

فصل دوم

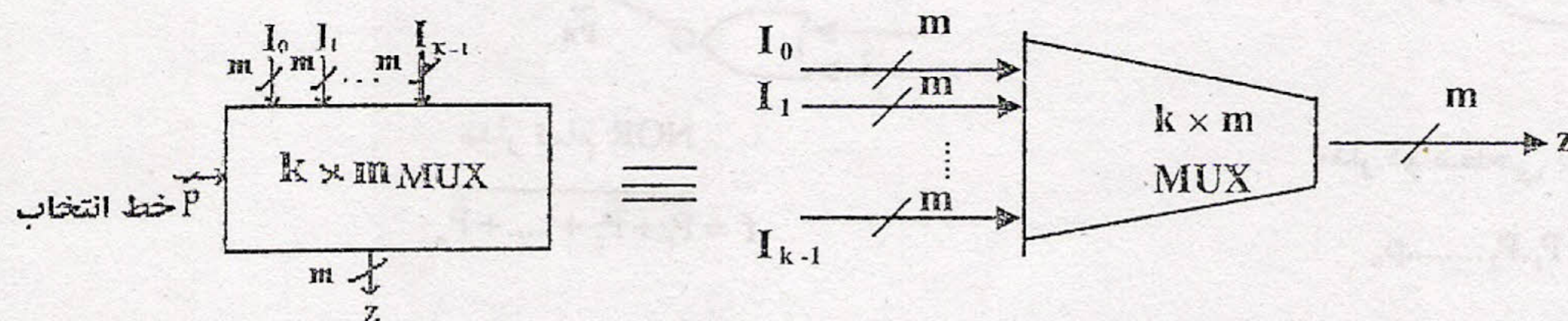
طراحی مدارهای ترکیبی MSI و کاربرد آن‌ها

مدارهای مجتمع: *Integated circuits – chips (IC)*

طراحی مدارهای تمام NAND و تمام NOR باعث پیدایش IC ها شده‌اند که انواع آن عبارتند از: SSI: که حداکثر شامل Gate - 10 می‌باشند. از این نوع IC فقط به عنوان Gate می‌توان استفاده نمود. MSI: در این دسته از مدارهای مجتمع حداکثر 100 تا gate وجود دارد، به عنوان مثال می‌توان یک 32-bit Adder را بر روی آن پیاده سازی نمود. LSI: این دسته از تراشه‌ها حداکثر 1000 gate را در خود جای داده‌اند. که می‌توان یک CPU را بر روی آن پیاده‌سازی نمود. VLSI: حداکثر شامل Gate - 10000 می‌باشد و می‌توان یک Microprocessor را در آن جای داد. USI (VLSI): حداکثر شامل Gate - 100000 می‌باشد. در این درس تعدادی از MSI ها طرح می‌کنیم که در درس معماری کامپیوتر کاربرد دارد.

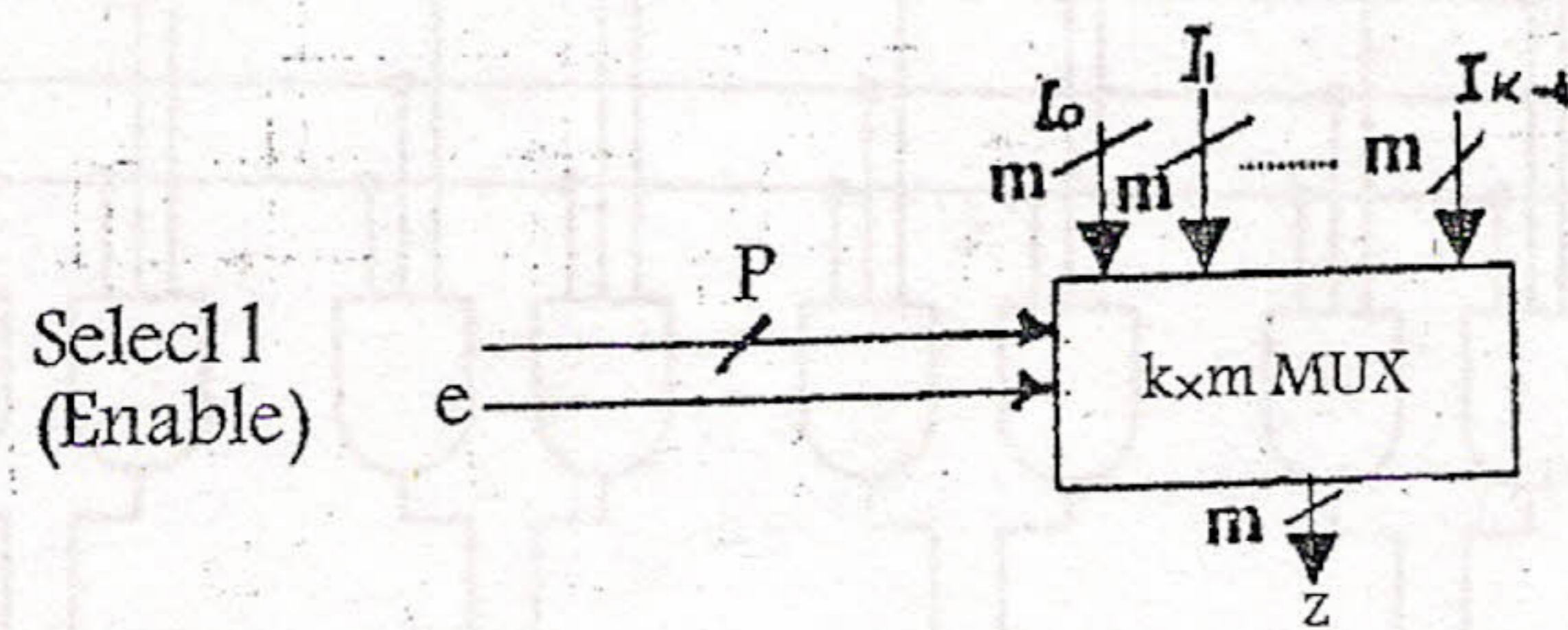
تسهیم کننده: *(MUX) Multiplexer*

تسهیم کننده مدار ترکیبی است که از بین چندین منبع ورودی متمایز یکی را انتخاب و اطلاعات شامل در آن را به یک خروجی منفرد و مشترک هدایت می‌کند. گزینش منابع ورودی به وسیله خطوط انتخاب انجام می‌گیرد.



برای این که گزینش منابع به سهولت انجام گیرد، باید تساوی $K = 2^p$ برقرار باشد.

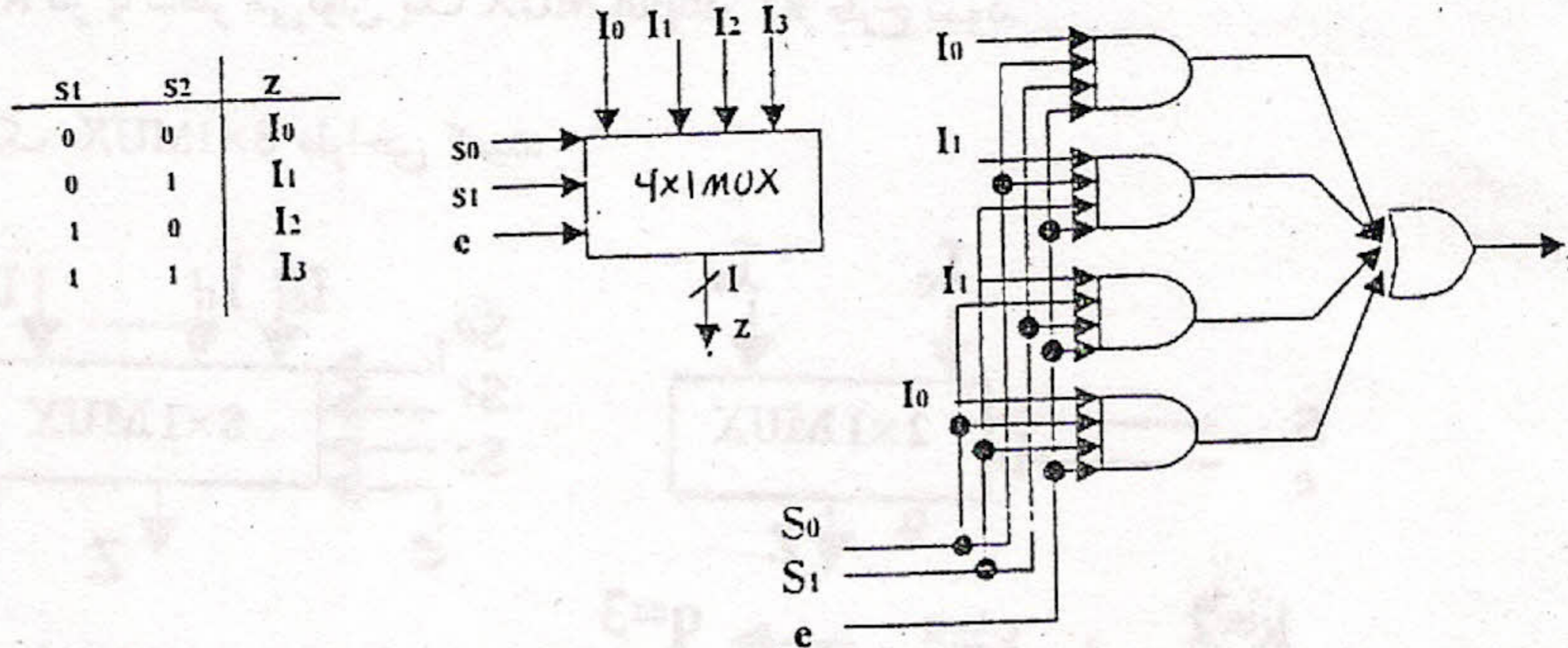
در صورتی که MUX دارای K منبع ورودی و هر منبع شامل اطلاعات m بیتی باشد، آنرا $(K \times m \text{ MUX})$ Kinput - m bit MUX گویند.



توجه: اگر $e=1$ باشد، آن گاه مدار به حالت عادی کار می کند. اگر $e=0$ باشد، مدار کار نخواهد کرد (خاموش خواهد بود).

مثال: مطلوب است $4 \times 1 \text{ MUX}$

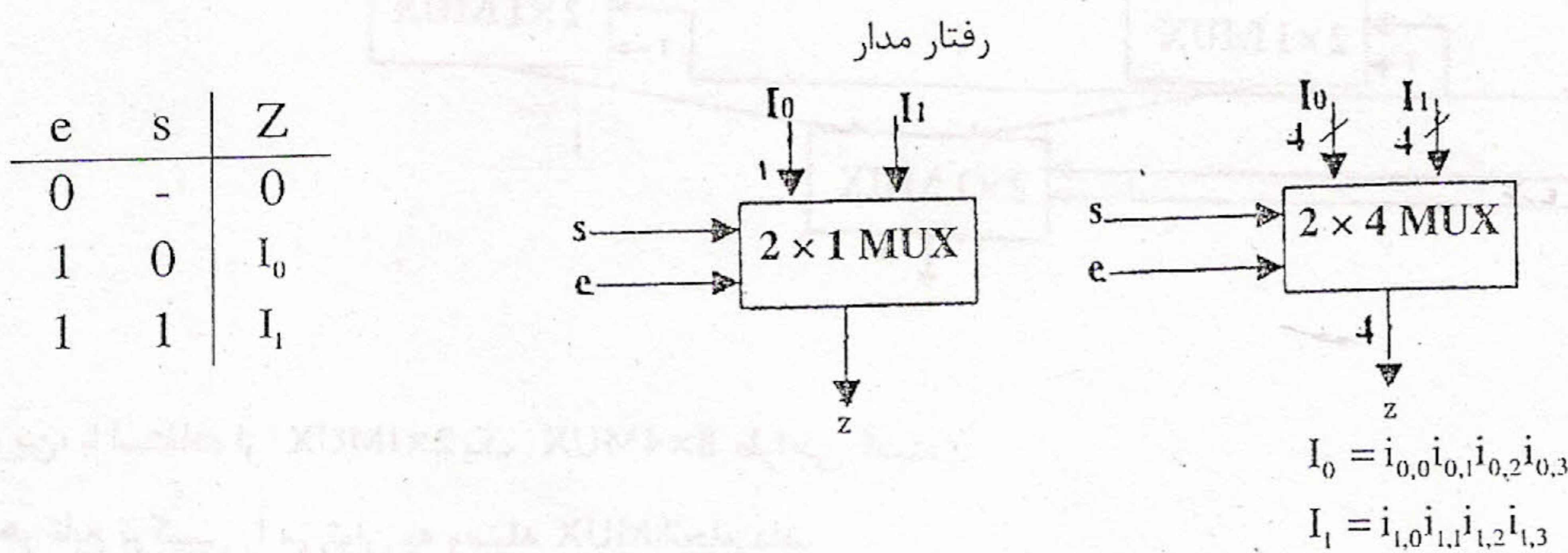
$$\left. \begin{matrix} k = 2^p \\ 4 = 2^2 \end{matrix} \right\} \Rightarrow P = 2$$

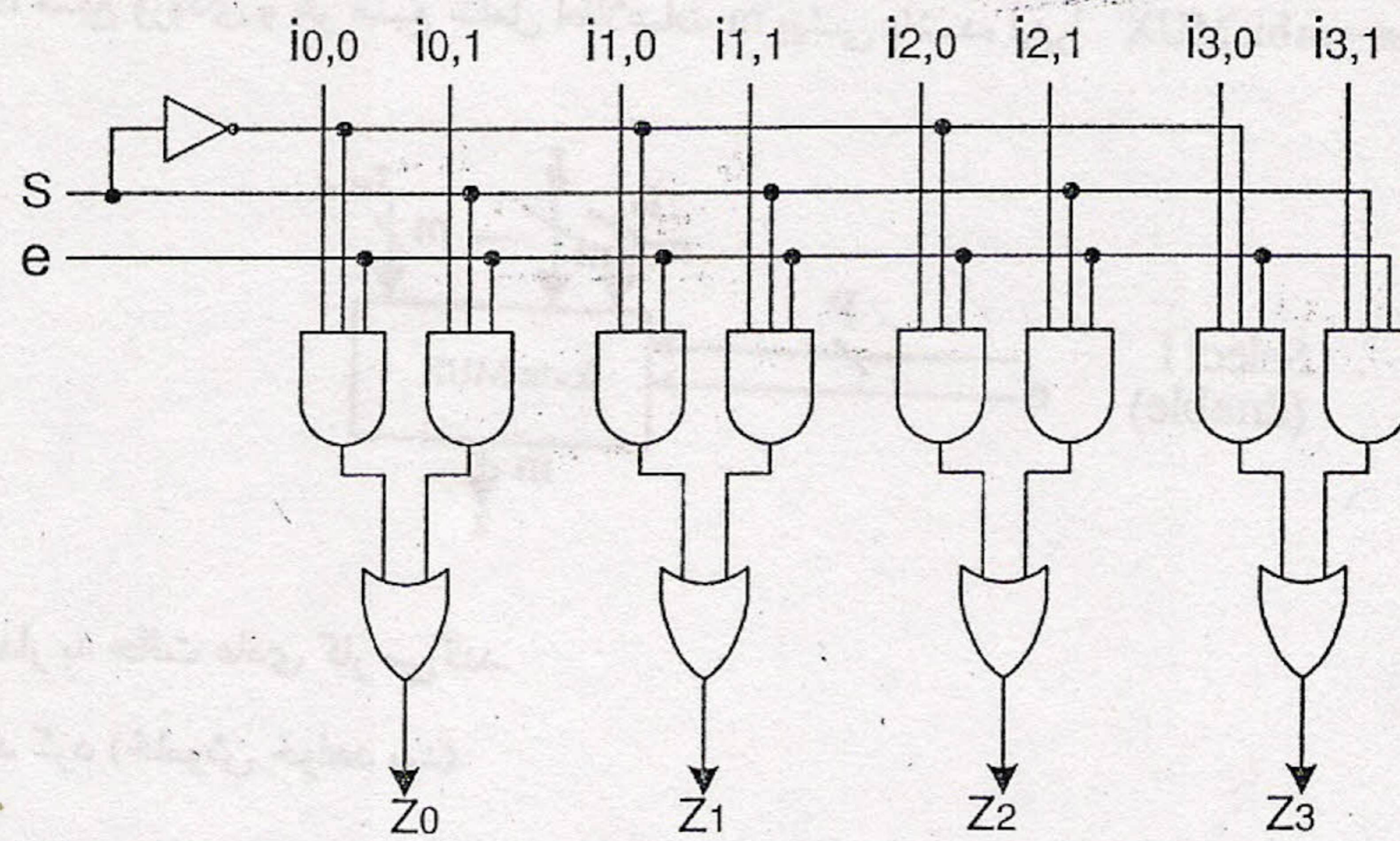


کاربردهای MUX:

- (۱) برای هدایت اطلاعات از بین چند منبع متمایز به سوی یک مقصد مشترک.
- (۲) برای طراحی سیستم BUS (گذرگاه یا معبر) مطالعه این قسمت در درس معماری کامپیوتر خواهد بود.
- (۳) با اتصال شبکه‌ای MUX ها می توان تعداد بیت‌های منابع ورودی را افزایش داد.

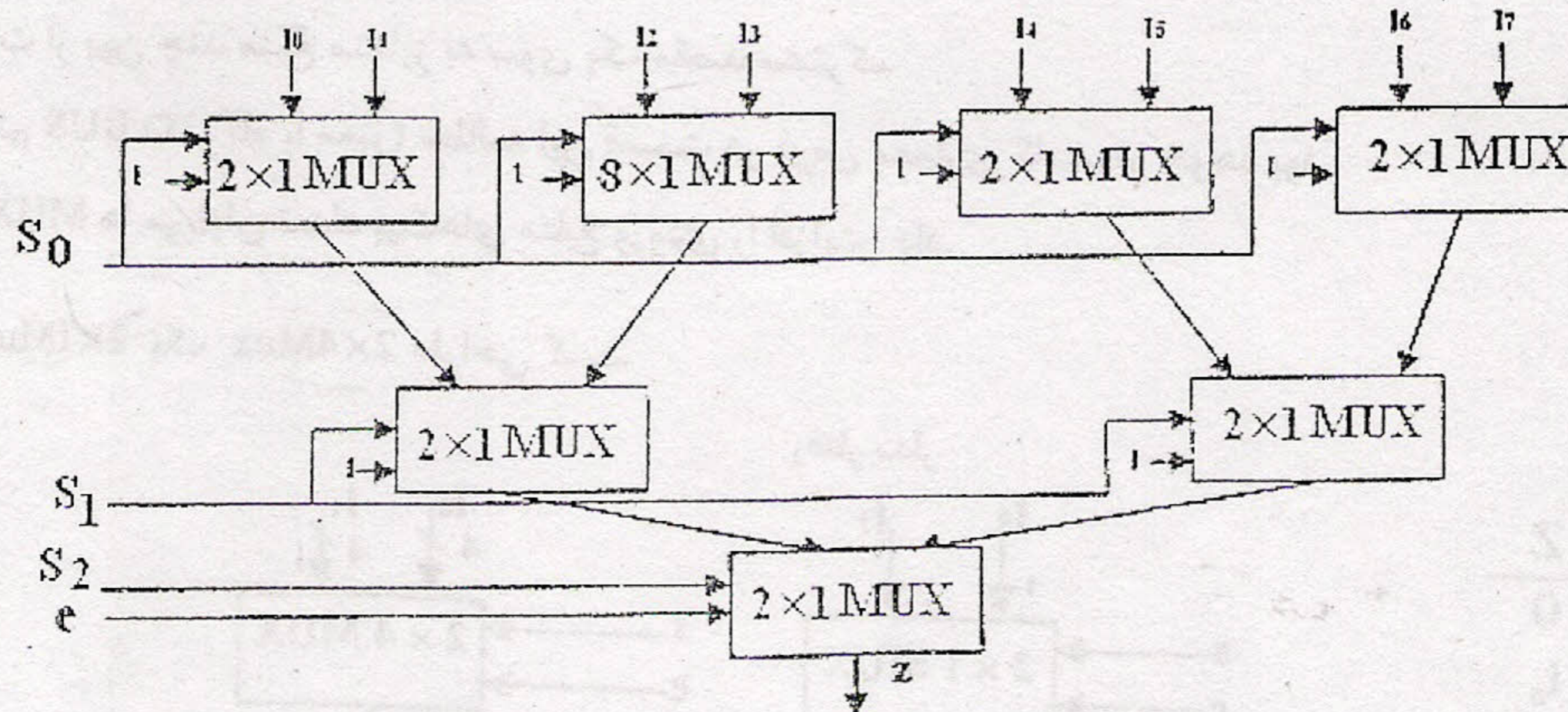
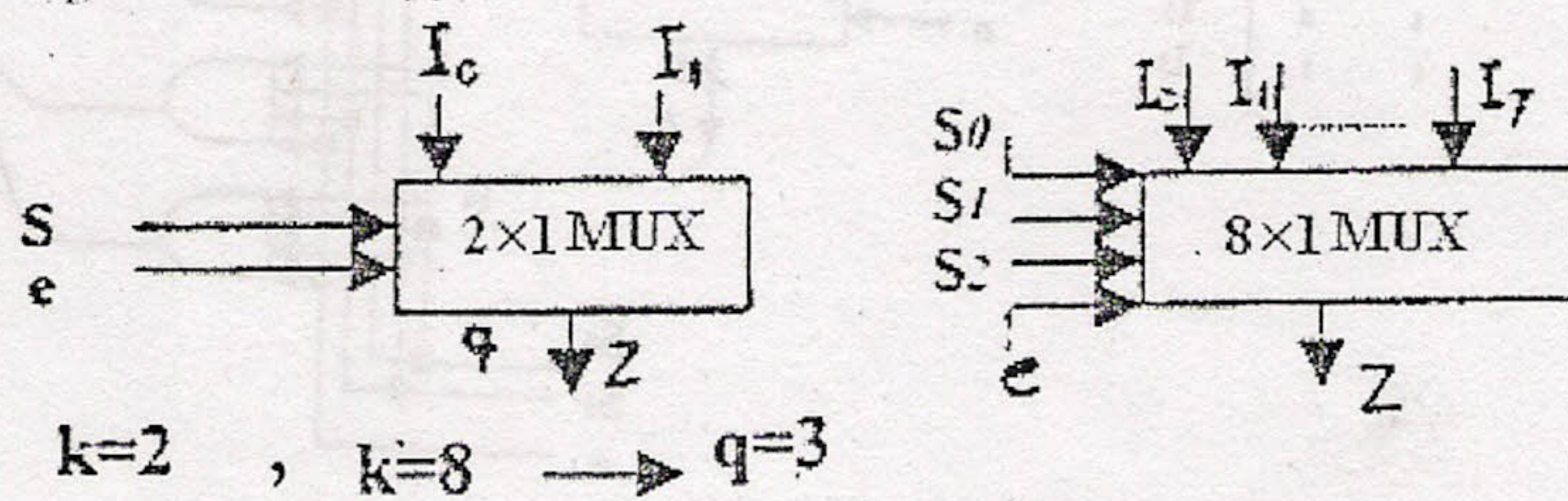
مثال: با استفاده از $2 \times 1 \text{ Mux}$ یک $2 \times 4 \text{ Mux}$ طراحی کنید.





۴) با اتصال درختی K-input MUX در q سطر می توان یک k^q-input MUX طرح نمود.

مثال: با استفاده از 2x1 MUX یک 8x1 MUX طراحی کنید.



تمرین: با استفاده از 2x1 MUX یک 8x4 MUX طراحی کنید.

۵. هر تابع ترکیبی را می توان به وسیله MUX انجام داد.

روش a: هر تابع ترکیبی n متغیری را می توان به وسیله یک 2ⁿ x 1 MUX انجام داد، مشروط بر این که n متغیر تابع به عنوان n خط انتخاب (Select) MUX به کار روند. این روش پیاده سازی را MUX نوع 0- می نامند، زیرا از صفر متغیر در منابع ورودی استفاده می شود.

روش b: هر تابع ترکیبی n متغیری را می توان به وسیله $2^{n-1} \times 1\text{MUX}$ انجام داد، مشروط بر این که $(n-1)$ متغیر تابع به عنوان $(n-1)$ خط انتخاب وصل شود و از متغیر باقی مانده در ورودی MUX استفاده بشود. این روش پیاده سازی را MUX نوع 1- می نامند زیرا فقط از یک متغیر در منابع ورودی استفاده می شود.

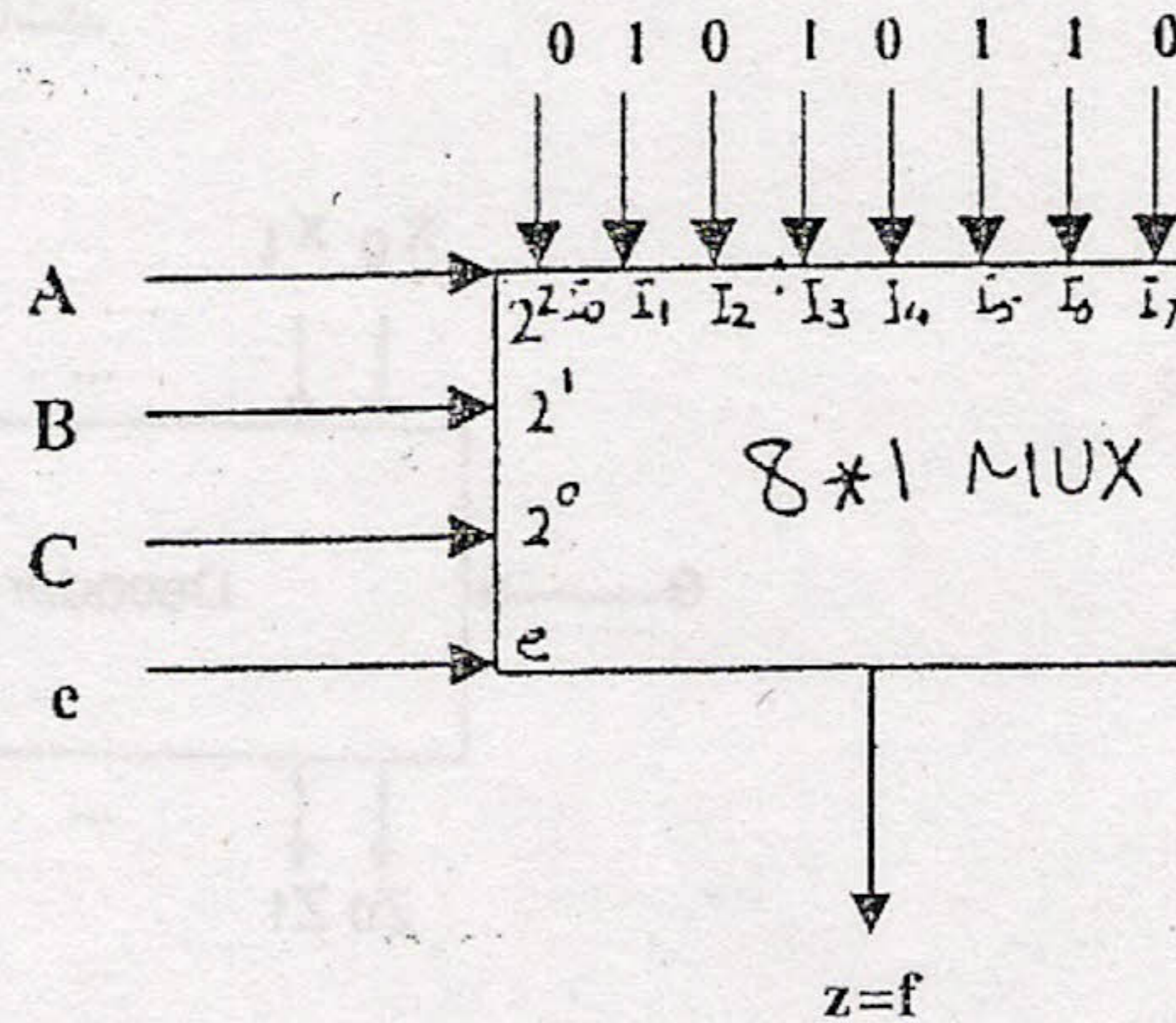
مثال: تابع زیر را به وسیله MUX انجام دهید.

$$f = (A,B,C) = \sum(1,3,5,6)$$

$$n = 3$$

$$2^n = 2^3 = 8 \Rightarrow 8 \times 1\text{MUX}$$

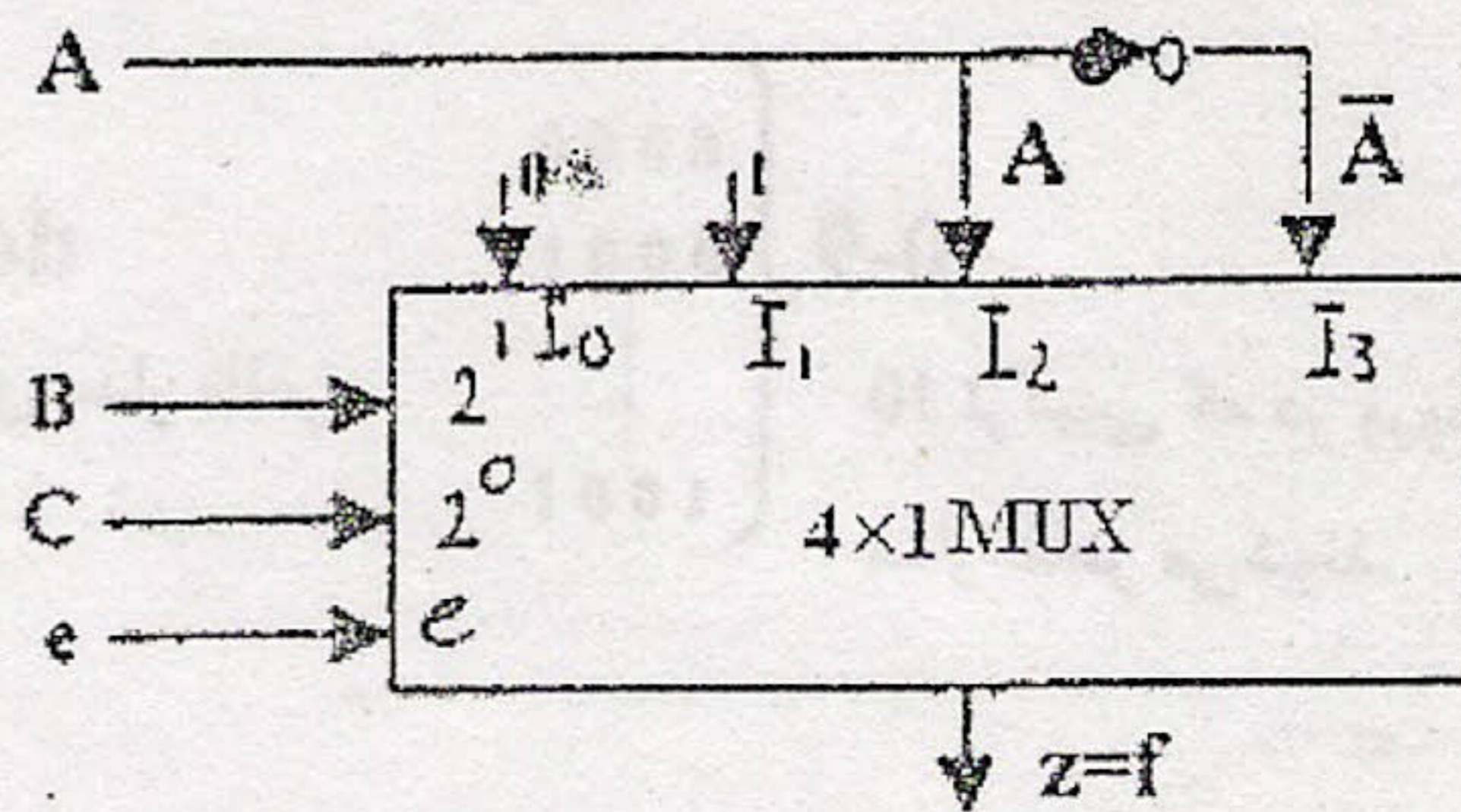
A	B	C	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



روش b:

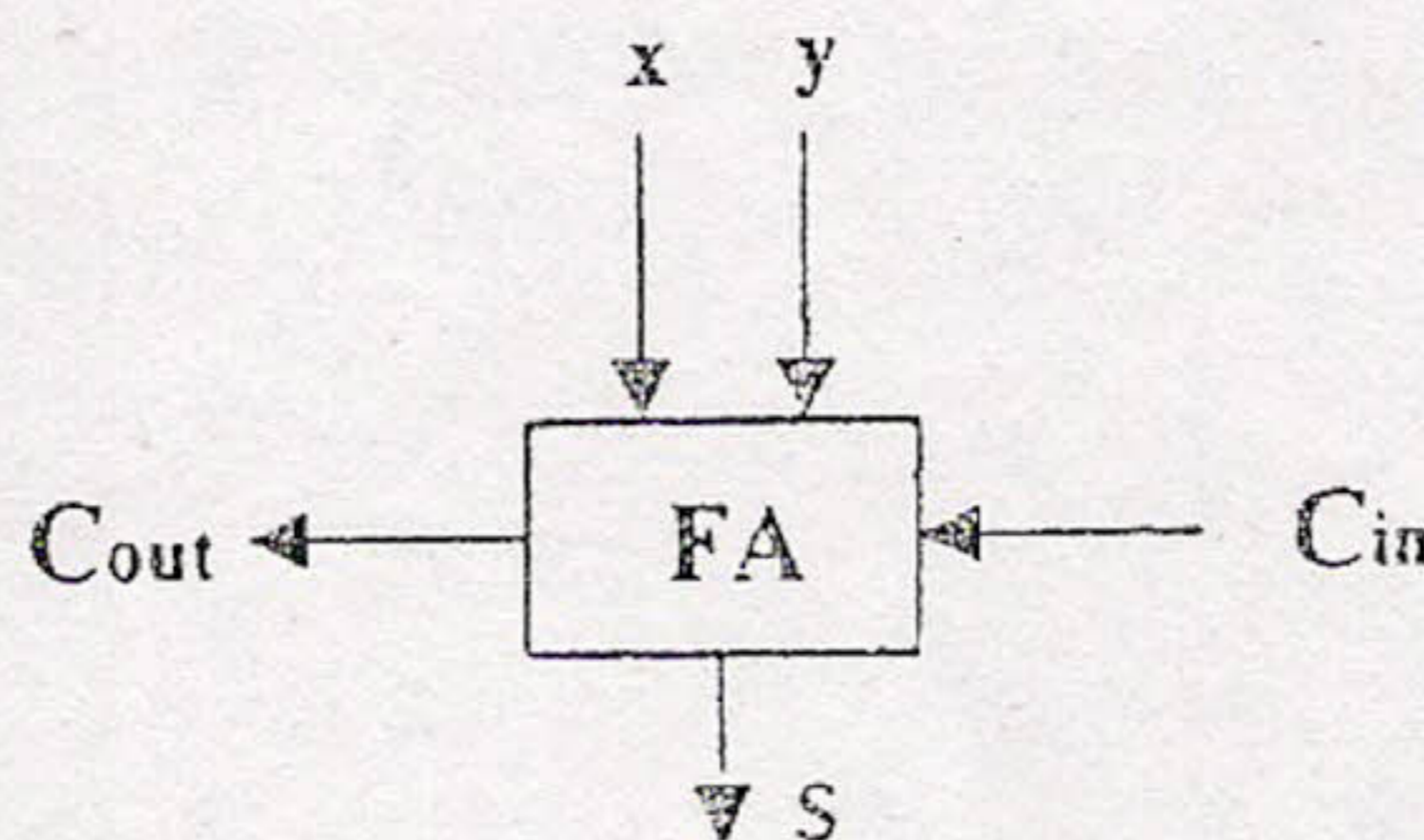
$$n = 3 \Rightarrow 2^{n-1} = 2^{3-1} = 2^2 = 4 \Rightarrow 4 \times 1\text{MUX}$$

	I_0	I_1	I_2	I_3
\bar{A}	0	①	2	③
A	4	⑤	⑥	7
	0	1	A	\bar{A}



در روش b ستون های جدول را با منابع ورودی MUX و سطرهای جدول را با مکمل متغیر و خود متغیر باقی مانده نام گذاری می کنیم. در سطری که با مکمل متغیر نام گذاری شده است، اندیس دهدهی مین ترم هایی را می نویسیم که متغیر باقی مانده در آن صفر و و در سطری که با خود متغیر نام گذاری شده است، اندیس دهدهی مین ترم هایی را می نویسیم که متغیر باقی مانده در آن یک می باشد. سپس دور مین ترم هایی که مقدار تابع به ازای آن یک می باشد، دایره رسم می کنیم. اگر ستونی اصلا دایره نداشته باشد، آن منبع را به مقدار ثابت صفر و اگر همگی دایره داشت، آن منبع را به یک ثابت و اگر ستونی فقط یک دایره داشته باشد، آن منبع را به برچسب سطری وصل می کنیم که دایره در آن وجود دارد.

تمرین: با استفاده از MUX یک FULL-ADDER طراحی کنید.

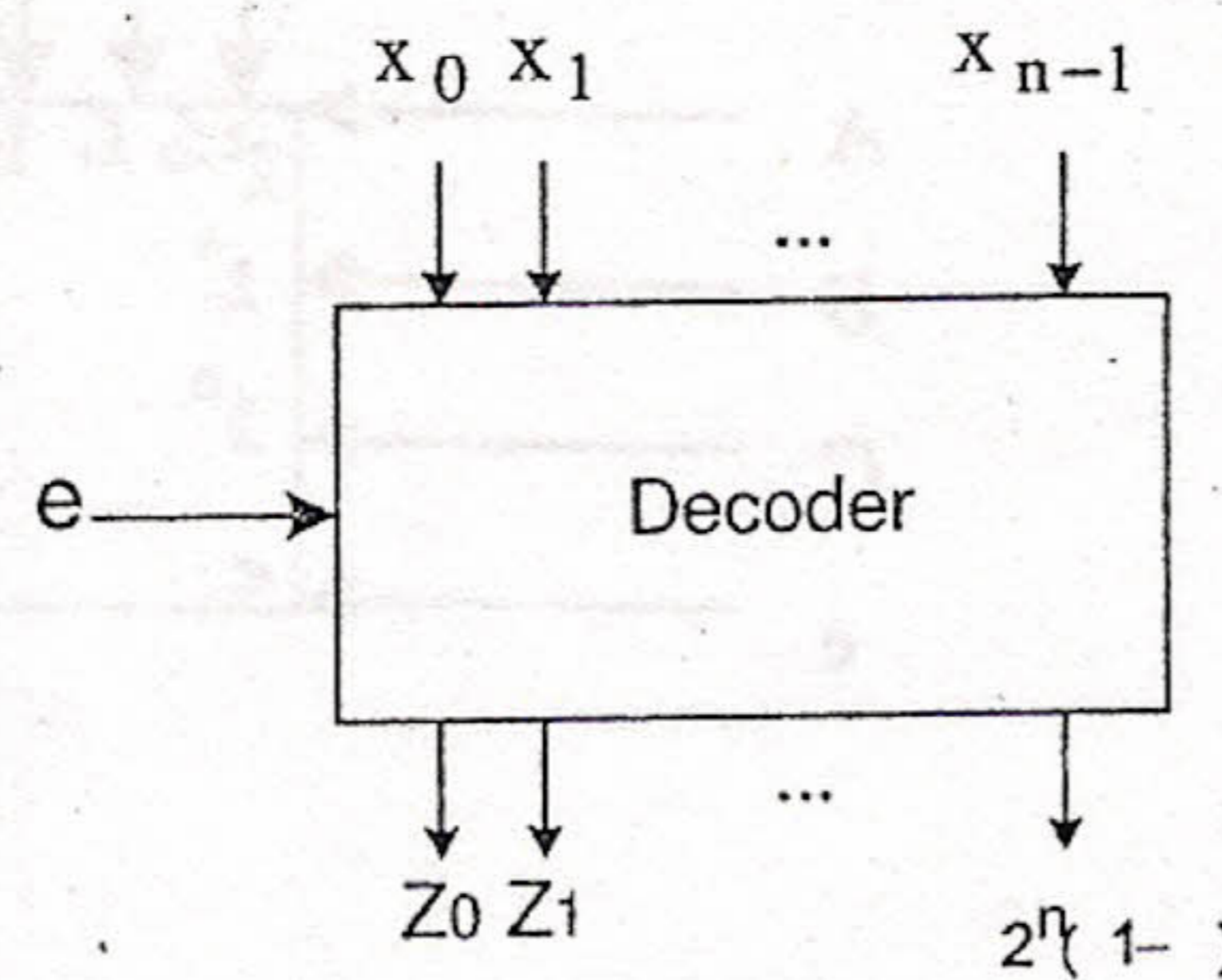


تمرین - با اعمال قضیه شانون به متغیرهای a, b تابع $f(a, b, c) = a + b\bar{c}$ را به وسیله MUX انجام دهید.

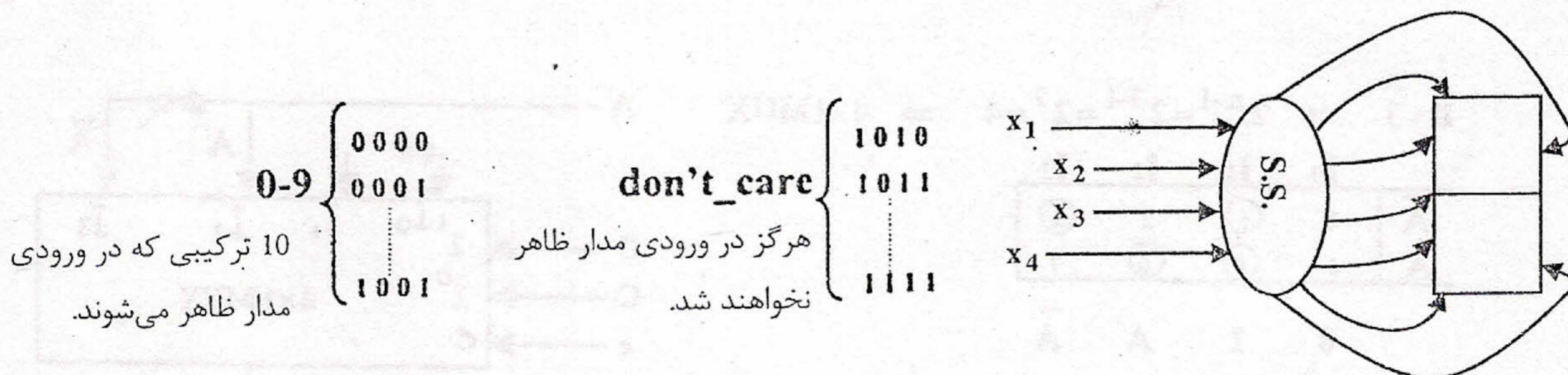
تمرین - با استفاده از جدول ارزش تابع $f(a, b, c, d) = \sum_1(4, 5, 6, 7, 10, 14)$ را به وسیله MUX های نوع 0 تا 3 طرح کنید.

مدار مجتمع رمز گشا: Decoder

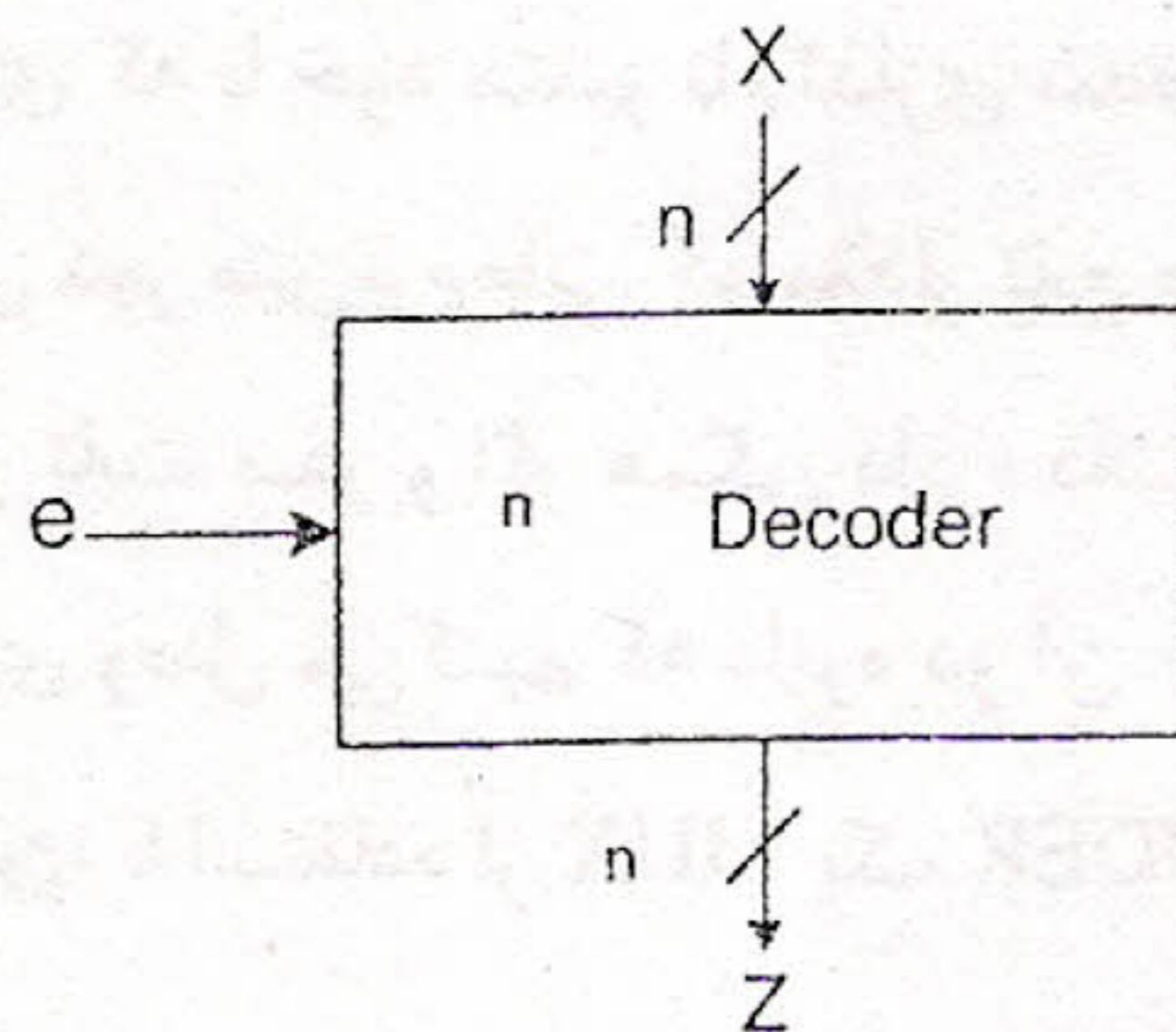
مدار ترکیبی است با n خط ورودی و 2^n خط خروجی که در آن به ازای هر یک از ترکیبات ورودی حداکثر 2^n خط خروجی فعال می‌شود. در واقع هر خروجی متناظر با یک Minterm می‌باشد.



در صورتی که در ورودی ترکیبات don't-care وجود داشته باشد، آن‌گاه تعداد خروجی‌ها کمتر از 2^n خواهد بود. مثلاً $m < 2^n$ مانند Seven - Segment Decoder که در آن تعداد خروجی‌ها $7 < 2^4$ می‌باشد.



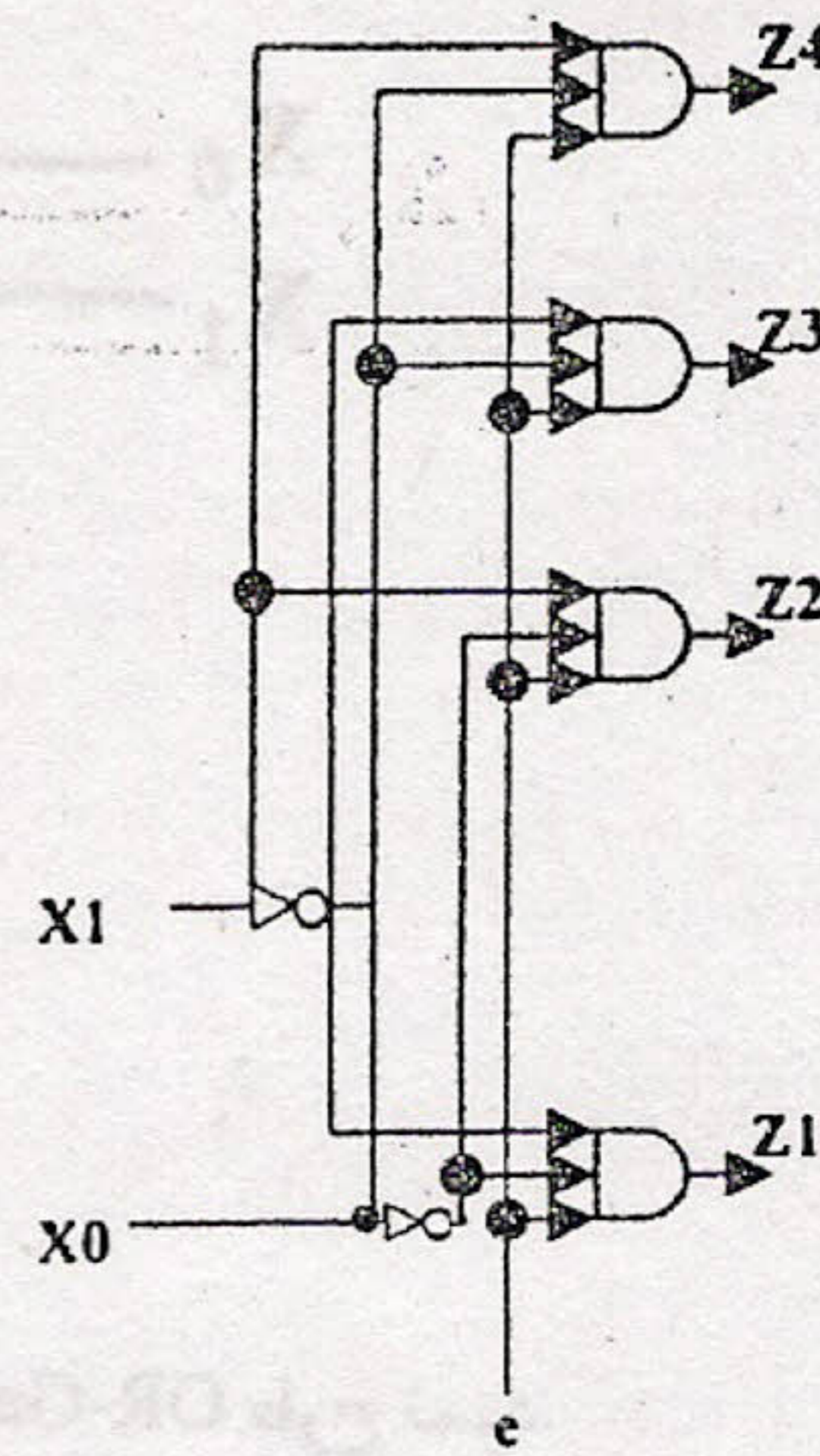
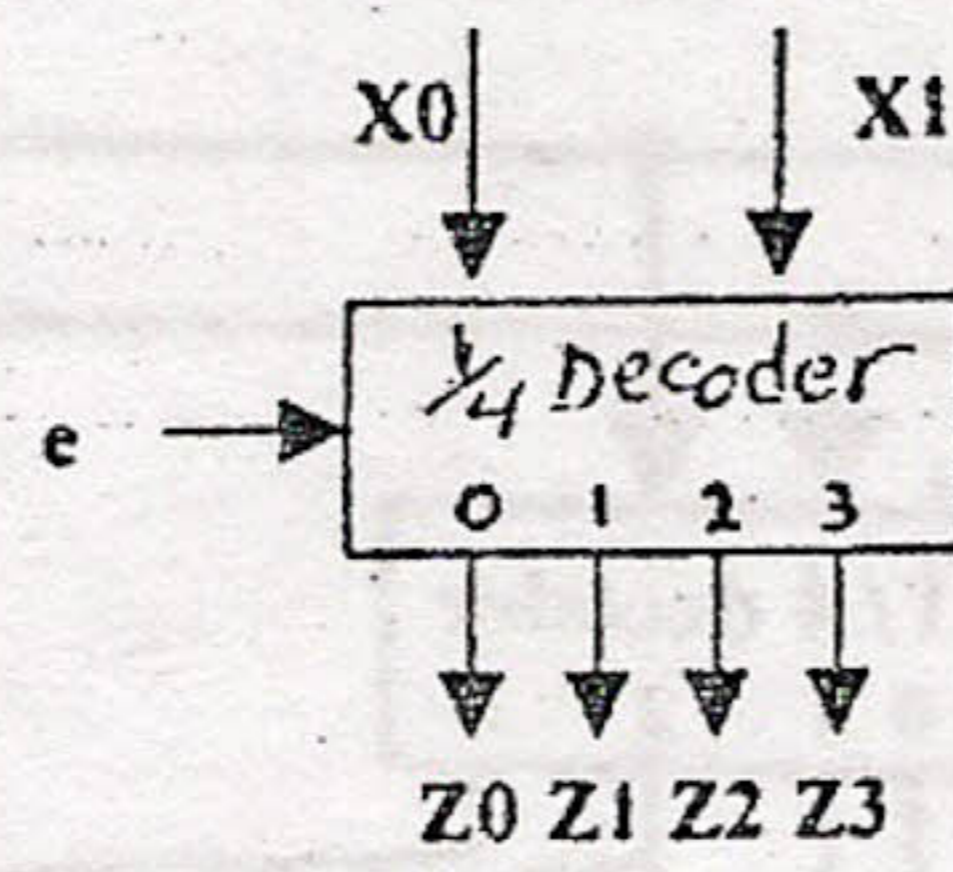
ولی آنچه در این درس منظور است، Decoder $\frac{1}{2^n}$ می‌باشد که در آن به ازای هر یک از ترکیبات ورودی تنها یک خروجی فعال و بقیه غیر فعال خواهند بود.



مثال: مطلوب است طراحی $\frac{1}{4}$ Decoder

$2^n = 4 \Rightarrow n = 2$

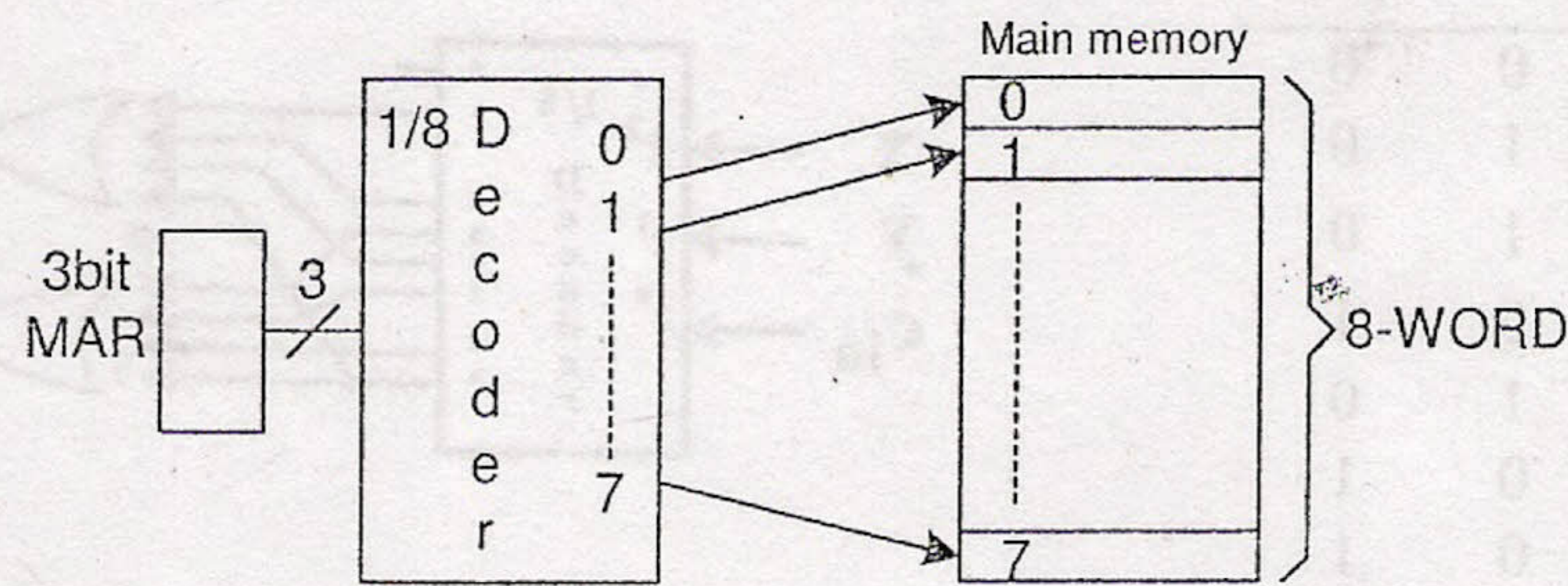
x_0	x_1	z_0	z_1	z_2	z_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



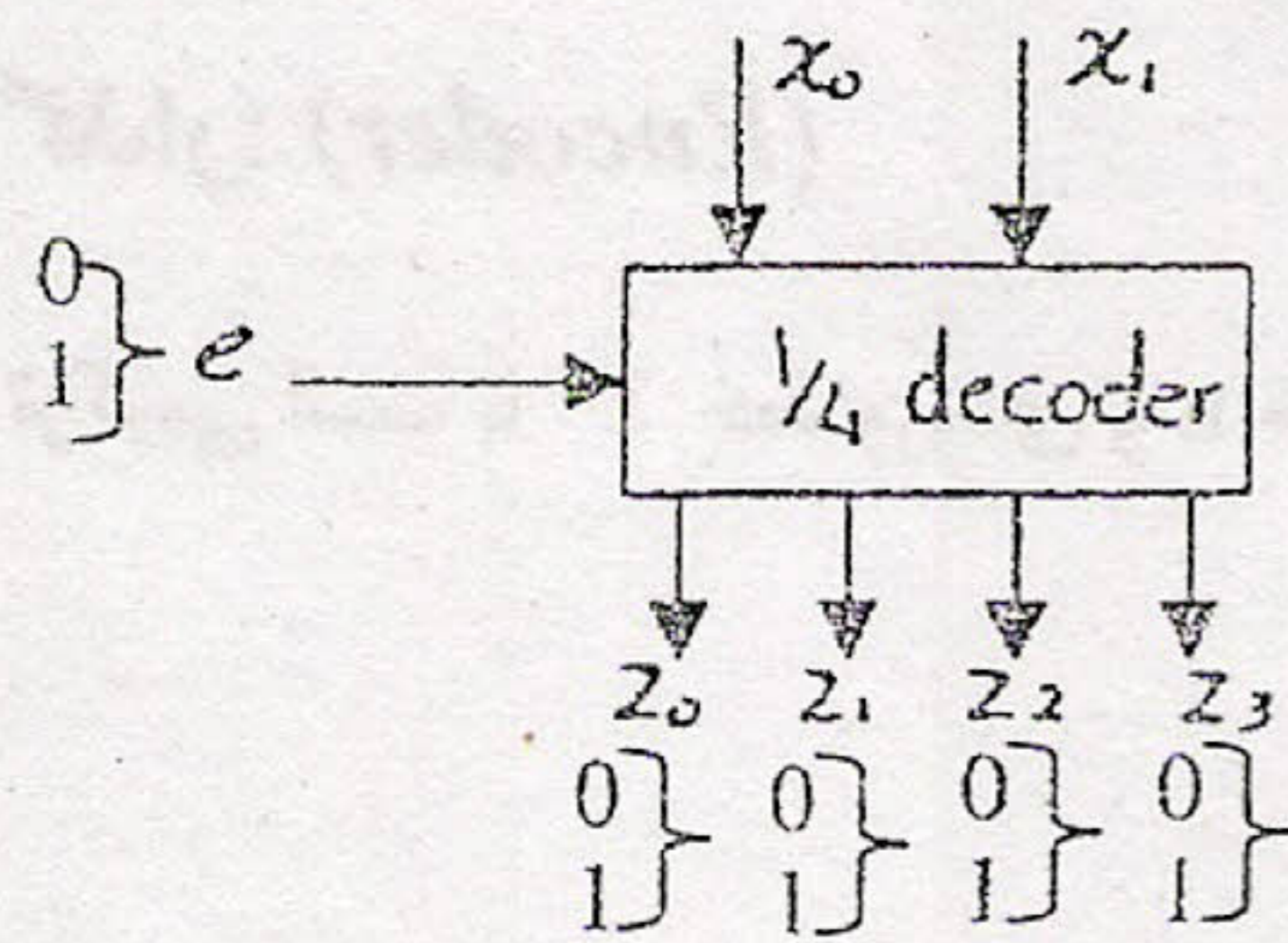
کاربردهای رمزگشا:

- (۱) برای تبدیل کدها به یکدیگر مانند: Seven Segment Decoder که ورودی Binary را در خروجی به ارقام Decimal تبدیل می‌کند.
- (۲) برای نشانی دهی کلمات حافظه اصلی: فرض کنید حافظه اصلی شامل ۸ کلمه باشد.

$8 = 2^3 \Rightarrow 3\text{-bit MAR}$



(۳) در صورتی که خط Enable رمزگشا را به عنوان منبع مشترک تک بیتی تلقی کنیم، آن‌گاه از Decoder می‌توان برای هدایت اطلاعات تک بیتی از یک منبع مشترک به یکی از چندین مقصد متمایز استفاده نمود. چنین مداری را DeMUX گویند.

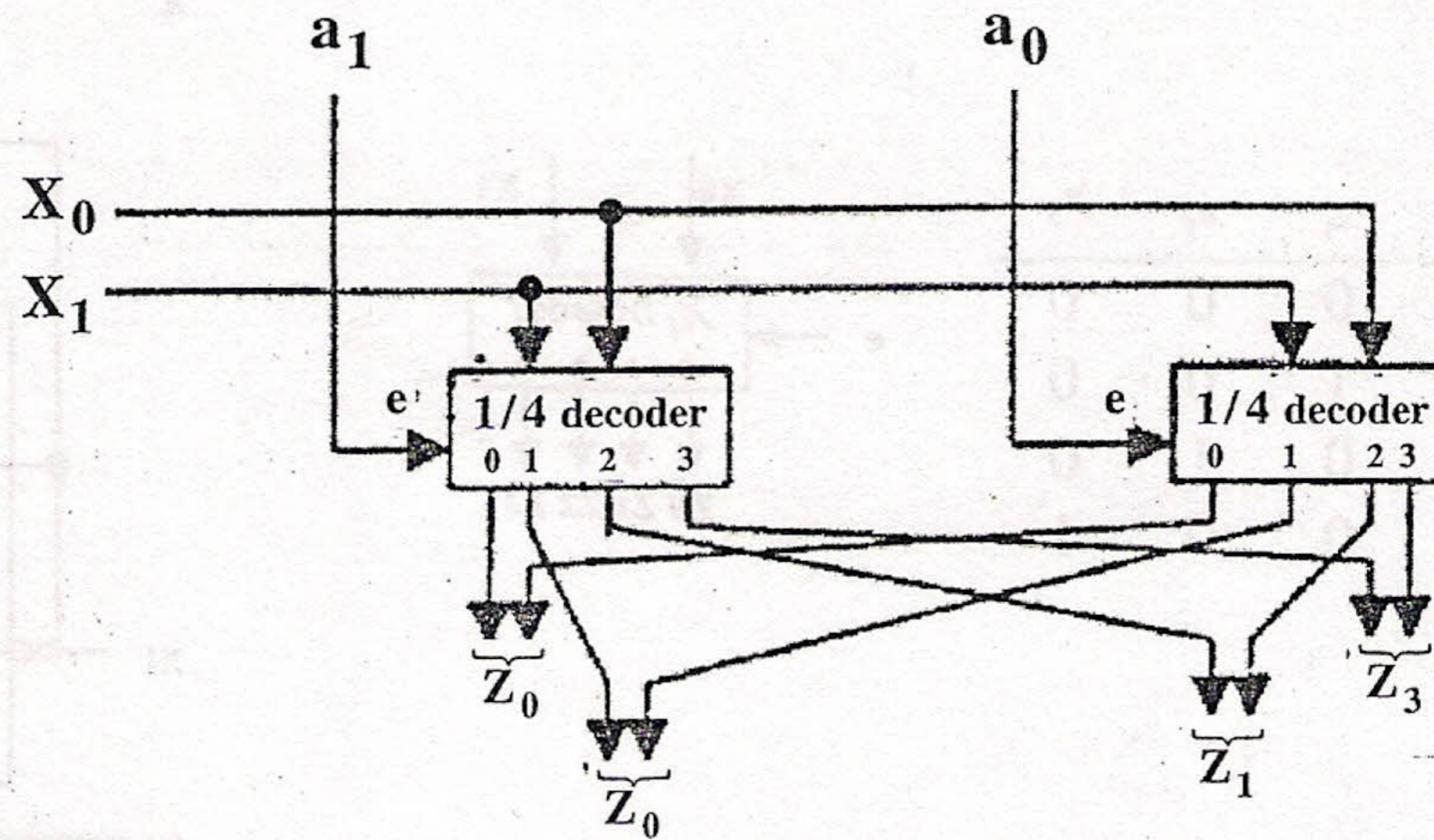


مثلاً $\frac{1}{4}$ Decoder \equiv 4-output 1-bit deMUX است که با دادن آدرس مناسب به x_0, x_1 ، می‌توان اطلاعات تک بیتی واقع در خط

enable را به یکی از چهار مقصد z_3, z_2, z_1, z_0 هدایت نمود.

به‌طور کلی $\frac{1}{2^n}$ Decoder \equiv 2^n -output 1-bit deMUX

مثال: با استفاده از Decoder یک 2-bit deMUX 4-output طرح کنید.



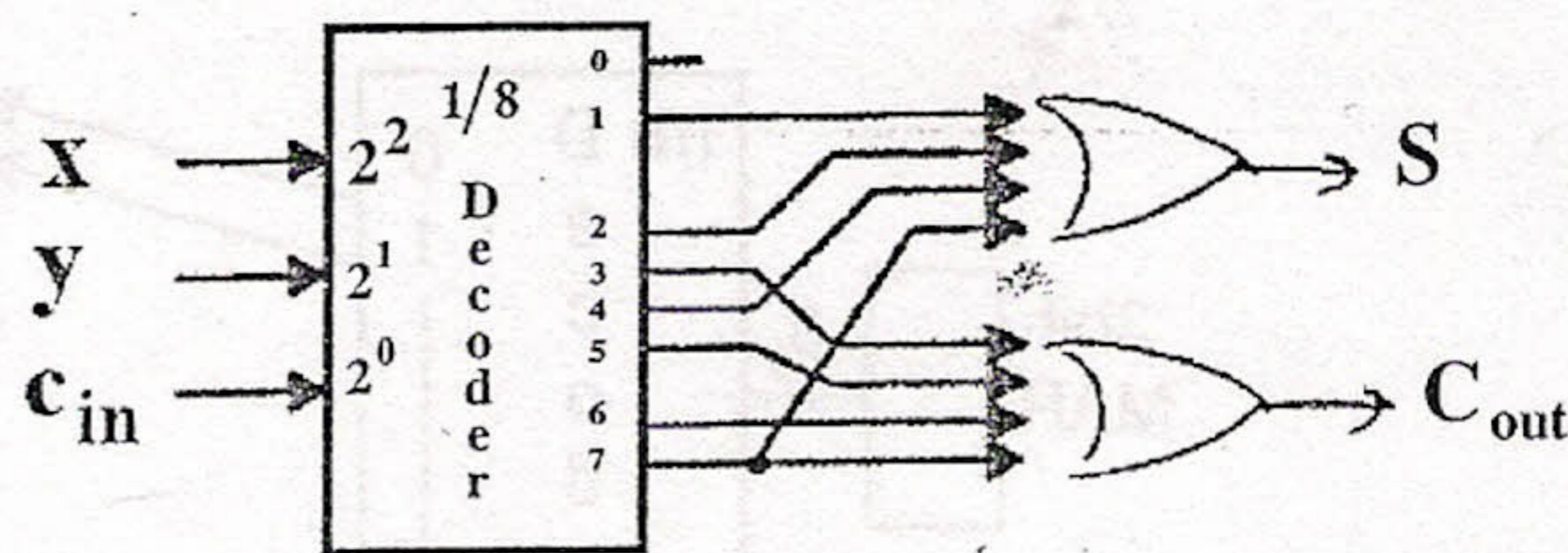
(۴) هر مدار ترکیبی با n ورودی و m خروجی را می توان به وسیله یک Decoder $\frac{1}{2^n}$ و m عدد OR-Gate طرح نمود.

مثال: با استفاده از Decoder یک Full-adder طرح کنید.

$$n = 3, 2^n = 8 \Rightarrow \frac{1}{8} \text{ Decoder}$$

$$m = 2 \Rightarrow 2 \text{ OR-Gate}$$

x	y	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

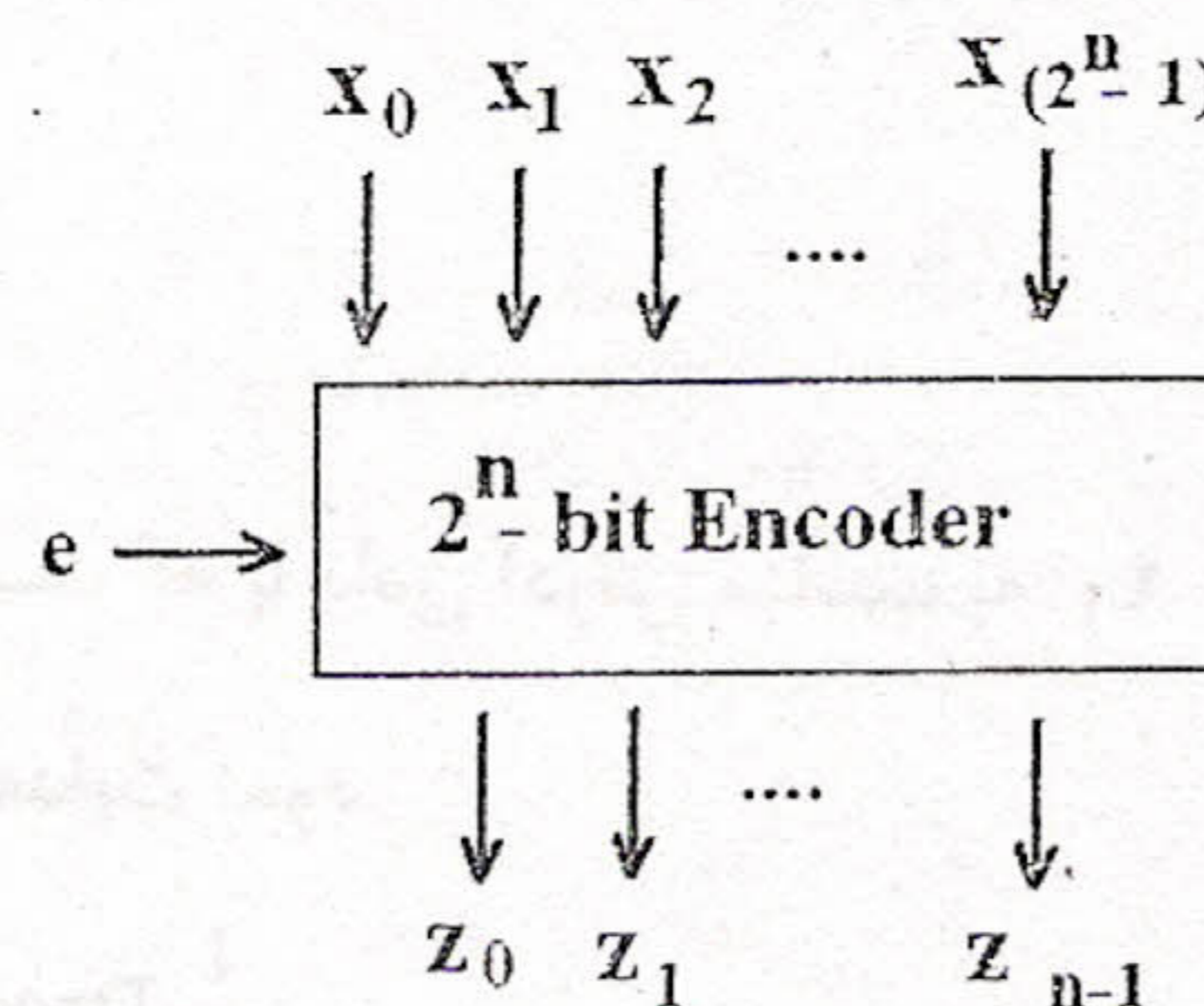


$$S(x, y, C_m) = \sum_i (1, 2, 4, 7)$$

$$C_{out}(x, y, C_m) = \sum_i (3, 5, 6, 7)$$

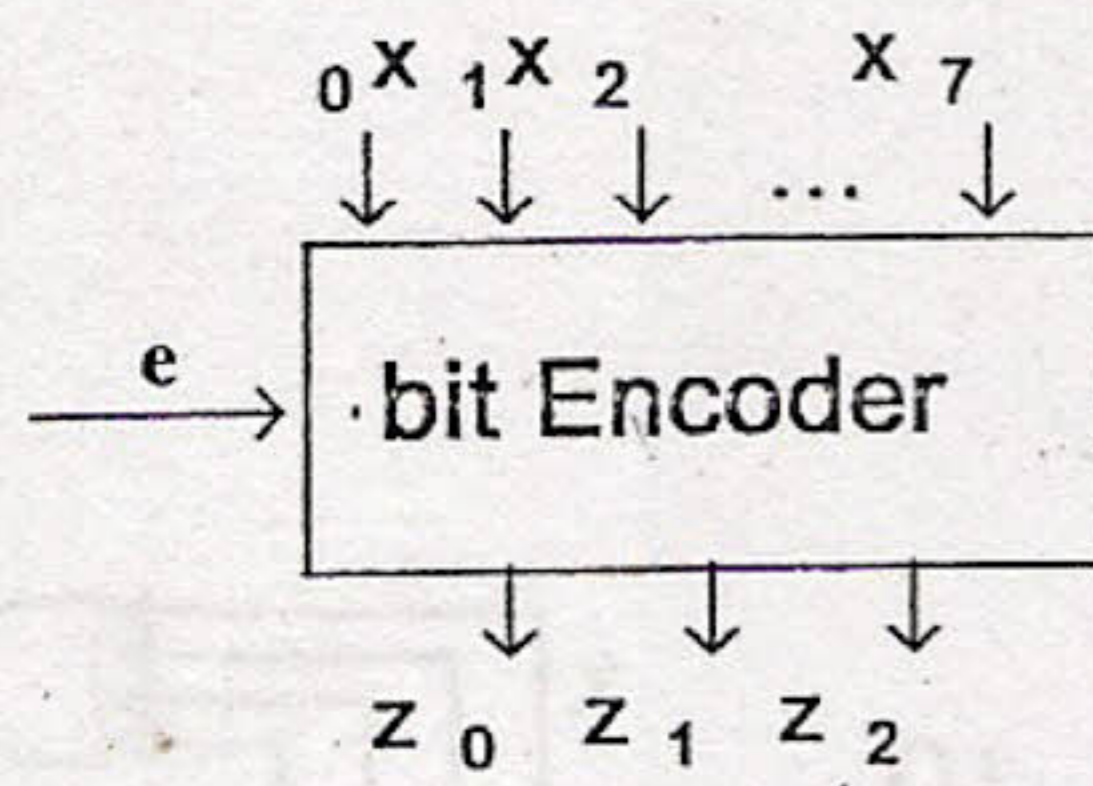
رمز گذار: (Encoder)

مدار ترکیبی است با 2^n خط ورودی و n خروجی که در آن خروجی ها آدرس خط ورودی فعال را تولید می کند.



مثال: مطلوب است طراحی 8-bit Encoder

$$2^n = 2^3 = 8 \Rightarrow n = 3$$

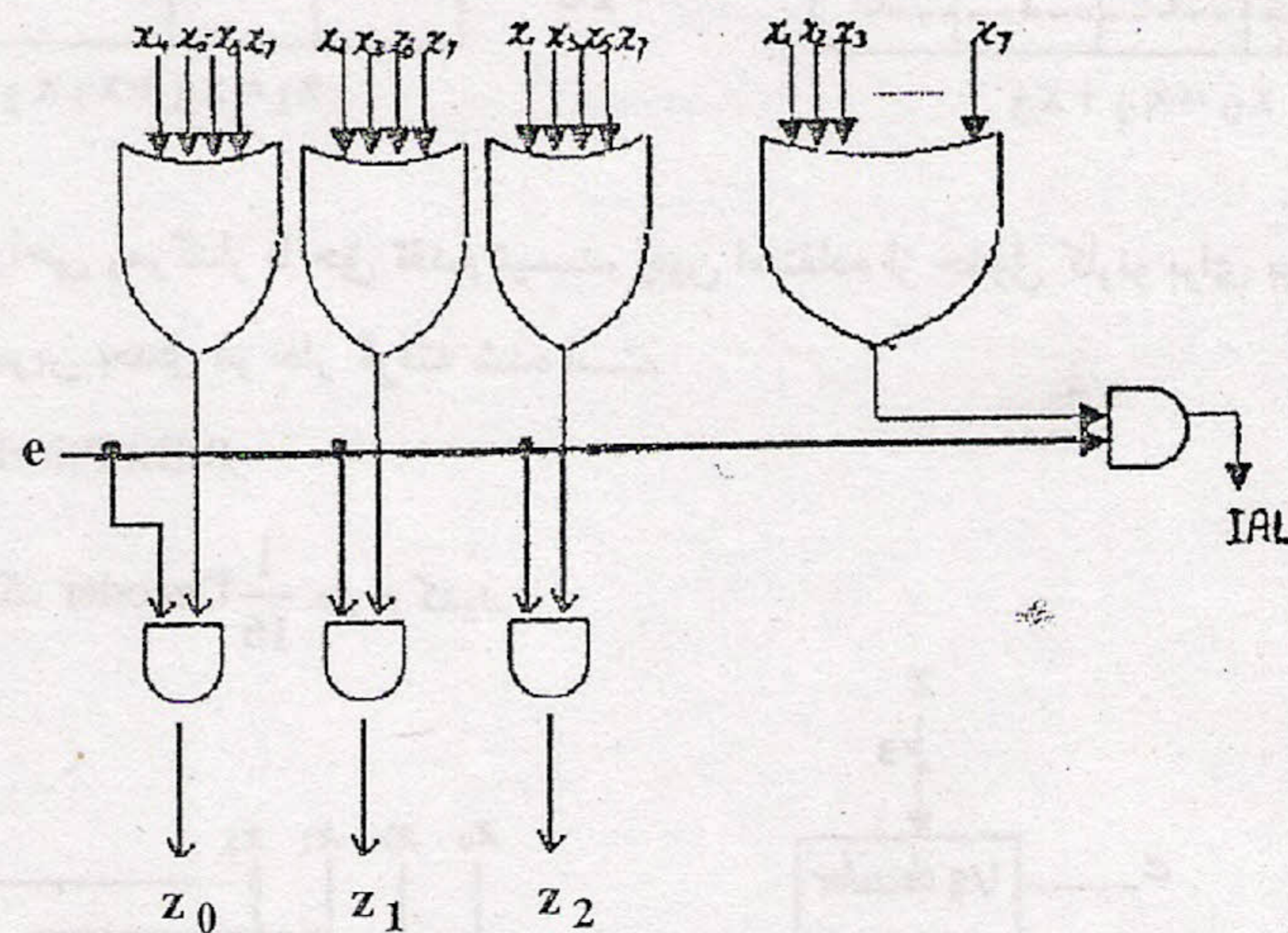


x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	z_0	z_1	z_2
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$$z_0 = x_4 + x_5 + x_6 + x_7$$

$$z_1 = x_2 + x_3 + x_6 + x_7$$

$$z_2 = x_1 + x_3 + x_5 + x_7$$



طراحی مدار به صورت فوق دو عیب اصلی دارد:

- مدار بین حالتی که فقط $x_0 = 1$ است و بقیه غیرفعال هستند و حالتی که هیچ کدام از ورودی‌ها فعال نباشد، نمی‌تواند تفاوت قایل شود. برای از بین بردن این عیب یک Input Active Line با استفاده از یک AND - Gate و یک OR - Gate به مدار اضافه می‌کنیم.
- در صورتی که بیش از یک خط ورودی به طور همزمان فعال باشند، مدار ممکن است آدرس نامعتبر تولید کند.

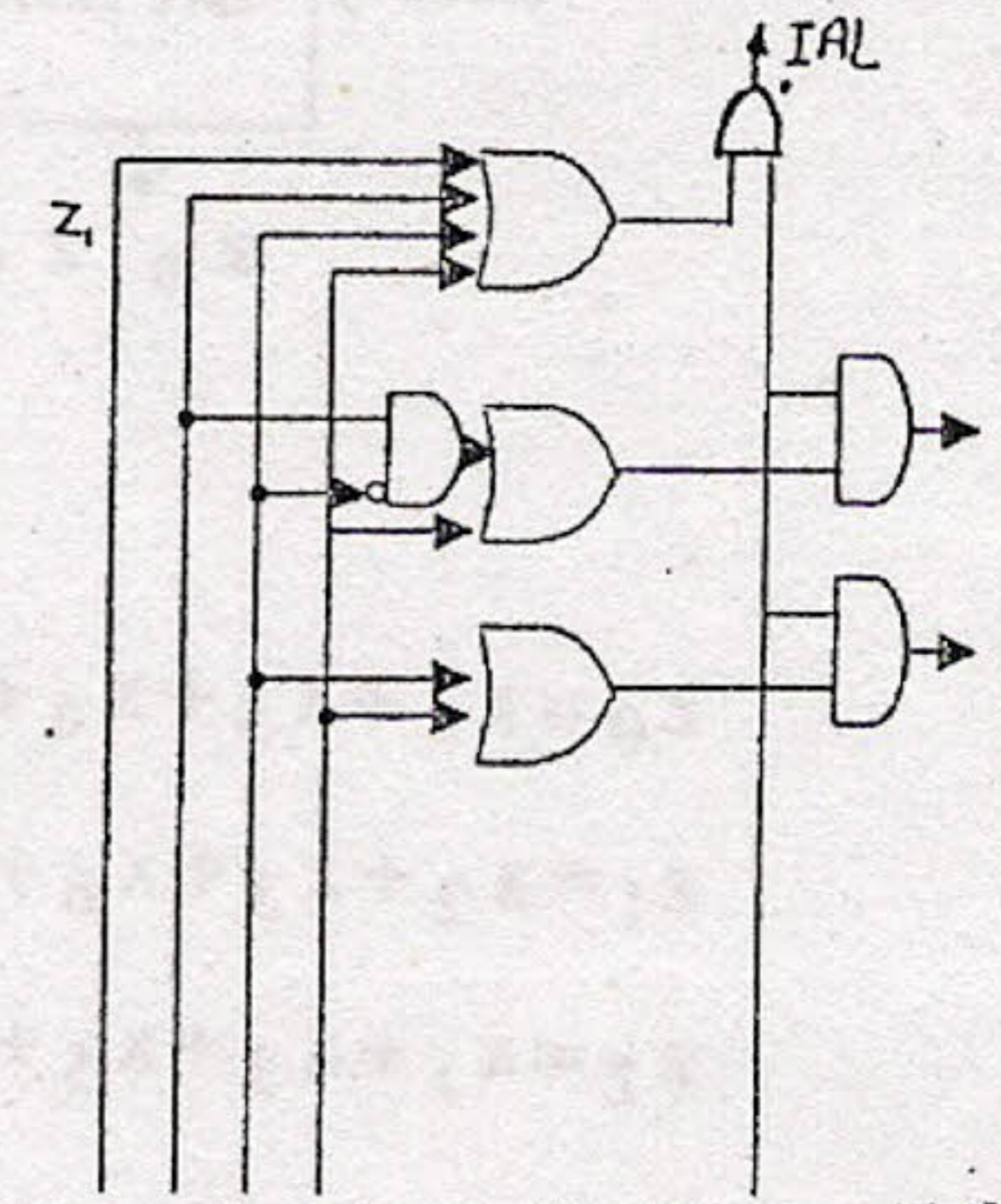
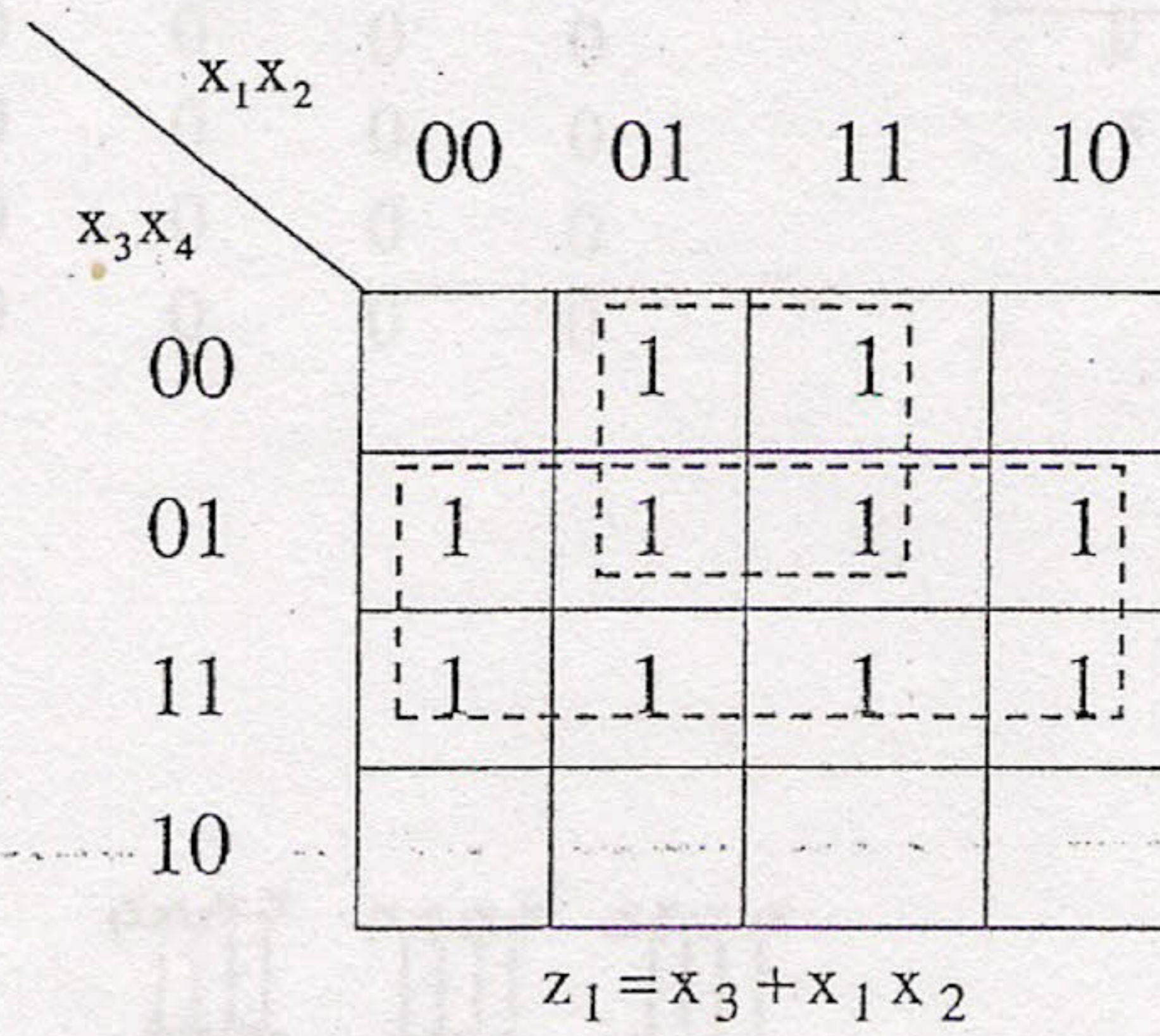
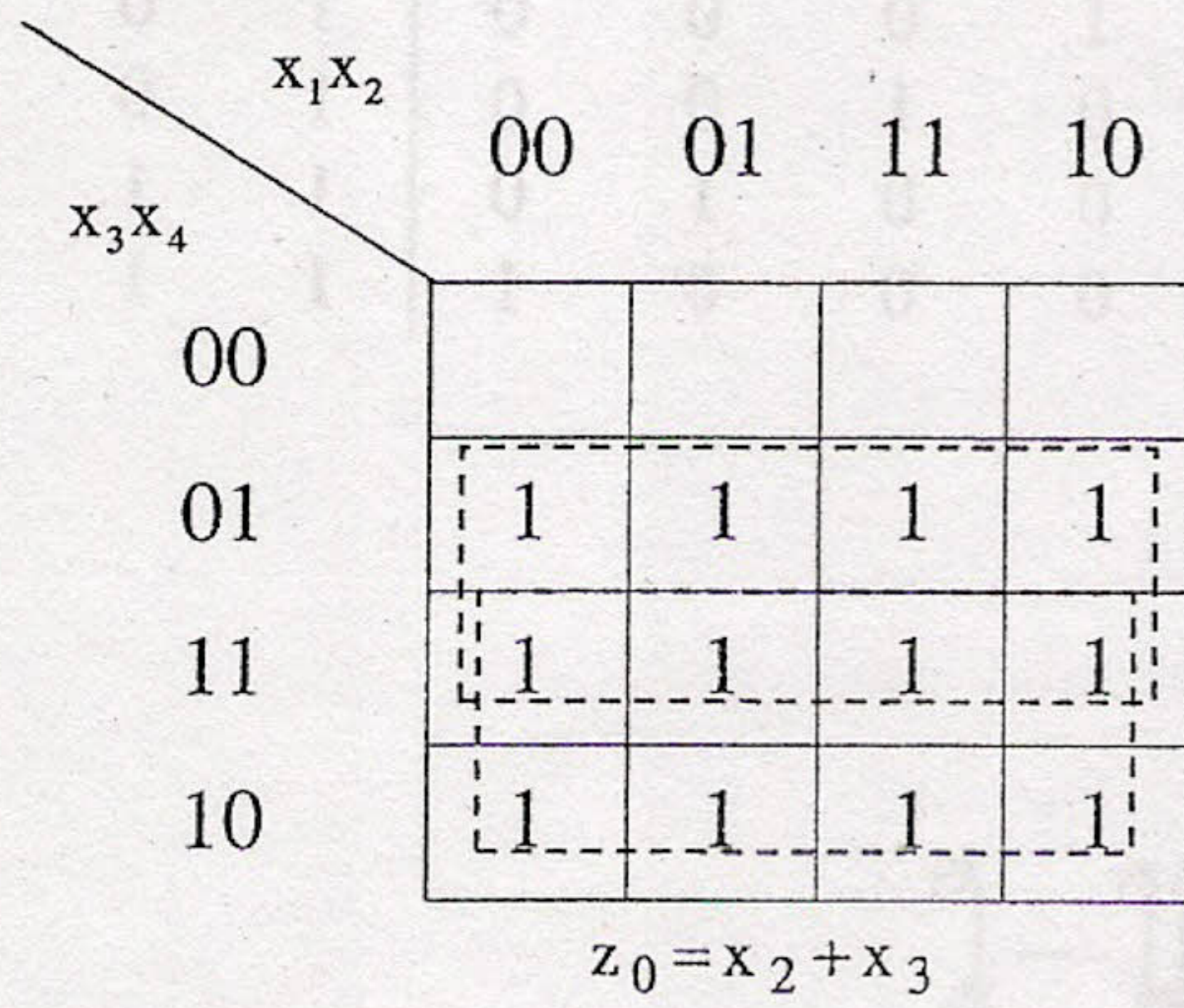
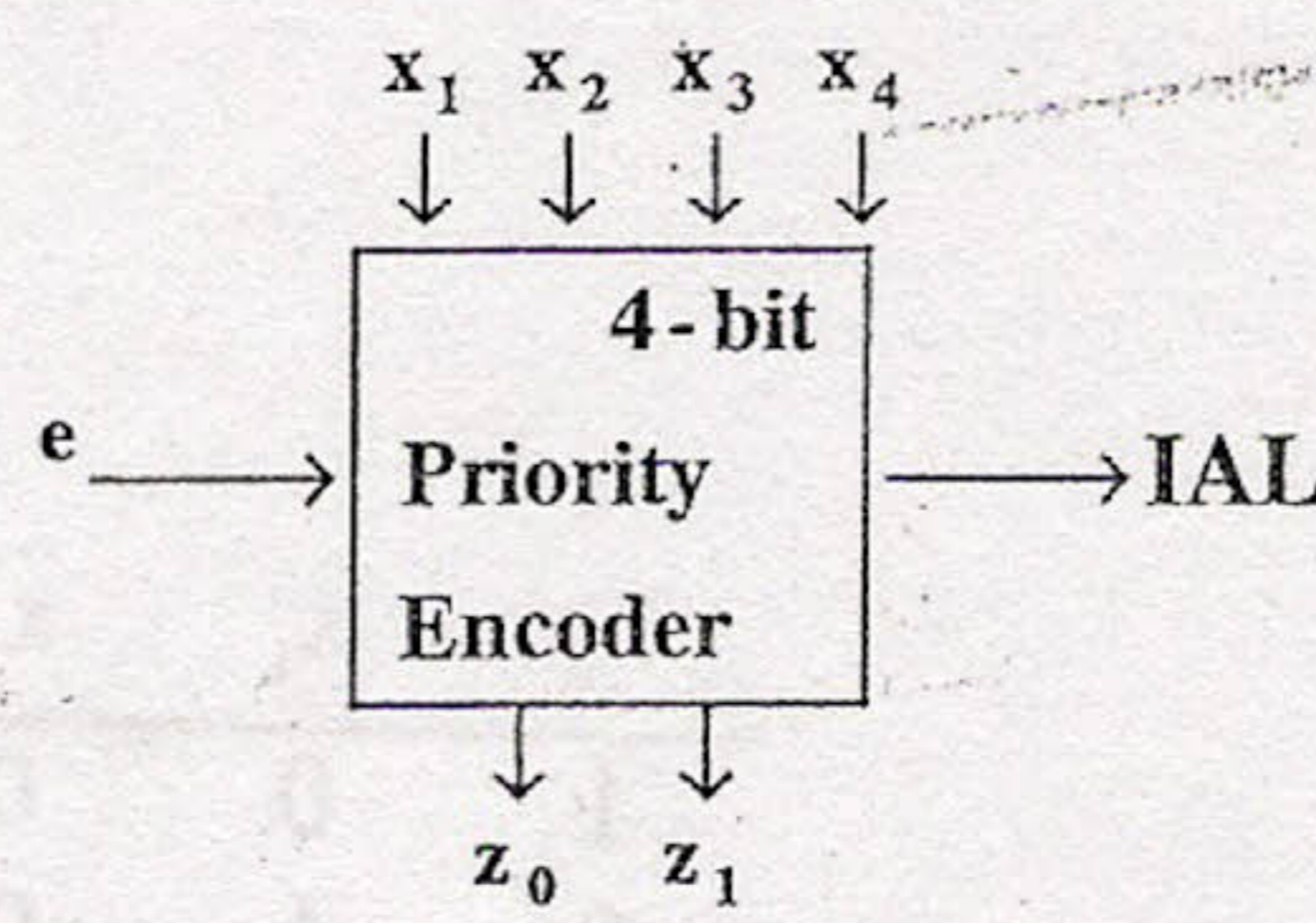
$$\left. \begin{matrix} x_1 = 1 \\ x_2 = 1 \end{matrix} \right\} \Rightarrow \text{آدرس نامعتبر که متناظر با خط ورودی فعال } x_3 \text{ می‌باشد فعال شود. } 011$$

برای رفع دومین عیب می‌توان از رمز گذار با حق تقدم (Priority Encoder) استفاده نمود که در آن اگر $x_i = x_j = 1$ یعنی x_i, x_j هر دو به طور همزمان فعال باشند و اندیس i بزرگتر از اندیس j باشد، آن گاه آدرس متناظر با x_i تولید می‌گردد.

مثال ۱:

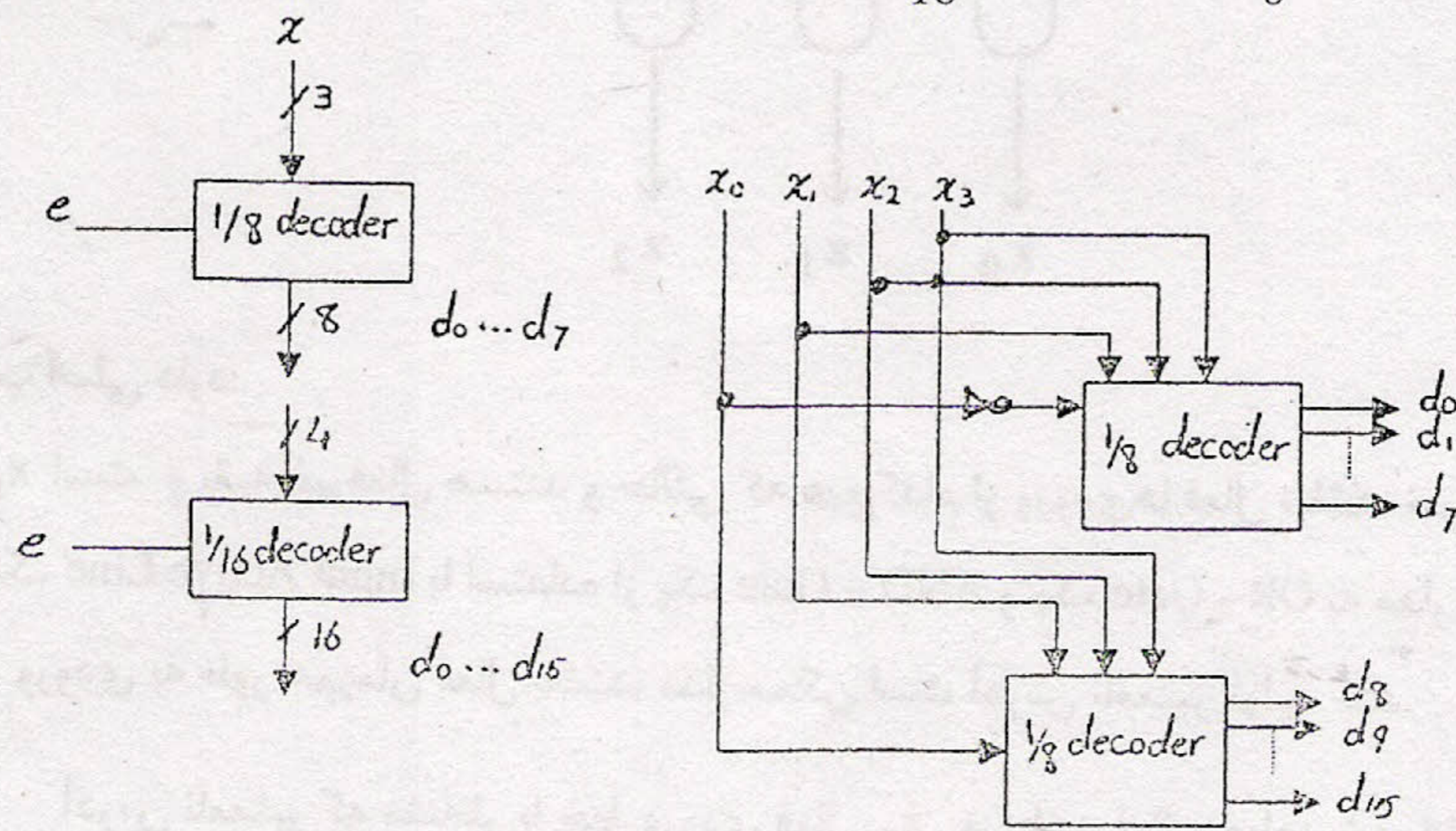
مطلوب است طراحی 4-Bit Priority Encoder

x_0	x_1	x_2	x_3	z_0	z_1
1	0	0	0	0	0
-	1	0	0	0	1
-	-	1	0	1	0
-	-	-	1	1	1



باید توجه نمود که این روش کلی برای طراحی رمزگذار با حق تقدم نیست، چون استفاده از جدول کارنو برای بیش از 6 متغیر ممکن نیست، روش کلی راه‌حلی است که برای تمرین بعدی در نظر گرفته شده است.
تمرین: مطلوب است طراحی 8-Bit Priority Encoder

مثال ۲: فقط با استفاده از $\frac{1}{8}$ Decoder، یک $\frac{1}{16}$ Decoder طرح کنید.



تمرین: با استفاده از نمودار بلوکی $\frac{1}{8}$ Decoder و $\frac{1}{4}$ Decoder، یک $\frac{1}{32}$ Decoder طرح کنید.

تمرین: با استفاده از نمودار بلوکی $\frac{1}{4}$ Decoder، یک $\frac{1}{16}$ Decoder طرح کنید.

تمرین: با استفاده از نمودار بلوکی $\frac{1}{4}$ Decoder، یک $\frac{1}{8}$ Decoder طرح کنید.

مثال ۳: با فرض Single Rail Logic یک 4-bit Magnitude Comparator طراحی کنید.

در صورتی که مبنای مقایسه bit یا بالاترین ارزش مکانی باشد، آن گاه بدون تشکیل جدول ارزش می توان مدار مقایسه کننده را طرح نمود.

$$A = A_3A_2A_1A_0$$

$$B = B_3B_2B_1B_0$$

متغیر بولی x_i را چنان تعریف می کنیم، اگر $A_i = B_i$ آن گاه $x_i = 1$ و اگر $A_i \neq B_i$ آن گاه $x_i = 0$ باشد. یعنی:

$$x_i = A_iB_i + \bar{A}_i\bar{B}_i = A_i \oplus B_i$$

$$(A = B) = x_3x_2x_1x_0$$

$$(A < B) = \bar{A}_3B_3 + x_3\bar{A}_2B_2 + x_3x_2\bar{A}_1B_1 + x_3x_2x_1\bar{A}_0B_0$$

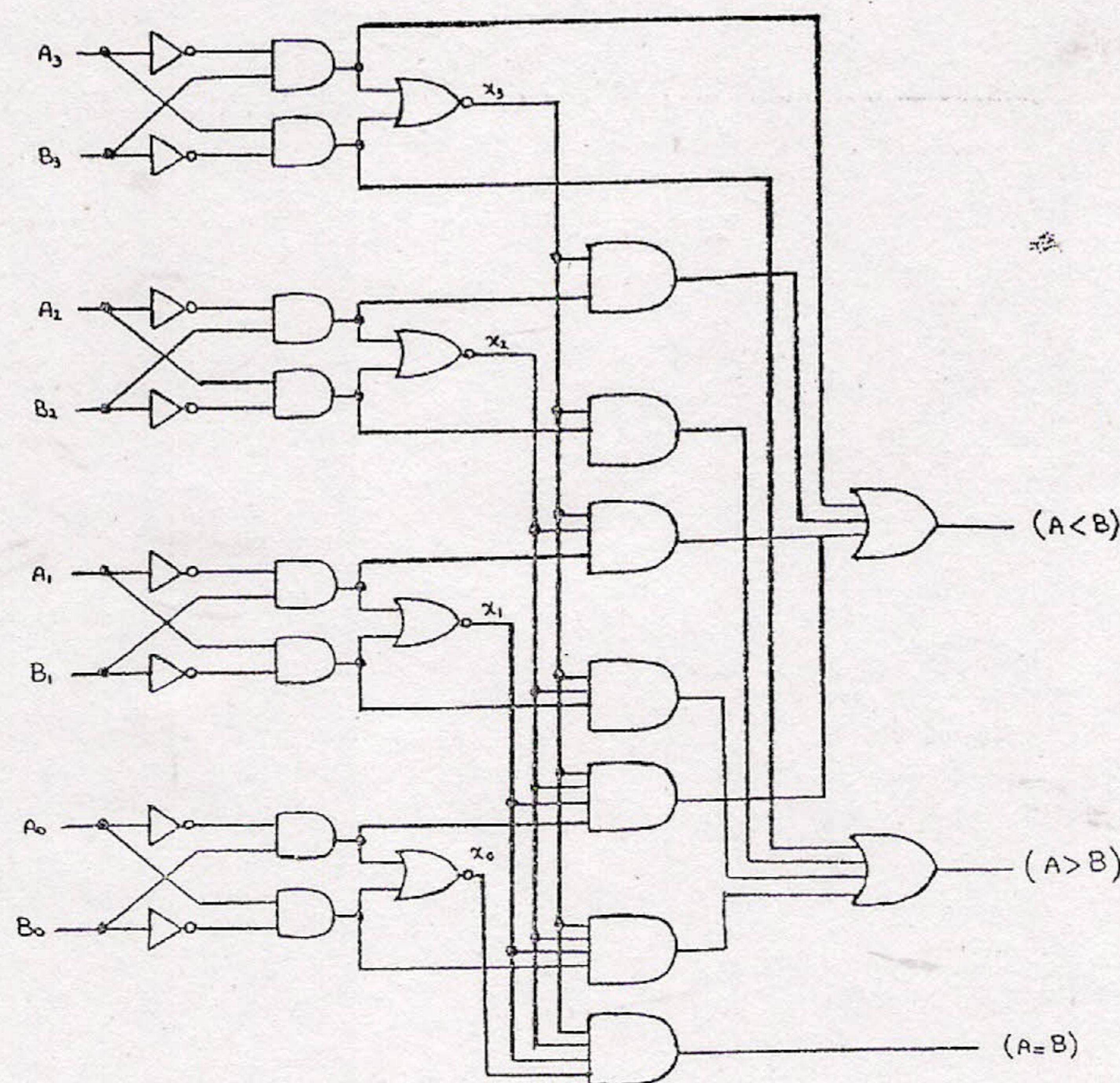
$$(A > B) = \bar{A}_3B_3 + x_3A_2\bar{B}_2 + x_3x_2A_1\bar{B}_1 + x_3x_2x_1A_0\bar{B}_0$$

می توان از x_i را از "Exclusive Nor" استفاده کرد و به دست آورد.

$$x_i = A_iB_i + \bar{A}_i\bar{B}_i = A_i \oplus B_i$$

اثبات:

$$\overline{A_i \oplus B_i} = \overline{(A_iB_i)(\bar{A}_i\bar{B}_i)} = (A_i + \bar{B}_i)(\bar{A}_i + B_i) = (0 + A_iB_i) + (\bar{B}_i\bar{A}_i + 0) = A_iB_i + \bar{B}_i\bar{A}_i$$



تمرین: برای تابع f که به وسیله جدول کارنو زیر تعریف شده است.

(A) کلیه PI تابع را به دست آورید.

(B) کلیه PI های ضروری را به دست آورید.

(C) فرم دو سطحی AND - OR تابع را به دست آورید.

(D) فرم دو سطحی OR - AND تابع را به دست آورید.

(E) فرم تمام NAND تابع را به دست آورید.
 (F) فرم تمام NAND تابع را به دست آورید.

X_1X_2	00	01	11	10
X_3X_4			1	-
00			1	-
01				1
11	-	-	-	1
10	1		1	

تمرین: اطلاعات زیر در مورد تابع 4 متغیره در دست است. تابع را با استفاده از جدول کارنو ساده کنید.

$$f(0, x_2, x_3, x_4) = \bar{x}_2 + x_3$$

$$f(1, 0, 1, x_4) = \bar{x}_4$$

$$f(1, x_2, 0, x_1) = \bar{x}_3$$

$$f(x_1, x_2, 1, 0) = 1$$

