$$

Implementarea este prezentată în figura 9.16.

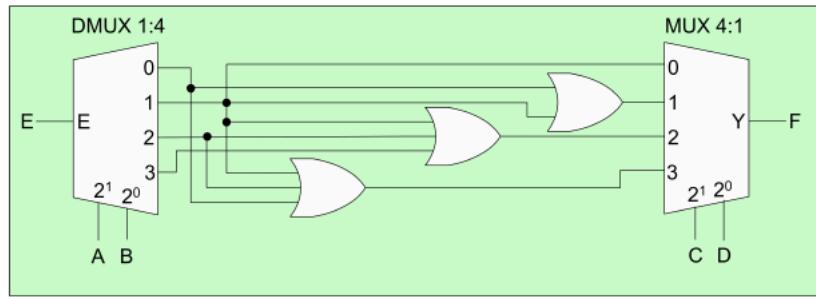


Figura 9.16 Implementarea funcției de la problema 14 cu DMUX și MUX.

15. Să se implementeze numai cu MUX 4:1 următoarele funcții:

- $F(A, B, C, D, E) = A \cdot B \cdot \overline{C} \cdot D + \overline{B} \cdot C \cdot D \cdot E + B \cdot C \cdot \overline{D} \cdot \overline{E} + \overline{A} \cdot B \cdot C \cdot \overline{D}$
- $F(A, B, C, D, E) = \sum(1, 2, 3, 4, 5, 6, 7, 10, 14, 20, 22, 28, 31)$

Soluție

Funcțiile au 5 variabile de intrare. Implementarea cu MUX 4:1 se poate face pe două nivele: pe primul nivel 4 circuite MUX 4:1, având pe intrările de selecție două variabile de intrare și pe intrările de date funcții reziduu ale unei variabile de intrare. Pe al doilea nivel, apare încă un MUX 4:1 având pe intrările de selecție două variabile de intrare (cele mai semnificative) și intrările de date conectate la ieșirile multiplexoarelor de pe primul nivel. Pentru implementare, funcția trebuie pusă sub formă canonica disjunctivă. Funcțiile reziduu se pot determina din observarea indecșilor mintermilor care apar în expresia funcției. Pentru o intrare de date, se grupează câte doi mintermi adiacenți. Mintermii 0 și 1 vor genera funcția reziduu pentru $ABCD = 0000$, mintermii 2 și 3 vor genera funcția reziduu pentru $ABCD = 0001$, ... mintermii 30 și 31 vor genera funcția reziduu pentru $ABCD = 1111$.

Dacă în expresia funcției nu apare niciunul din cei doi mintermi, se va considera funcția reziduu egală cu 0. Dacă în expresia funcției apar ambii dintre cei doi mintermi, se va considera funcția reziduu egală cu 1. Dacă în expresia funcției apare mintermul par dar nu și cel impar, se va considera funcția reziduu negată. Dacă în expresia funcției apare mintermul impar dar nu și cel par, se va considera funcția reziduu directă.

Pentru funcția b), datele pot deduse din tabel:

Minterm	A	B	C	D	E	$F(A, B, C, D, E)$	Reziduu(E)
0	0	0	0	0	0	0	
1	0	0	0	0	1	1	E
2	0	0	0	1	0	1	
3	0	0	0	1	1	1	
4	0	0	1	0	0	1	
5	0	0	1	0	1	1	1
6	0	0	1	1	0	1	
7	0	0	1	1	1	1	1
8	0	1	0	0	0	0	
9	0	1	0	0	1	0	0
10	0	1	0	1	0	1	
11	0	1	0	1	1	0	\bar{E}
12	0	1	1	0	0	0	
13	0	1	1	0	1	0	0
14	0	1	1	1	0	1	
15	0	1	1	1	1	0	\bar{E}
16	1	0	0	0	0	0	
17	1	0	0	0	1	0	0
18	1	0	0	1	0	0	
19	1	0	0	1	1	0	0
20	1	0	1	0	0	1	
21	1	0	1	0	1	0	\bar{E}
22	1	0	1	1	0	1	
23	1	0	1	1	1	0	\bar{E}
24	1	1	0	0	0	0	
25	1	1	0	0	1	0	0
26	1	1	0	1	0	0	
27	1	1	0	1	1	0	0
28	1	1	1	0	0	1	
29	1	1	1	0	1	0	\bar{E}
30	1	1	1	1	0	0	
31	1	1	1	1	1	1	E

Pentru funcția b) circuitul este prezentat în figura 9.17.

16. Analizați circuitul prezentat în figura 9.18.

Soluție

Circuitul conține un DMUX 1:4 cu selecția provenind de la intrarea de 2 biți A , cu ieșirile conectate "unu-la-unu" la intrările unui MUX 4:1 cu selecția provenind de la intrarea de 2 biți B . Circuitul va propaga la ieșirea Y valoarea 1 aplicată pe intrarea de date/validare a demultiplexorului doar când cele două intrări sunt egale $A[1 : 0] = B[1 : 0]$. Circuitul implementează un comparator de două numere reprezentate pe câte 2 biți.

17. Determinați funcțiile logice implementate de circuitele din figura 9.19.

Soluție

a) Circuitul implementează o funcție de 4 intrări cu multiplexor 8:1, intrarea A fiind conectată ca funcții reziduu pe intrările de date ale multiplexorului, iar intrările B, C, D realizând selecția datelor. Rezultă funcția:

$$F(A, B, C, D) = \sum m_i \cdot d_i \text{ unde } m_i(B, C, D).$$

$$F(A, B, C, D) = m_0 \cdot 1 + m_1 \cdot 1 + m_2 \cdot \bar{A} + m_3 \cdot A + m_4 \cdot A + m_5 \cdot 1 + m_6 \cdot 0 + m_7 \cdot \bar{A} = m_0 + m_1 + m_2 \cdot \bar{A} + m_3 \cdot A + m_4 \cdot A + m_5 + m_7 \cdot \bar{A}$$

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

18. Implementați următoarele funcții cu multiplexor 8:1 și un inversor:

- a) $F(A, B, C, D) = \sum(0, 3, 5, 7, 8, 14, 15)$
- b) $F(A, B, C, D) = \sum(0, 2, 5, 7, 8, 9, 11, 12, 13)$
- c) $F(A, B, C, D) = \sum(3, 4, 5, 9, 10, 12, 14, 15)$
- d) $F(A, B, C, D) = \sum(2, 4, 6, 7, 9, 10)$



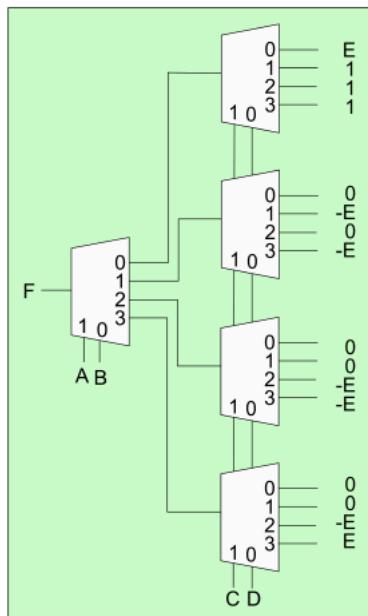


Figura 9.17 Funcție de 5 intrări implementată cu multiplexoare 4:1, problema 15-b).

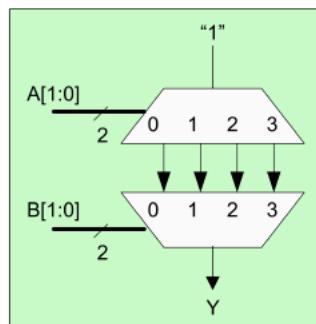


Figura 9.18 Circuitul de analizat la problema 16.

$$\text{e)} \quad F(A, B, C, D) = \sum(1, 3, 5, 7, 11, 13, 15)$$

$$\text{f)} \quad F(A, B, C, D) = \sum(0, 1, 3, 4, 8, 9, 15)$$

$$\text{g)} \quad F(A, B, C, D) = \sum(2, 3, 5, 6, 8, 9, 12, 14)$$

Soluție

Funcțiile au 4 intrări. Implementarea cu MUX 8:1 necesită conectarea a 3 intrări pe intrările de selecție și a unei variabile de intrare pe intrările de date, sub forma de funcții reziduu. Dacă se conectează pe intrările de selecție variabilele mai semnificative, funcțiile reziduu se determină prin gruparea a doi indecsăi adiacenți din expresia funcției în forma canonica disjunctivă, similar cu modul descris la problema 15.

$$\text{a)} \quad F(A, B, C, D) = \sum(0, 3, 5, 7, 8, 14, 15) = m_0 \cdot \overline{D} + m_1 \cdot D + m_2 \cdot \overline{D} + m_3 \cdot D + m_4 \cdot \overline{D} + m_5 \cdot 0 + m_6 \cdot 0 + m_7 \cdot 1$$

unde $m_i(A, B, C)$

Circuitul este prezentat în figura 9.20.

19. Determinați funcția logică implementată de un multiplexor 8:1 în cazul realizării următoarelor conexiuni:

$$\text{a)} \quad S_2 = A, \quad S_1 = B, \quad S_0 = C, \quad I_1 = I_2 = I_7 = 0, \quad I_3 = I_5 = 1, \quad I_0 = I_4 = D \text{ și } I_6 = \overline{D}$$

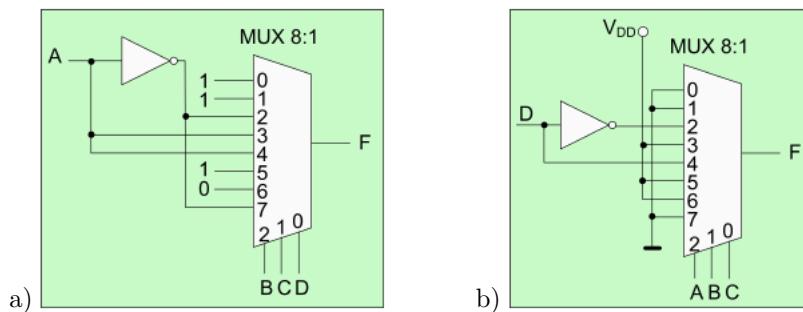
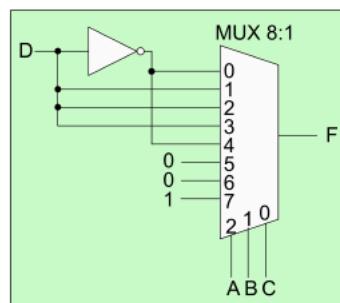
$$\text{b)} \quad S_2 = C, \quad S_1 = B, \quad S_0 = A, \quad I_1 = I_2 = I_6 = 0, \quad I_3 = I_4 = E, \quad I_0 = D, \quad I_5 = D \cdot E \text{ și } I_7 = 1$$

$$\text{c)} \quad S_2 = C, \quad S_1 = A, \quad S_0 = B \cdot C, \quad I_0 = I_3 = 0, \quad I_1 = I_2 = D + E, \quad I_4 = A \cdot B, \quad I_5 = I_6 = E \text{ și } I_7 = 1$$

Soluție

$$\text{a)} \quad F(A, B, C, D) = S_2 \cdot S_1 \cdot S_0 \cdot I_7 + S_2 \cdot S_1 \cdot \overline{S}_0 \cdot I_6 + S_2 \cdot \overline{S}_1 \cdot S_0 \cdot I_5 + S_2 \cdot \overline{S}_1 \cdot \overline{S}_0 \cdot I_4 + \overline{S}_2 \cdot S_1 \cdot S_0 \cdot I_3 + \overline{S}_2 \cdot S_1 \cdot \overline{S}_0 \cdot I_2 + \overline{S}_2 \cdot \overline{S}_1 \cdot I_1 + \overline{S}_2 \cdot \overline{S}_1 \cdot \overline{S}_0 \cdot I_0 = A \cdot B \cdot C \cdot 0 + A \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot C \cdot 1 + A \cdot \overline{B} \cdot \overline{C} \cdot D + \overline{A} \cdot B \cdot C \cdot 1 + \overline{A} \cdot B \cdot \overline{C} \cdot 0 + \overline{A} \cdot \overline{B} \cdot C \cdot 0 + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D = A \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot \overline{C} \cdot D + \overline{A} \cdot B \cdot C + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D = \sum(12, 11, 10, 9, 7, 6, 1)$$

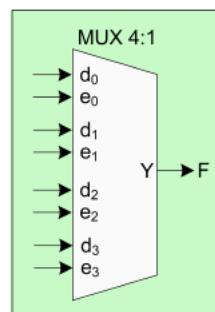


**Figura 9.19** Funcții logice implementate cu multiplexor 8:1, problema 17.**Figura 9.20** Funcție de 4 intrări implementată cu multiplexor 8:1, problema 18-a).

20. Figura 9.21 prezintă un multiplexor 4:1 ale cărui intrări de selecție sunt decodificate (se presupune că o singură intrare e_3, e_2, e_1, e_0 este activată la un moment dat). Tabelul de adevăr care caracterizează acest multiplexor este prezentat în continuare:

e_3	e_2	e_1	e_0	Y
1	0	0	0	d_3
0	1	0	0	d_2
0	0	1	0	d_1
0	0	0	1	d_0

- a) Să se scrie funcția logică a ieșirii.
- b) Să se implementeze circuitul cu porti logice.
- c) Care este valoarea ieșirii în cazul în care nicio intrare de selecție nu este activată ($e_3 = e_2 = e_1 = e_0 = 0$)?
- d) Care este valoarea ieșirii în cazul în care două sau mai multe intrări de selecție sunt activate simultan ($e_3 = e_2 = 1, e_1 = e_0 = 0$ sau $e_3 = e_2 = e_1 = e_0 = 1$)?

**Figura 9.21** Multiplexor 4:1 cu selecția decodificată, problema 20.

Soluție

- a) Se observă că tabelul de adevăr este incomplet definit: din cele $2^4 = 16$ combinații ale intrărilor e_3, e_2, e_1, e_0 sunt prezente doar 4. Presupunerea este că nu se activează două intrări simultan. Cu această presupunere,

expresia logică a funcției se poate scrie:

$$Y(e_3, e_2, e_1, e_0, d_3, d_2, d_1, d_0) = e_3 \cdot d_3 + e_2 \cdot d_2 + e_1 \cdot d_1 + e_0 \cdot d_0$$

b) Circuitul este prezentat în figura 9.22.

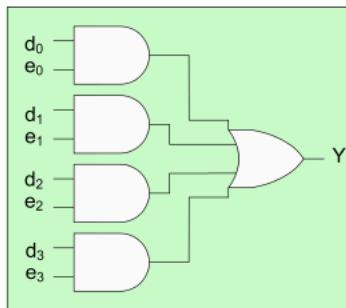


Figura 9.22 Structură internă a multiplexorului 4:1 cu selecția decodificată, problema 20.

- c) Conform ecuației logice a ieșirii, în cazul în care nicio intrare de validare nu este activată ieșirea este $Y = 0$.
d) Conform ecuației logice a ieșirii, în cazul în care mai mult de o intrare de validare este activată, ieșirea realizează funcția OR a intrărilor de date corespunzătoare validate.

$$Y(1, 1, 0, 0, d_3, d_2, d_1, d_0) = d_3 + d_2$$

$$Y(1,1,1,1,d_3,d_2,d_1,d_0) = d_3 + d_2 + d_1 + d_0$$

21. Analizați circuitul prezentat în figura 9.23 și deduceti ecuațiile funcțiilor de transfer ale acestuia.

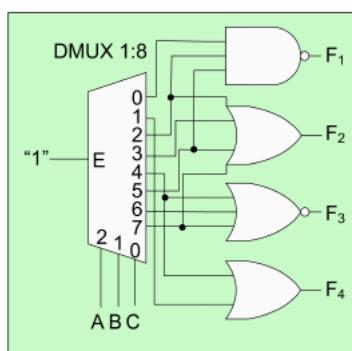


Figura 9.23 Circuit de analizat, problema 21.

Solutie

Circuitul DMUX generează la ieșire toți mintermii variabilelor A, B, C .

$F_1(A, B, C) = \overline{m_0} \cdot \overline{m_2} \cdot \overline{m_5} = \overline{0} = 1$ (deoarece este o coconjuncție de mintermi de indecsă diferenți)

$$F_2(A, B, C) = m_2 + m_3 + m_5 + m_7 = \sum(2, 3, 5, 7)$$

$$F_3(A, B, C) = \frac{m_2 + m_3 + m_5 + m_7}{m_4 + m_6 + m_7} = \sum(0, 1, 2, 3, 5)$$

$$F_4(A, B, C) = m_1 + m_4 = \sum (1, 4)$$

22. Implementați cu multiplexor 2:1 următoarele funcții logice, conectând intrarea A pe intrarea de selecție a multiplexorului: $A \cdot B$, $A + B$, $A \oplus B$, $\overline{A} + B$.

23. Implementați cu multiplexor 4:1 următoarele funcții logice, conectând intrările A și B pe intrările de selecție ale multiplexorului: $A + \overline{B} \cdot C$, $A \cdot B + \overline{A} \cdot C$, $F(A, B, C) = \sum(0, 2, 4, 5)$, $\overline{A} + \overline{B} \cdot \overline{C}$.
Implementați aceleași funcții cu multiplexor 2:1 și porți adiționale, conectând intrarea A pe selecția multiplexorului.

24. Proiectați arborii de multiplexare organizati după regulile:

Proiectați arborii de multiplexare organizați după regulile:

- a) $A|BC|F$ b) $AB|C|F$ c) $A|BC|D|F$ d) $AB|CD|F$

Soluție

- a) Un arbore de multiplexare utilizează mai multe nivele de multiplexare de dimensiuni mici pentru a implementa un multiplexor de dimensiuni mari. Regula $A|BC|F$ trebuie înțeleasă ca o funcție $F(A, B, C)$.

Pe primul nivel (cel de intrare) se conectează pe intrările de selecție intrările mai puțin semnificative ale funcției, BC . Pe primul nivel, există două circuite MUX 4:1 care generează semnale ce se conectează la intrările de date ale multiplexorului MUX 2:1 de pe al doilea nivel.

Pe al doilea nivel (cel ce generează ieșirea) se conectează pe selecția multiplexorului intrarea cea mai semnificativă, A . Intrările de date provin de la ieșirile multiplexoarelor de pe nivelele inferioare. Pe ultimul nivel există întotdeauna un singur multiplexor. Dimensiunea acestuia (în acest caz) este MUX 2:1 (deoarece are un singur bit de selecție A).

Circuitul rezultat este prezentat în figura 9.24. Circuitul este echivalent cu un MUX 8:1 (având 3 intrări de selecție, ABC).

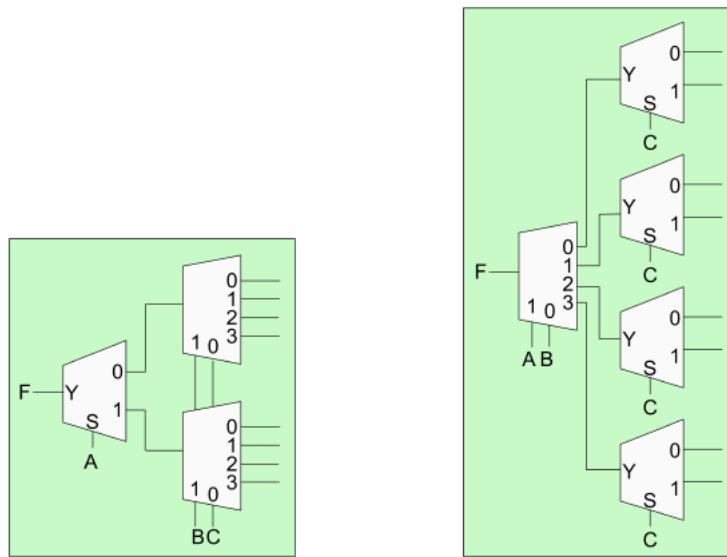


Figura 9.24 Arbore de multiplexare $A|BC|F$ și $AB|C|F$, problema 24-a,b.

b) Circuitul are un MUX 4:1 pe al doilea nivel (cu selecții AB) și 4 MUX 2:1 pe primul nivel (cu selecție C). Circuitul este echivalent cu un MUX 8:1 (având 3 intrări de selecție ABC).

c) Circuitul are un MUX 2:1 pe al treilea nivel (cu selecție A), 2 x MUX 4:1 pe al doilea nivel (cu selecție BC) și 8 x MUX 2:1 pe primul nivel (cu selecția D).

d) Circuitul are un MUX 4:1 pe al doilea nivel (cu selecție AB) și 4 x MUX 4:1 pe primul nivel (cu selecție CD).

25. Să se implementeze funcția $F(A, B, C) = \sum(2, 3, 7)$ cu un DMUX 1:4 și o poartă logică OR.

26. Implementați funcțiile cu MUX 8:1.

- a) $F_a = \sum(1, 3, 5, 9, 14, 15) + d(4, 6, 12, 13)$
- b) $F_b = \sum(0, 3, 4, 5, 6, 7, 9, 12, 14, 15) + d(10)$
- c) $F_c = \sum(0, 1, 5, 7, 9, 10, 14) + d(3, 8)$
- d) $F_d = \sum(2, 6, 7, 9, 10) + d(0, 8)$
- e) $F_e = \sum(5, 6, 7, 8, 9, 10, 13, 14, 15) + d(0, 11)$
- f) $F_f = \sum(1, 2, 3, 5, 7, 10, 13, 14, 15) + d(6, 8)$

Soluție

a) Implementarea funcției cu multiplexor nu necesită minimizare. În schimb, pentru că funcția are 4 intrări iar multiplexorul necesar MUX 8:1 doar 3 intrări de selecție, este necesară exprimarea funcției cu o variabilă reziduu. Pentru aceasta se poate utiliza diagrama V-K, prezentată în figura 9.25. Funcția se descrie:

$$F_a = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D + \overline{A} \cdot \overline{B} \cdot C \cdot D + \overline{A} \cdot B \cdot \overline{C} + A \cdot \overline{B} \cdot \overline{C} \cdot D + A \cdot B \cdot C = m_0 \cdot D + m_1 \cdot D + m_2 + m_4 \cdot D + m_7,$$

unde mintermii $m_i(A, B, C)$.

Se observă că valorile indiferente s-au considerat 0, 1 astfel încât pe intrările de date să fie cât mai multe valori constante (ca să nu se mărească inutil fan-out-ul portilor ce preced circuitul).

Implementarea funcției F_a cu MUX 8:1 este prezentată în figura 9.25.

d) Implementarea funcției cu multiplexor nu necesită minimizare. În schimb, pentru că funcția are 4 intrări iar multiplexorul necesar MUX 8:1 doar 3 intrări de selecție, este necesară exprimarea funcției cu o variabilă

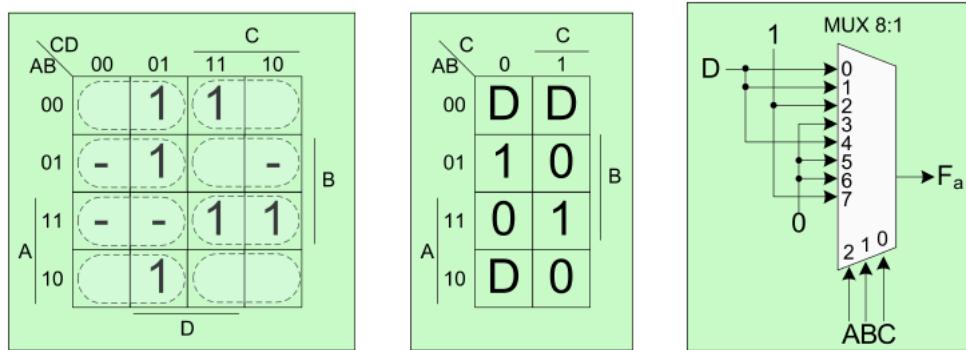


Figura 9.25 F_a , problema 26-a: diagrama V-K și implementare cu MUX 8:1.

reziduu. Pentru aceasta se poate utiliza diagrama V-K, prezentată în figura 7.8. Funcția se rescrie: $F_d = \overline{A} \cdot \overline{B} \cdot C \cdot \overline{D} + \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot \overline{C} + A \cdot \overline{B} \cdot \overline{D} = m_1 \cdot \overline{D} + m_3 + m_2 + m_5 \cdot \overline{D}$, unde mintermii $m_i(A, B, C)$.

Implementarea funcției F_d cu MUX 8:1 este prezentată în figura 9.26.

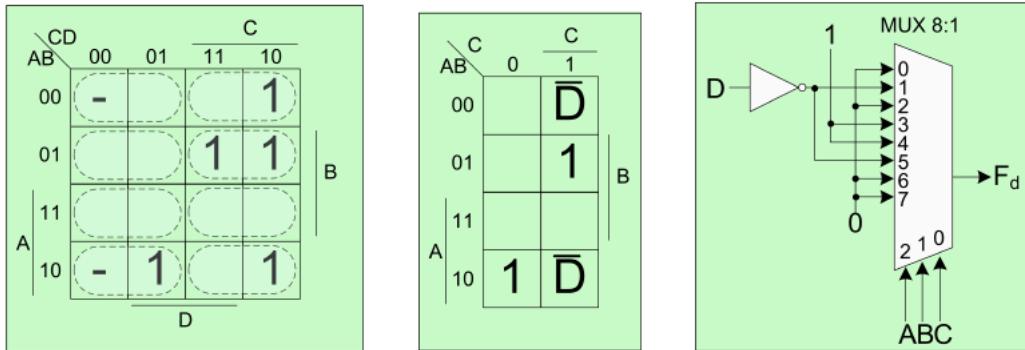


Figura 9.26 F_d , problema 26-d: diagrama V-K și implementare cu MUX 8:1.

27. Minimizați funcția logică și implementați-o cu porți NAND. Evaluați ”costul” implementării. Propuneți o soluție de eliminare a hazardului combinațional și evaluați costul suplimentar al acesteia. Implementați funcția cu MUX 8:1 și un număr minim de porți logice suplimentare.
- $F_a(A, B, C, D) = \sum(5, 6, 7, 8, 13, 14, 15) + d(2, 4)$
 - $F_b(A, B, C, D) = \sum(2, 3, 4, 6, 8, 11) + d(5, 7, 15)$

Soluție

Orice funcție de 4 intrări se poate implementa cu un MUX 8:1 dacă pe intrările de selecție se conectează 3 variabile de intrare (A, B, C) iar pe intrările de date se conectează una din cele $2^{2^1} = 4$ funcții dependente de variabila a patra. Aceste funcții sunt 0, 1, D , \overline{D} . Se remarcă faptul că toate aceste funcții se pot realiza cu maxim un inversor. Implementarea nu necesită minimizare. Reprezentarea funcției în diagrama V-K permite construirea unei diagrame reduse urmată de trecerea la implementare.

- a) Diagrama V-K a funcției considerate, diagrama V-K redusă și implementarea cu MUX 8:1 sunt reprezentate în figura 9.27.

9.4 Pentru cei ce vor să devină profesioniști

Multiplexor 2:1 modelat cu operator ternar, specificație continuă (cod Verilog)

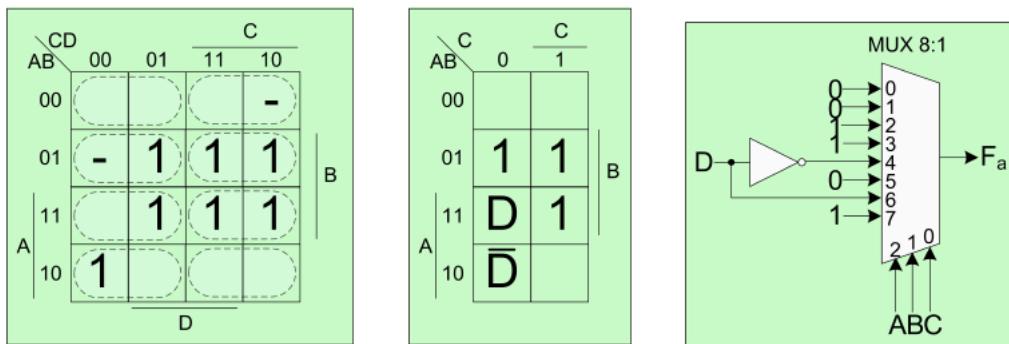


Figura 9.27 F_a , problema 27-a: diagrama V-K, diagrama V-K redusă și implementare cu MUX 8:1 a unei funcții de 4 intrări.

```
assign Y = S ? I1 : IO;
```

Multiplexor 2:1 modelat cu specificație de decizie (cod Verilog)

```
always @(*)
if (S) Y <= I1;
else
  Y <= IO;
```

Multiplexor 4:1 modelat cu specificație "case" (cod Verilog)

```
always @(*)
case (S)
  2'b11: Y <= I3;
  2'b10: Y <= I2;
  2'b01: Y <= I1;
  default: Y <= IO;
endcase
```

Multiplexor 4:1 cu validare (cod Verilog)

```
always @(*)
if (E)
  case (S)
    2'b11: Y <= I3;
    2'b10: Y <= I2;
    2'b01: Y <= I1;
    default: Y <= IO;
  endcase
else
  Y <= '1b0;
```

Multiplexor de dimensiune generică (cod Verilog)

```
parameter N = 4;
wire[N-1:0] S; // selectie
wire[(1<<N)-1:0] I; // date de intrare
wire Y; // iesire
assign Y = I[S];
```

Demultiplexor 1:2 modelat cu specificație continuă

```
assign Y1 = S & I;
assign Y0 = ~S & I;
```

Demultiplexor de dimensiune generică

```
parameter N = 4;
wire[N-1:0] S; // selectie
wire[(1<<N)-1:0] Y; // date de iesire
wire I; // intrare
assign Y = I<<S;
```

1. Să se conceapă un circuit de adunare a doi operanzi de 8 biți proveniți din 4 surse posibile și stocarea rezultatului în aceleși posibile locații.

2. Să se implementeze funcția $F(A, B, C, D, E) = \sum(3, 6, 7, 10, 11, 19, 22, 26, 27, 30, 31)$ conectând intrarea D pe intrarea de validare a unui multiplexor 16:1.

Implementați aceeași funcție cu un MUX 8:1 cu intrare de validare și un inversor.

Soluție

Diagrama V-K asociată funcției este prezentată în figura 9.28.

		CDE											
		AB		000	001	011	010	110	111	101	100		
A	00	0	1	1	3	2	1	6	1	7	5	4	
	01	8	9	1	1	10		14	15	13	12		
	11	24	25	1	27	1	26	1	30	1	31	29	28
	10	16	17	1	19	18	1	22	1	23	21	20	
		E				D				E			

Figura 9.28 Diagrama V-K asociată funcției de la problema 2.

O funcție poate fi implementată cu un multiplexor care are una din intrări conectată la intrarea de validare dacă expresia funcției se scrie sub forma unei funcții AND între o intrare și o funcție ce depinde de celelalte variabile. În acest caz, funcția trebuie scrisă ca $F(A, B, C, D, E) = D \cdot G(A, B, C, E)$.

Din diagrama V-K se observă că funcția F are valoarea 1 numai în căsuțele pentru care $D = 1$. Din diagrama V-K, prin minimizare și excluderea variabilei D , se poate deduce expresia funcției:

$$G(A, B, C, E) = B \cdot \bar{C} + \bar{C} \cdot E + A \cdot B + \bar{A} \cdot \bar{B} \cdot C + A \cdot C \cdot \bar{E},$$

$$G(A, B, C, E) = \sum(1, 2, 3, 4, 5, 9, 10, 12, 13, 14, 15).$$

Ca alternativă analitică, funcția F se poate prelucra astfel:

$$\begin{aligned} F(A, B, C, D, E) &= \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D \cdot E + \bar{A} \cdot \bar{B} \cdot C \cdot D \cdot \bar{E} + \bar{A} \cdot \bar{B} \cdot C \cdot D \cdot E + \bar{A} \cdot B \cdot \bar{C} \cdot D \cdot \bar{E} + \bar{A} \cdot B \cdot \bar{C} \cdot D \cdot E + \\ &= D \cdot (\bar{A} \cdot \bar{B} \cdot \bar{C} \cdot E + \bar{A} \cdot \bar{B} \cdot C \cdot \bar{E} + \bar{A} \cdot B \cdot \bar{C} \cdot D \cdot E + A \cdot B \cdot C \cdot D \cdot \bar{E} + A \cdot B \cdot C \cdot D \cdot E = \\ &= D \cdot G(A, B, C, E), \text{ unde } G(A, B, C, E) = \sum(1, 2, 3, 4, 5, 9, 10, 12, 13, 14, 15). \end{aligned}$$

Circuitul de implementare cu MUX 16:1 este prezentat în figura 9.29.

- Proiectați un multiplexor 16:1 utilizând două multiplexoare 8:1 și un multiplexor 2:1.
Proiectați un multiplexor 16:1 utilizând multiplexoare 4:1.
Proiectați un multiplexor 16:1 utilizând numai multiplexoare 2:1.
- Proiectați un circuit care selectează o intrare de date din cele 3 intrări existente pe baza unui cod de selecție astfel: $AB = 00$, selectează D_0 , $AB = 01$, selectează D_1 și $AB = 1X$, selectează D_2 (s-a notat cu X o valoare indiferentă). Utilizați un circuit multiplexor 2:1.
- Descrieți modul de implementare a unui multiplexor 128:1 utilizând circuite multiplexor 16:1.



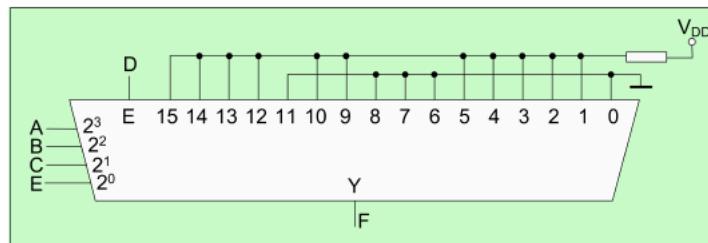


Figura 9.29 Circuitul de implementare cu MUX 16:1 cu intrare de validare a funcției de la problema 2.

Descrieți modul de implementare a unui multiplexor 128:1 utilizând circuite multiplexor 4:1.

Descrieți modul de implementare a unui multiplexor 256:1 utilizând circuite multiplexor 8:1.

Soluție

Un multiplexor 128:1 are 7 intrări de selecție ($2^7 = 128$), 128 de intrări de date și o ieșire. Împărțind numărul de intrări de date la numărul de intrări de date ale multiplexorului cu care se face implementarea rezultă numărul de multiplexoare necesare pentru primul nivel. Sunt necesare $128/16 = 8$ multiplexoare 16:1. Cele 8 multiplexoare de pe primul nivel vor fi selectate cu 4 biți de intrare (cei mai puțin semnificativi). Cei mai semnificativi 3 biți de intrare se vor conecta pe intrările de selecție ale unui alt multiplexor 16:1. Acesta va avea intrările de date conectate la ieșirile multiplexoarelor de pe primul nivel. Datorită faptului că pe al doilea nivel se implementează practic un MUX 8:1 cu un MUX 16:1, intrarea de selecție cea mai semnificativă se conectează la masa ("0" logic), iar intrările de date cu index între 8 și 15 vor fi indiferente (se vor conecta la masa).

6. Proiectați un demultiplexor 1:32 utilizând un număr minim de circuite demultiplexor 1:4 și 1:8.
7. Proiectați un circuit pentru conversia paralel-serie a datelor reprezentate pe 8 biți.
Proiectați un circuit pentru conversia serie-paralel a datelor reprezentate pe 8 biți.

Soluție

Circuitul de conversie a datelor paralel-serie constă dintr-un multiplexor 8:1 a cărui intrări de selecție provin de la un circuit care generează secvențial toate combinațiile de 3 biți (numărător). Datele de intrare reprezentate pe 8 biți se aplică pe intrările de date. Datele seriale se obțin pe ieșirea multiplexorului.

Circuitul de conversie a datelor serie-paralel constă dintr-un demultiplexor 1:8 a cărui intrări de selecție provin de la un circuit care generează secvențial toate combinațiile de 3 biți (numărător). Datele seriale se aplică pe intrarea demultiplexorului. Datele de ieșire reprezentate pe 8 biți se obțin pe ieșirile demultiplexorului.

În cazul funcționării împreună a circuitelor de conversie paralel-seriel și serie-paralel trebuie asigurată sincronizarea celor două coduri pentru a se reface corect poziția biților în cuvântul de 8 biți. Figura 9.30 prezintă soluția de transmitere a datelor cu serializare-deserializare.

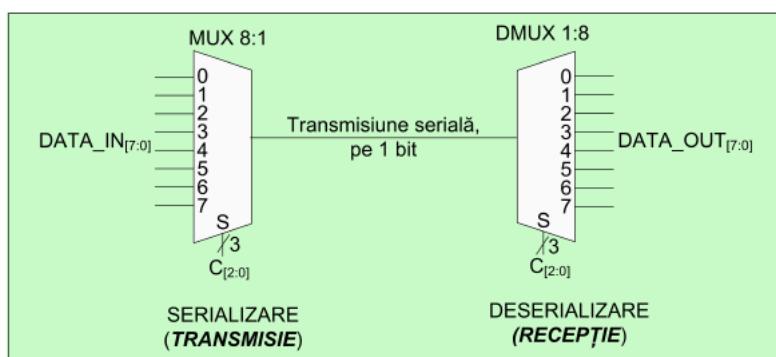


Figura 9.30 Soluția de transmitere a datelor cu serializare-deserializare implementată cu multiplexor-demultiplexor.

