

فصل دوم

طراحی مدارهای ترکیبی MSI و کاربرد آن‌ها

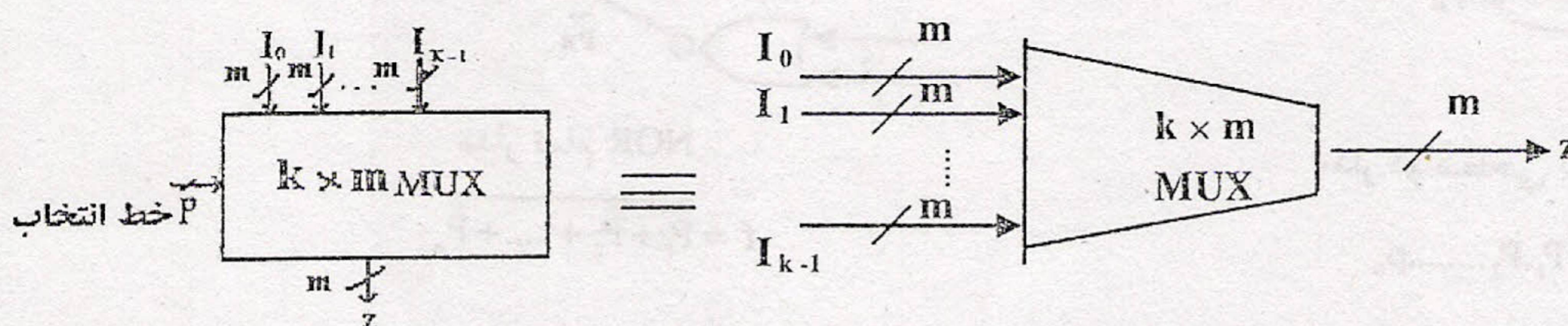
مدارهای مجتمع: *Intergated circuits – chips (IC)*

طراحی مدارهای تمام NOR و تمام NAND باعث پیدایش IC‌ها 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 – 32 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 پیاده سازی نمود. VLSI: حداکثر شامل Gate – 10000 می‌باشد و می‌توان یک Microprocessor را در آن جای داد. VVLSI (USI): حداکثر شامل Gate – 100000 می‌باشد. در این درس تعدادی از MSI‌ها طرح می‌کنیم که در درس معماری کامپیوتر کاربرد دارد.

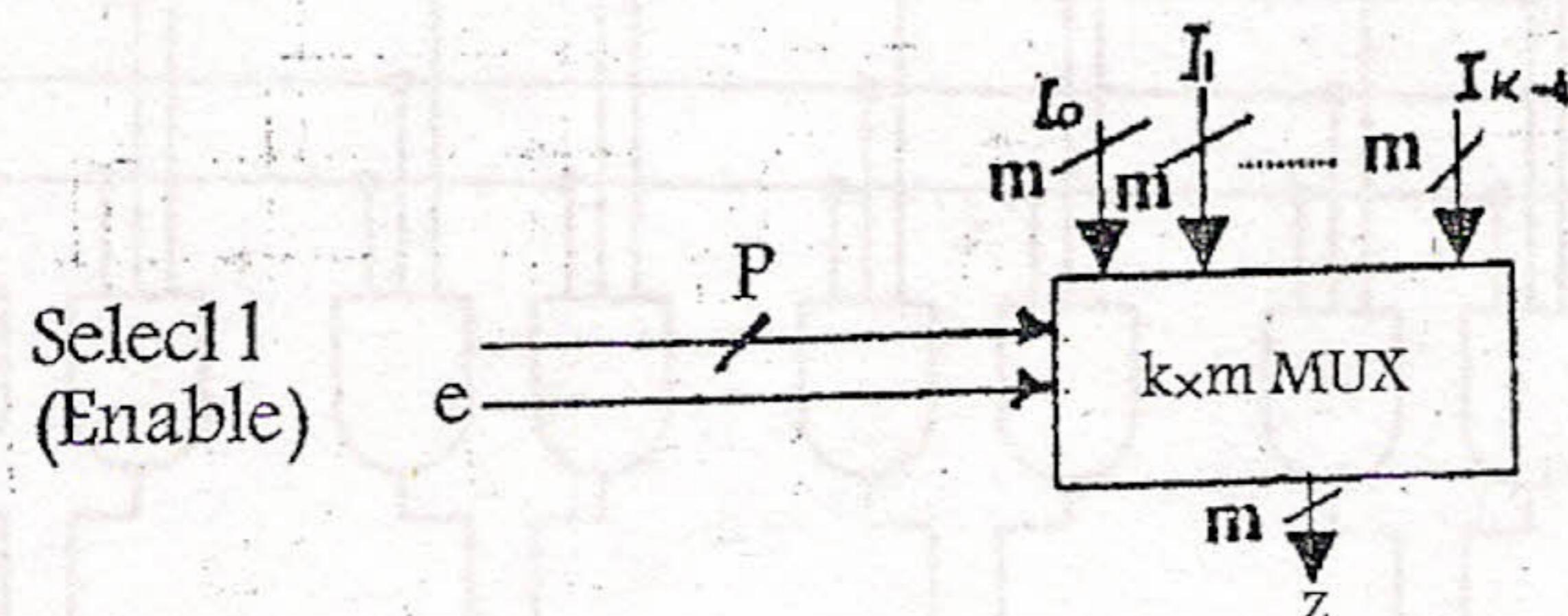
تسهیم کننده: *(MUX) Multiplexer*

تسهیم کننده مدار ترکیبی است که از بین چندین منبع ورودی متمايز 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 $(K \times m)$ MUX (Kinput-m bit MUX) می‌گویند.



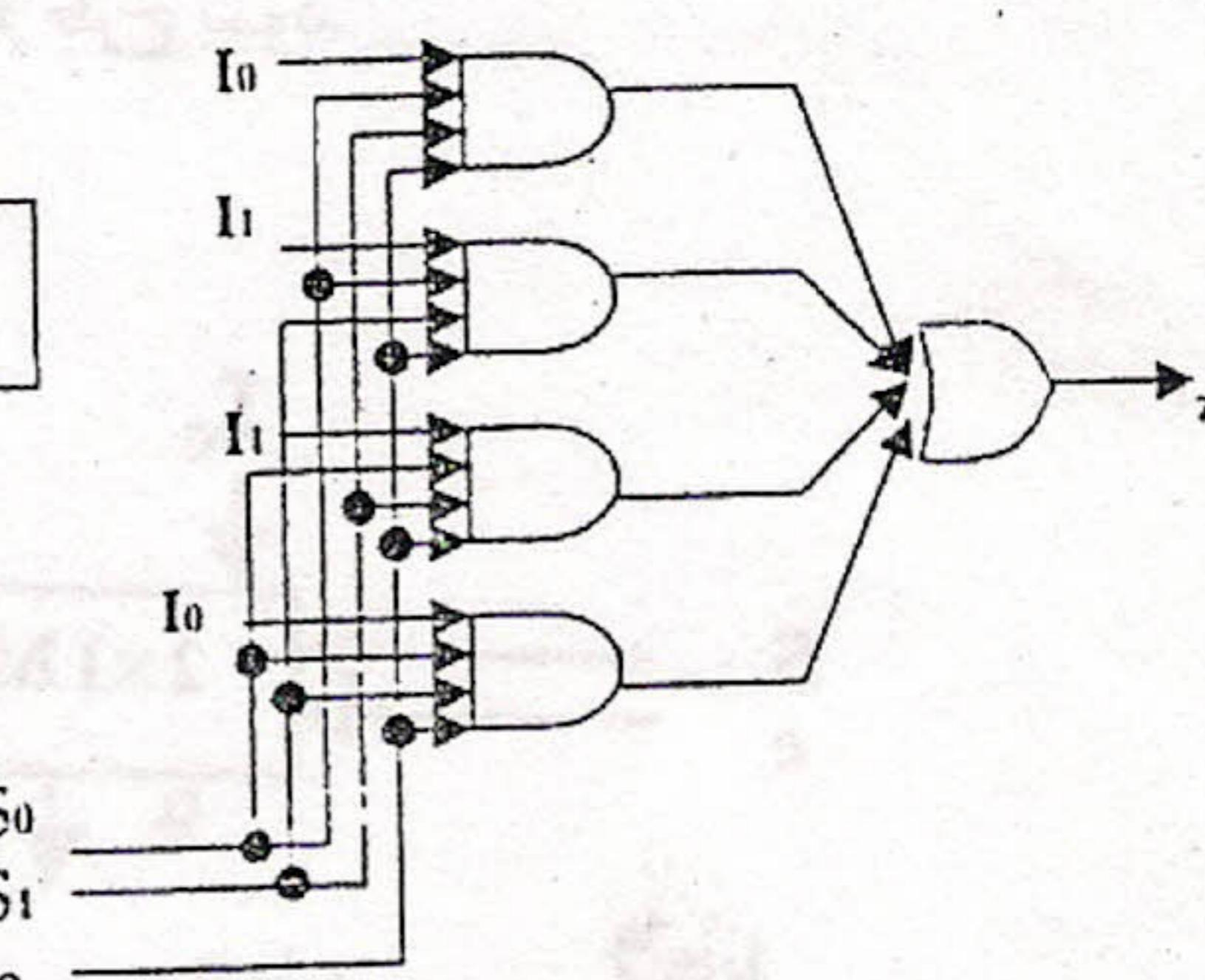
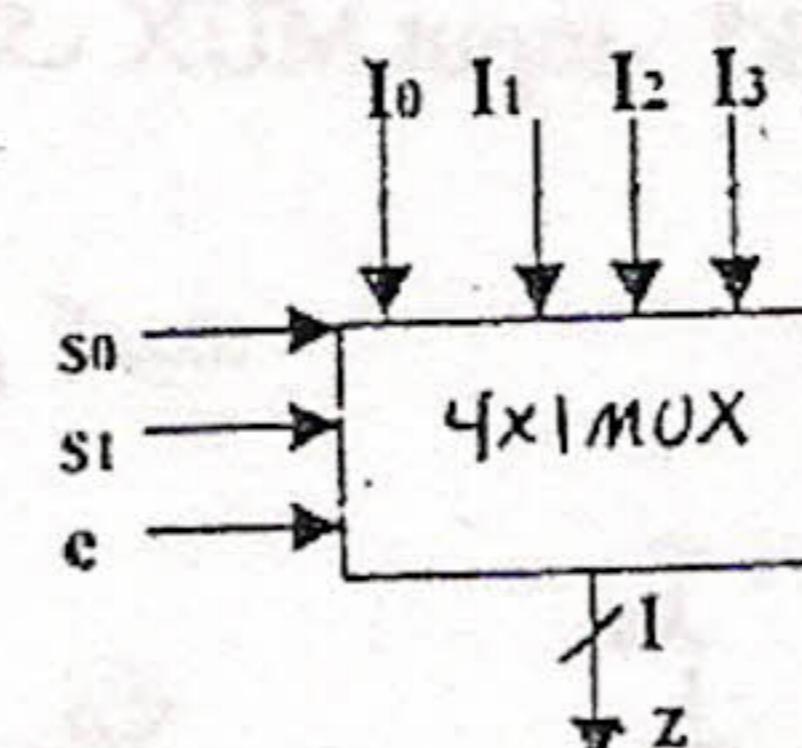
توجه: اگر $e=1$ باشد، آن‌گاه مدار به حالت عادی کار می‌کند.

اگر $e=0$ باشد، مدار کار نخواهد کرد (خاموش خواهد بود).

مثال: مطلوب است 4×1 MUX

$$\begin{aligned} k = 2^P \\ 4 = 2^2 \end{aligned} \Rightarrow P = 2$$

S_1	S_2	Z
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



کاربردهای MUX

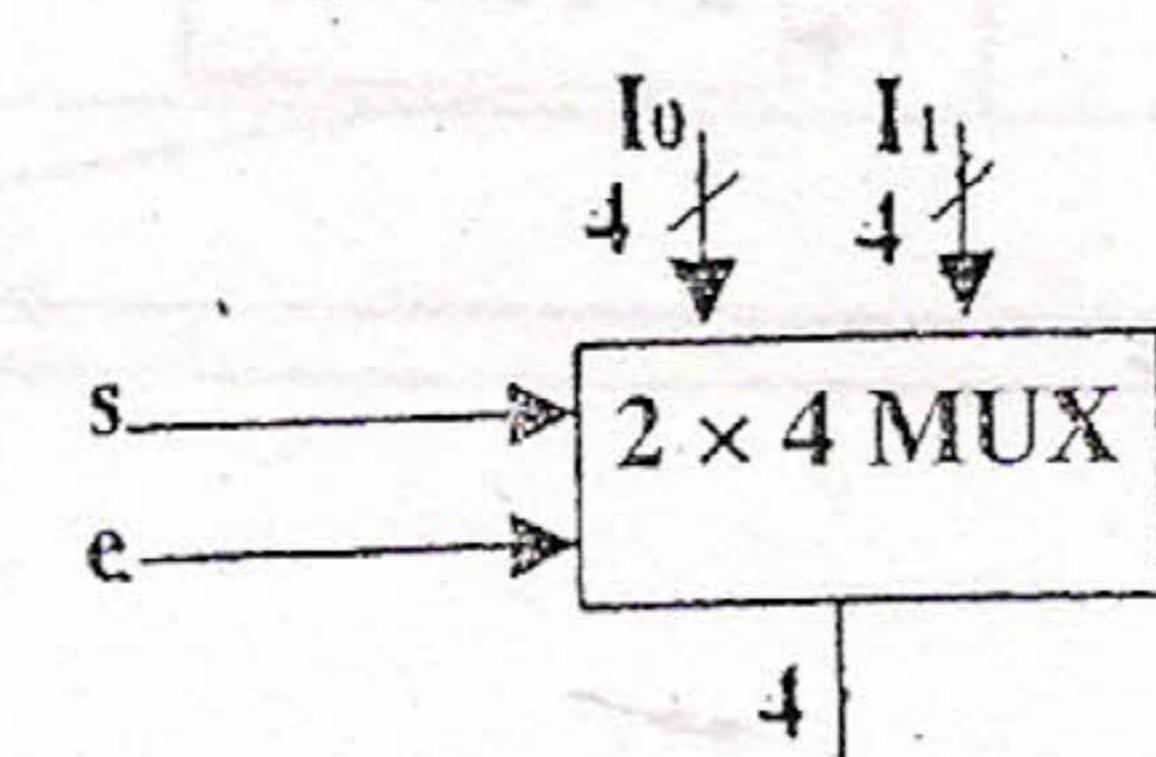
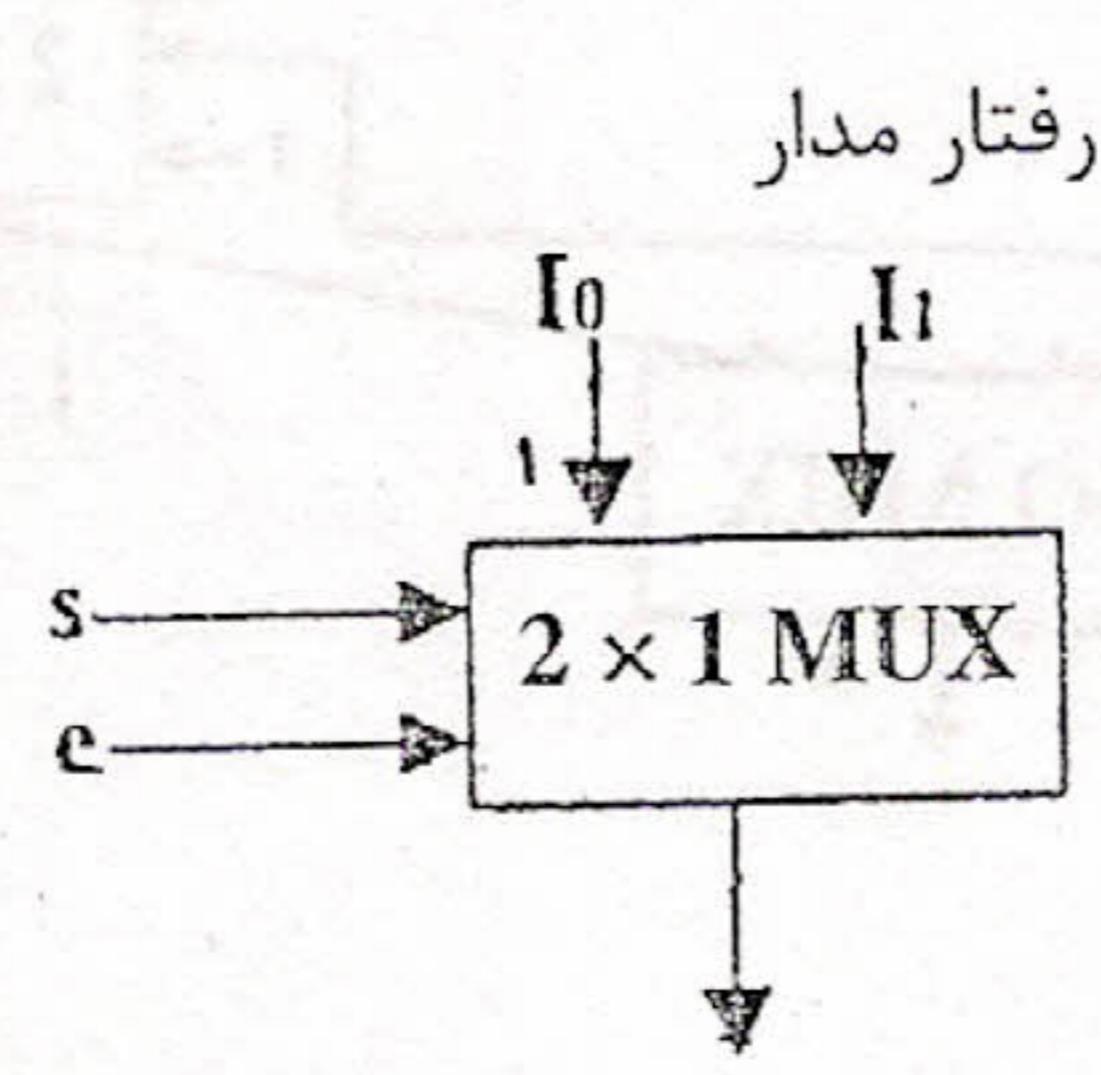
۱) برای هدایت اطلاعات از بین چند منبع متمایز به سوی یک مقصد مشترک.

۲) برای طراحی سیستم BUS (گذرگاه یا معتبر) مطالعه این قسمت در درس معماری کامپیوتر خواهد بود.

۳) با اتصال شبکه‌ای MUX ها می‌توان تعداد بیت‌های منابع ورودی را افزایش داد.

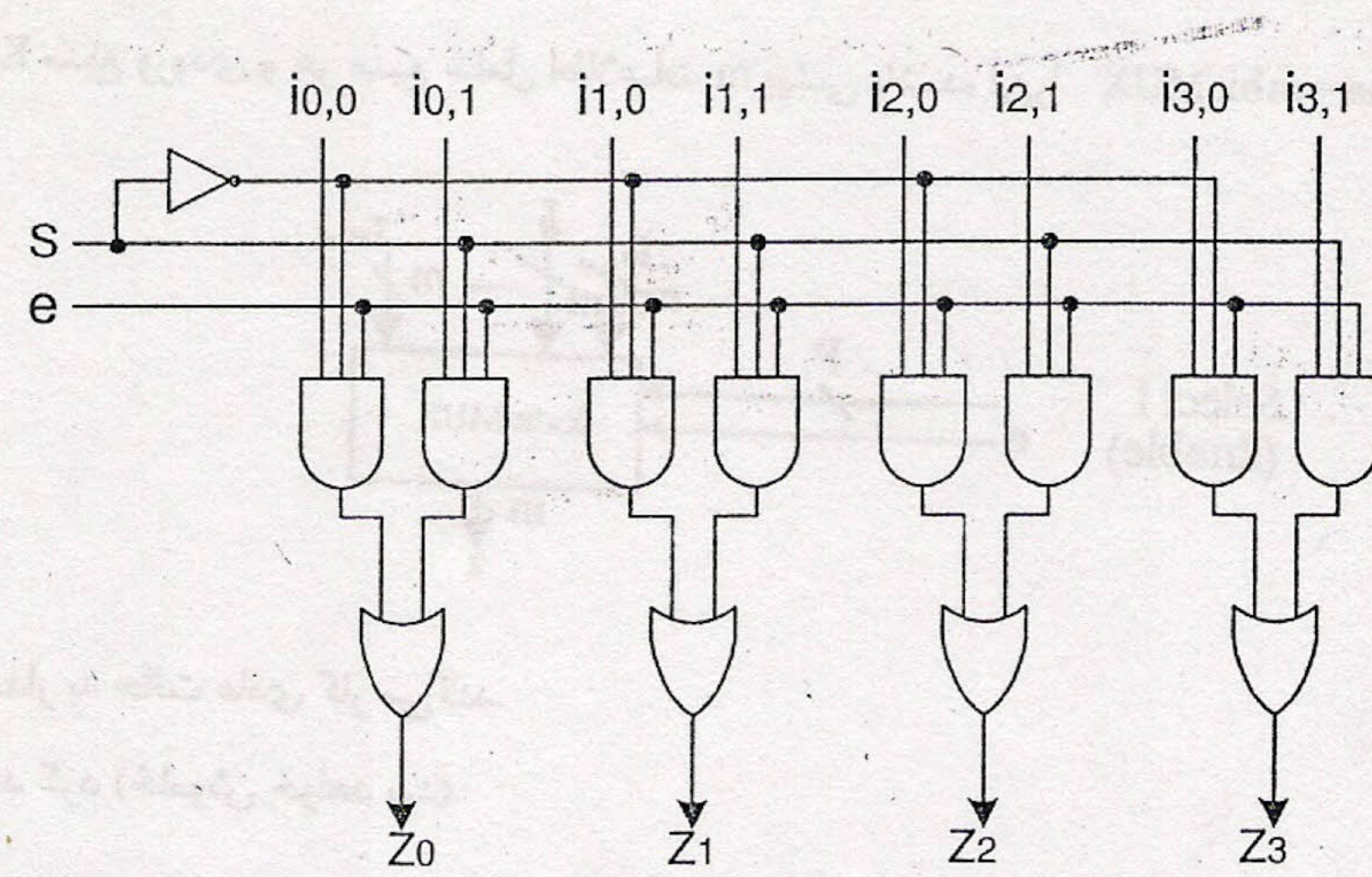
مثال: با استفاده از یک 2×1 Mux یک 2×4 Mux طراحی کنید.

e	s	Z
0	-	0
1	0	I_0
1	1	I_1



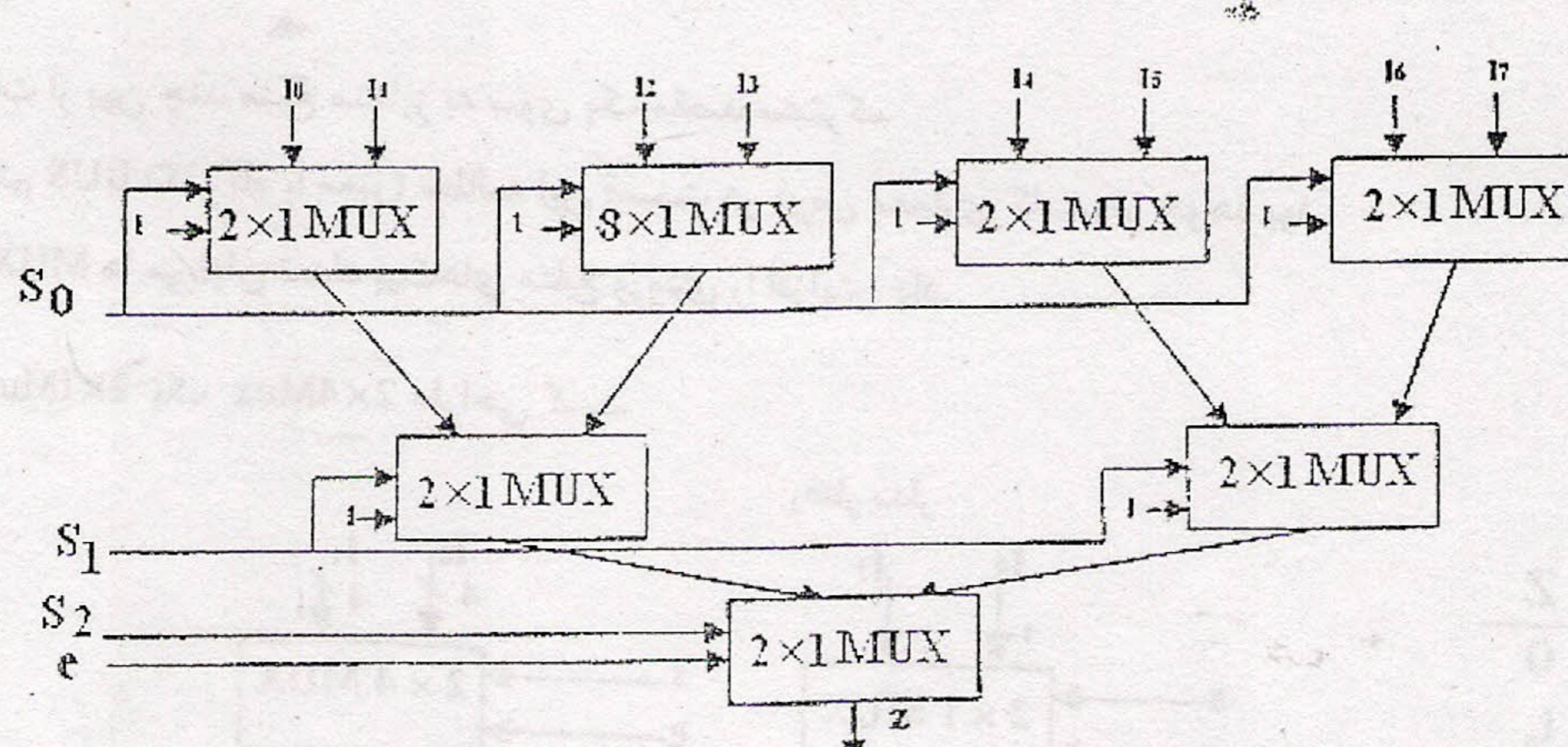
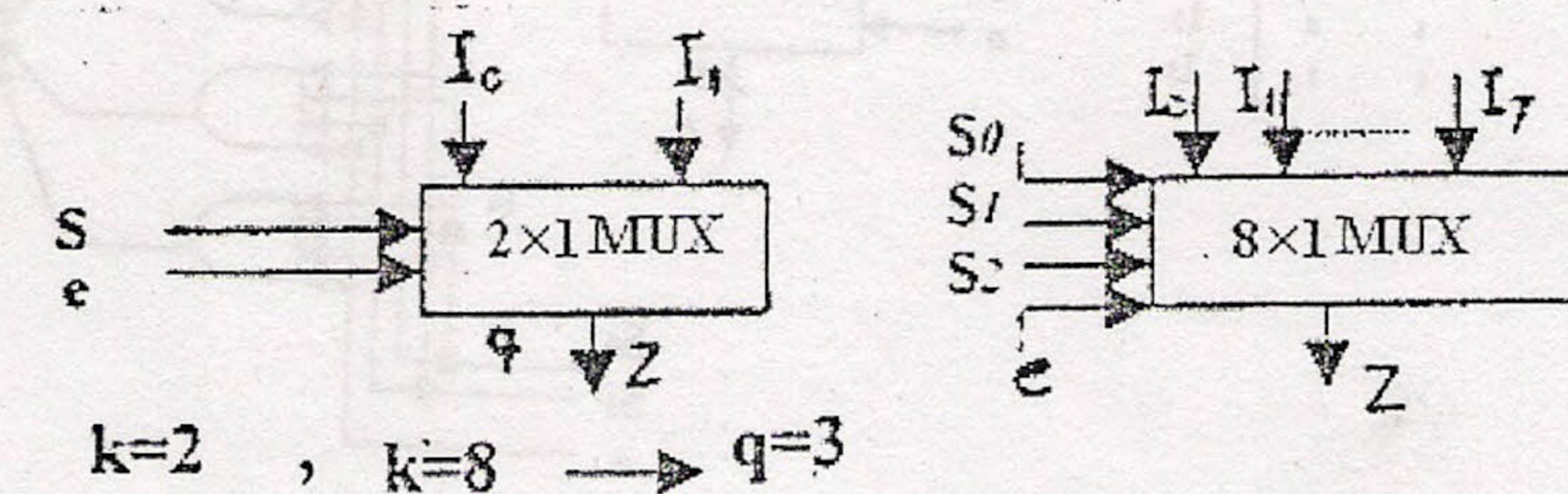
$$I_0 = i_{0,0} i_{0,1} i_{0,2} i_{0,3}$$

$$I_1 = i_{1,0} i_{1,1} i_{1,2} i_{1,3}$$



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

مثال: با استفاده از 2×1 MUX یک 8×1 MUX طراحی کنید.



تمرین: با استفاده از 2×1 MUX یک 8×4 MUX طراحی کنید.

۵. هر تابع ترکیبی را می‌توان به وسیله MUX انجام داد.

روش a: هر تابع ترکیبی n متغیری را می‌توان به وسیله یک $2^n \times 1$ MUX انجام داد، مشروط بر این که n متغیر تابع به عنوان n خط

انتخاب (Select) MUX به کار روند. این روش پیاده‌سازی را MUX نوع ۰-۱ نامند، زیرا از صفر متغیر در منابع ورودی استفاده می‌شود.

روش b: هرتابع ترکیبی n متغیر را می‌توان به وسیله $2^{n-1} \times 1$ MUX انجام داد، مشروط بر این که $(n-1)$ متغیر تابع به عنوان $(n-1)$ خط انتخاب وصل شود و از متغیر باقی‌مانده در ورودی MUX استفاده بشود. این روش پیاده‌سازی را MUX نوع ۱ می‌نامند زیرا فقط از یک متغیر در منابع ورودی استفاده می‌شود.

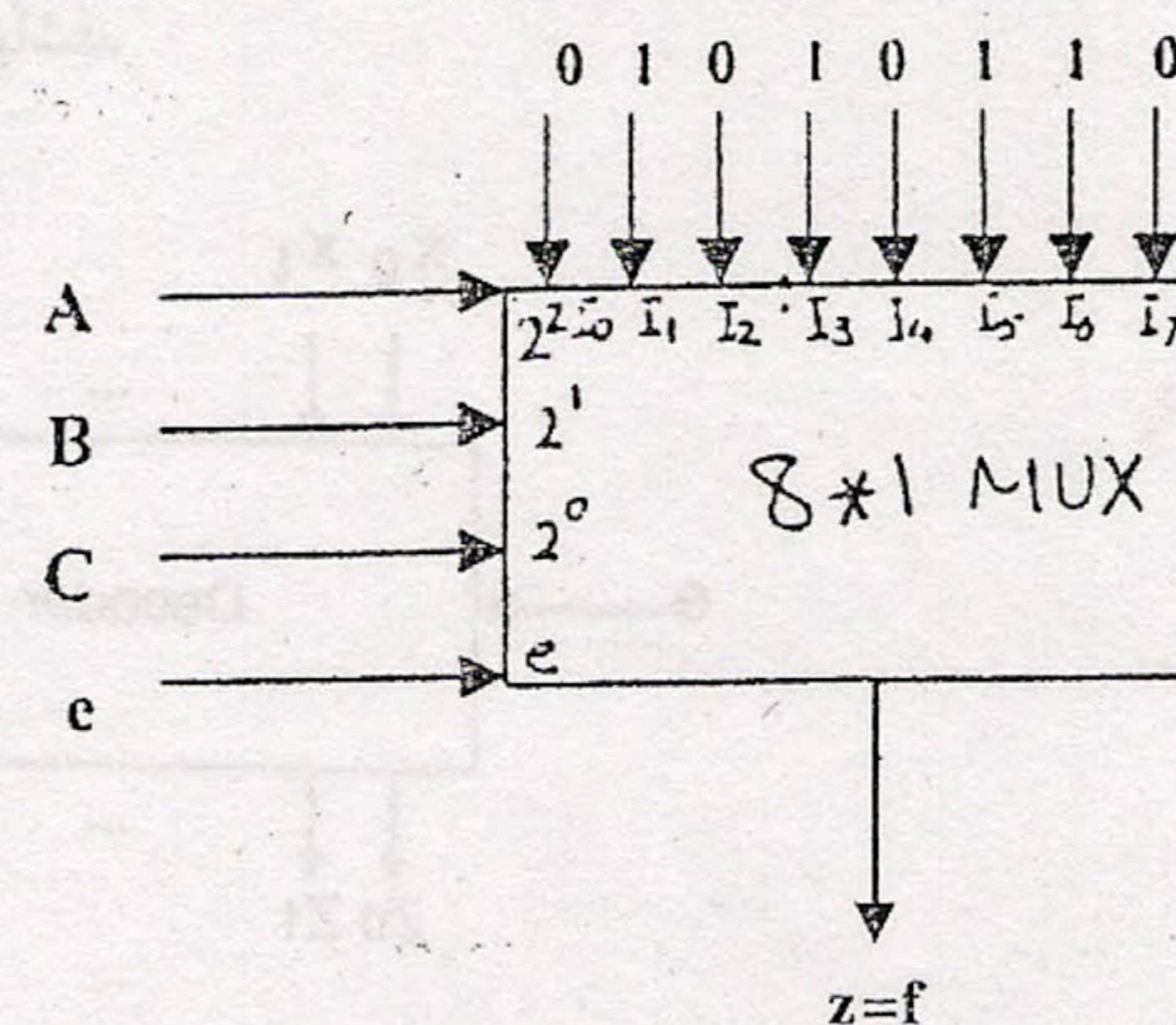
مثال: تابع زیر را به وسیله MUX انجام دهید.

$$f = (A, B, C) = \sum_1^1 (1, 3, 5, 6)$$

$$n = 3$$

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

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

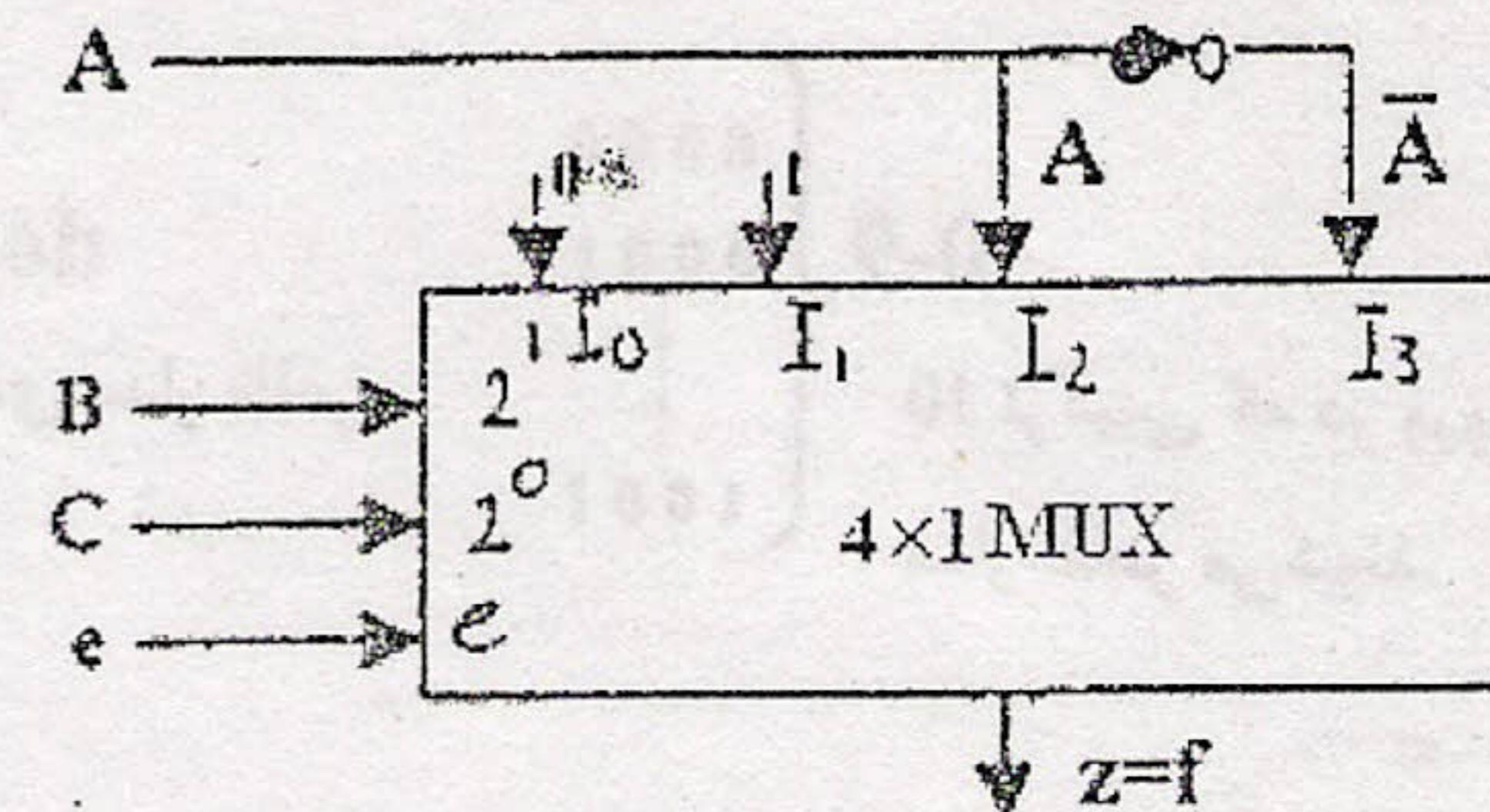


روش b

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

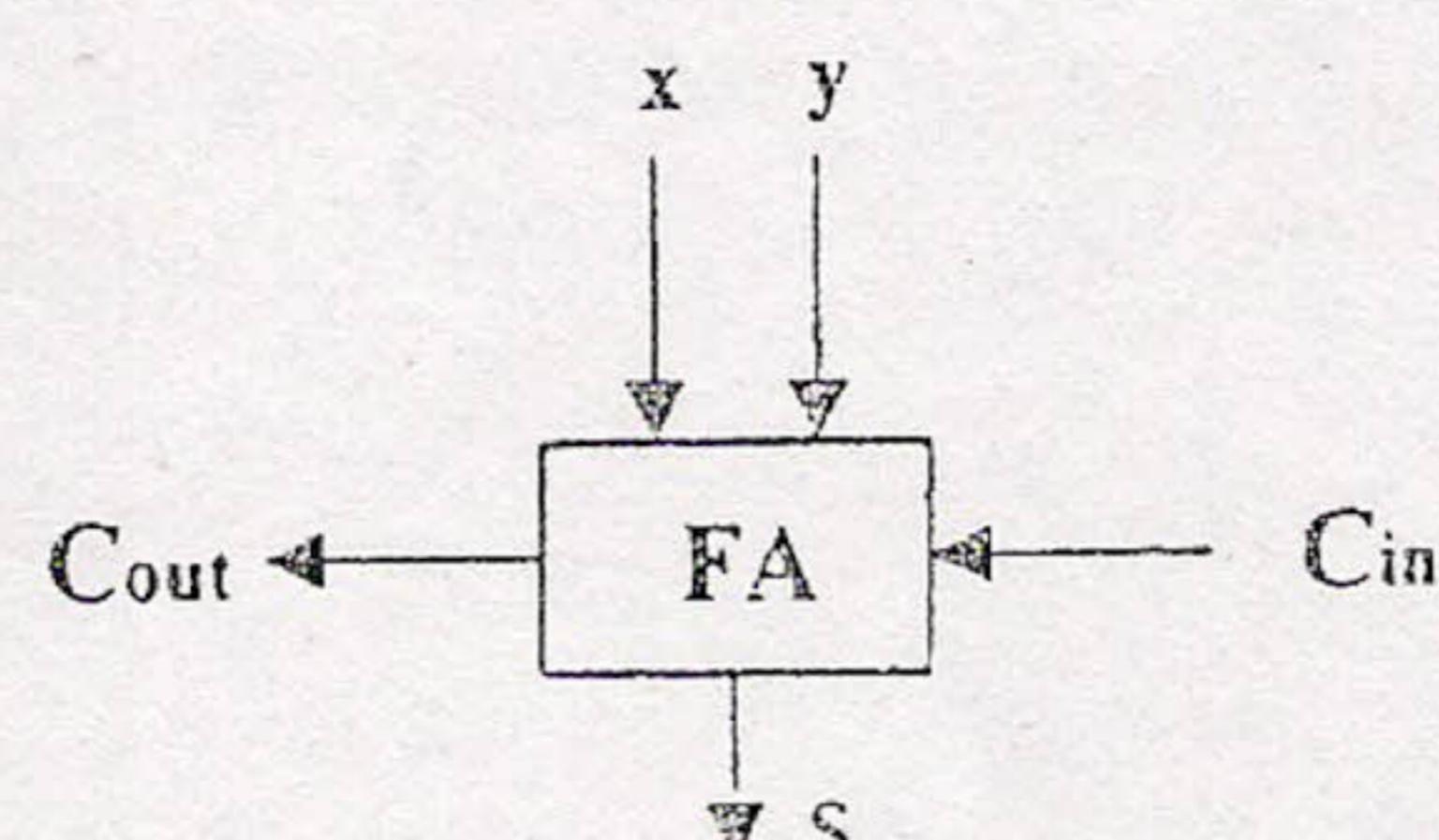
	I ₀	I ₁	I ₂	I ₃
\bar{A}	0	①	2	③
A	4	⑤	⑥	7

0	1	A	\bar{A}
---	---	---	-----------



در روش b ستون‌های جدول را با منابع ورودی MUX و سطرهای جدول را با مکمل متغیر و خود متغیر باقی‌مانده نام‌گذاری می‌کنیم. در سطری که با مکمل متغیر نام‌گذاری شده است، اندیس دهدۀ مین‌ترم‌هایی را می‌نویسیم که متغیر باقی‌مانده در آن صفر و در سطری که با خود متغیر نام‌گذاری شده است، اندیس دهدۀ مین‌ترم‌هایی را می‌نویسیم که متغیر باقی‌مانده در آن یک می‌باشد. سپس دور مین‌ترم‌هایی که مقدار تابع به ازای آن یک می‌باشد، دایره رسم می‌کنیم. اگر ستونی اصلاً 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 می‌کنیم که دایره در آن وجود دارد.

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

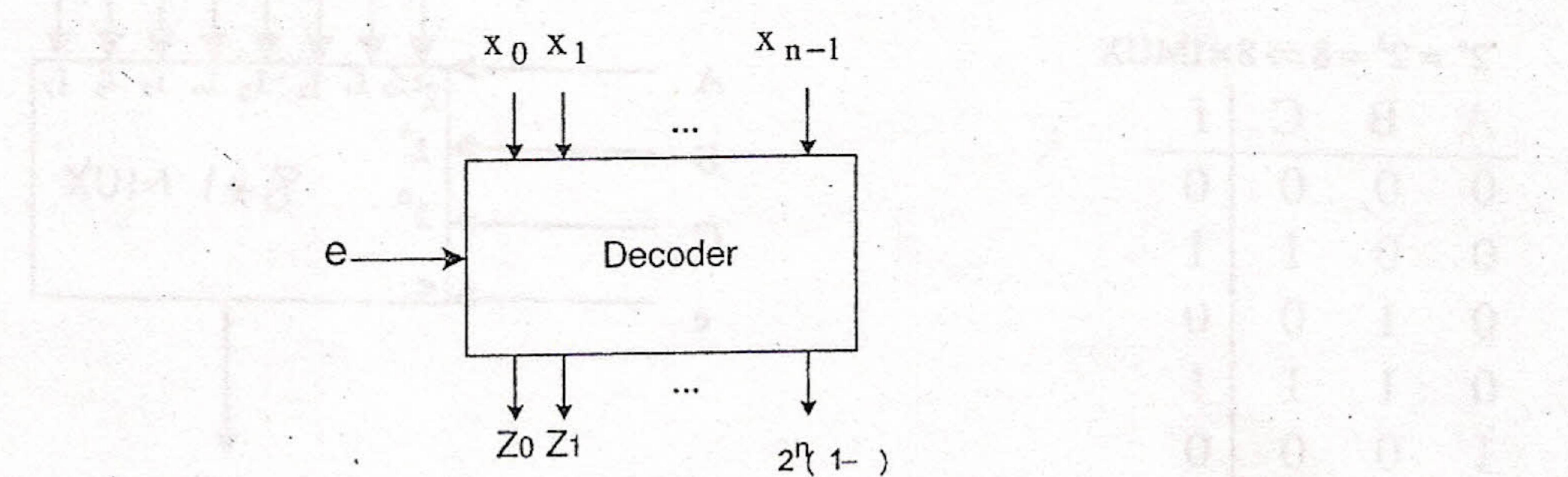


تمرین - با اعمال قضیه شانون به متغیرهای a, b, c تابع $f(a, b, c) = a + b \cdot \bar{c}$ را به وسیله MUX انجام دهید.

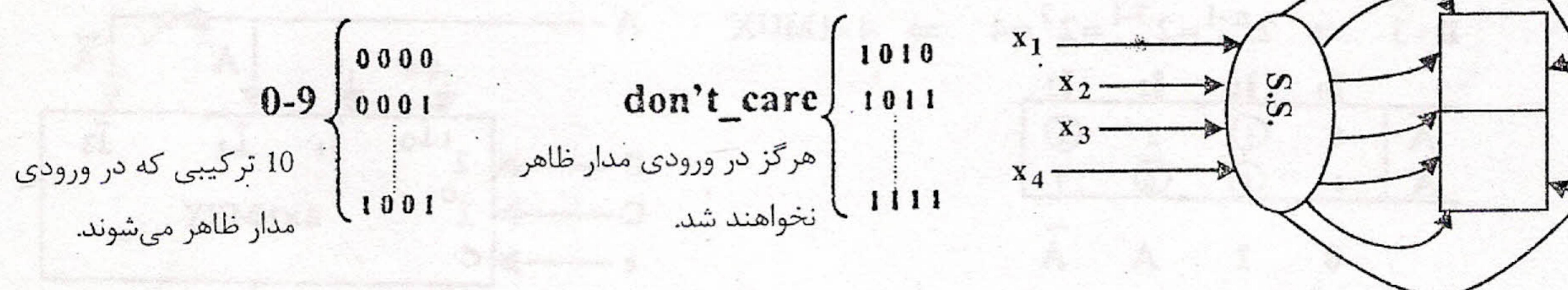
تمرین - با استفاده از جدول ارزش تابع $f(a, b, c, d) = \sum_1(4, 5, 6, 7, 10, 14)$ را به وسیله MUX های نوع 0 تا 3 طرح کنید.

مدار مجتمع رمزگشای Decoder:

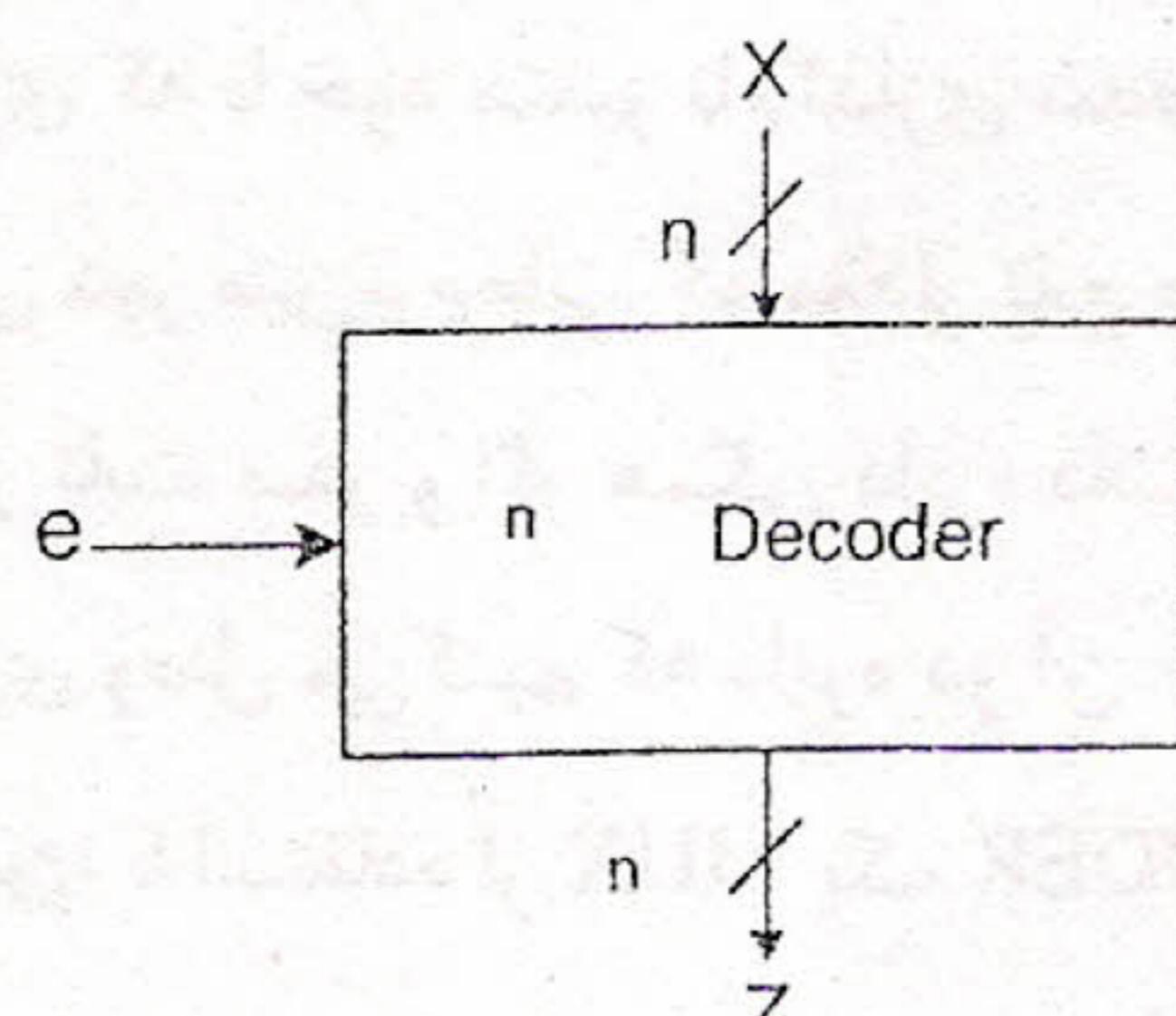
مدار ترکیبی است با n خط ورودی و 2^n خط خروجی که در آن به ازا هر یک از ترکیبات ورودی حداکثر 2^n خط خروجی فعال می‌شود. در واقع هر خروجی متناظر با یک Minterm می‌باشد.



در صورتی که در ورودی ترکیبات don't-care وجود داشته باشد، آن‌گاه تعداد خروجی‌ها کمتر از 2^n خواهد بود. مثلاً $2^4 < m < 2^5$ مانند که در آن تعداد خروجی‌ها 7 می‌باشد.



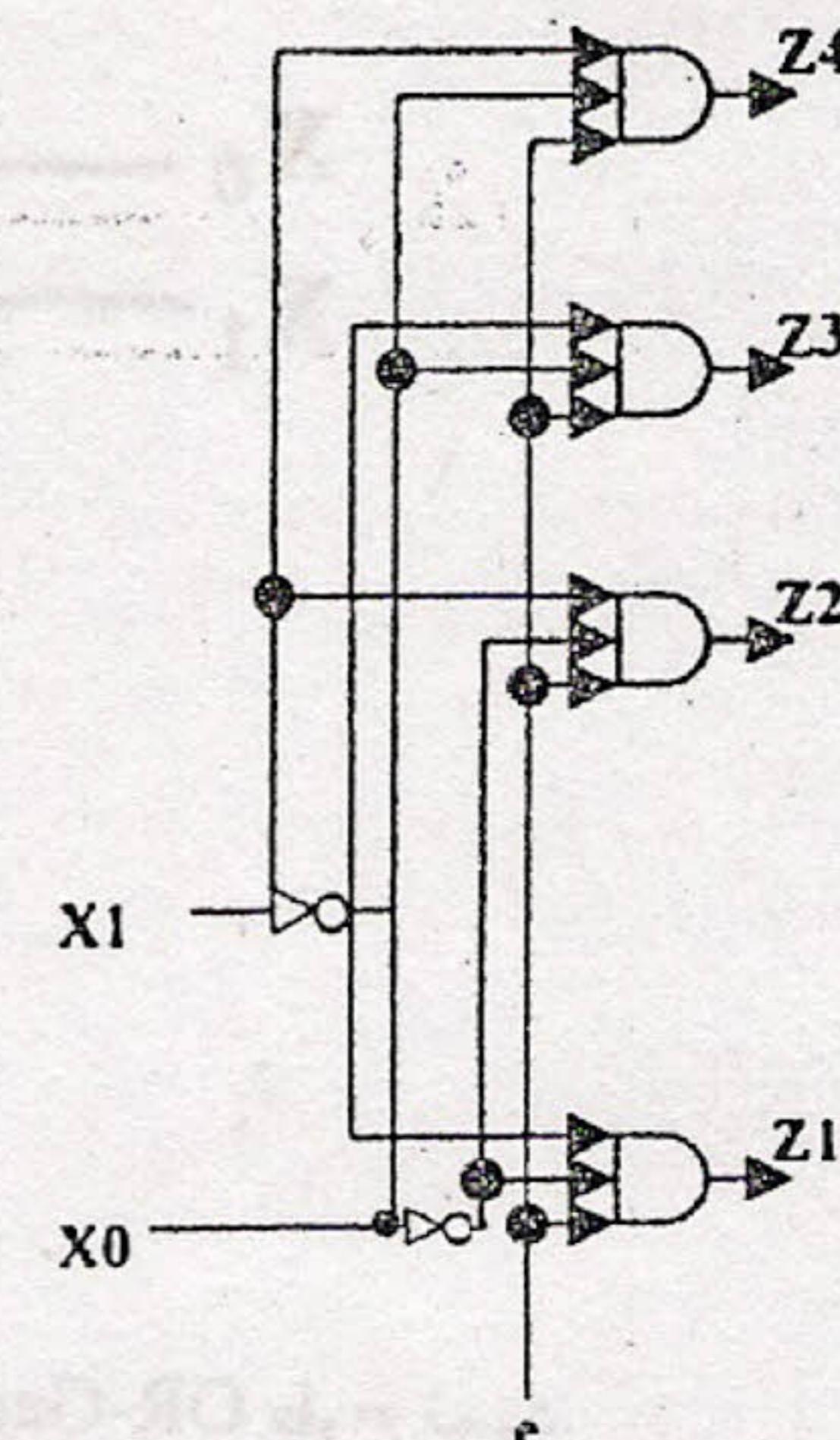
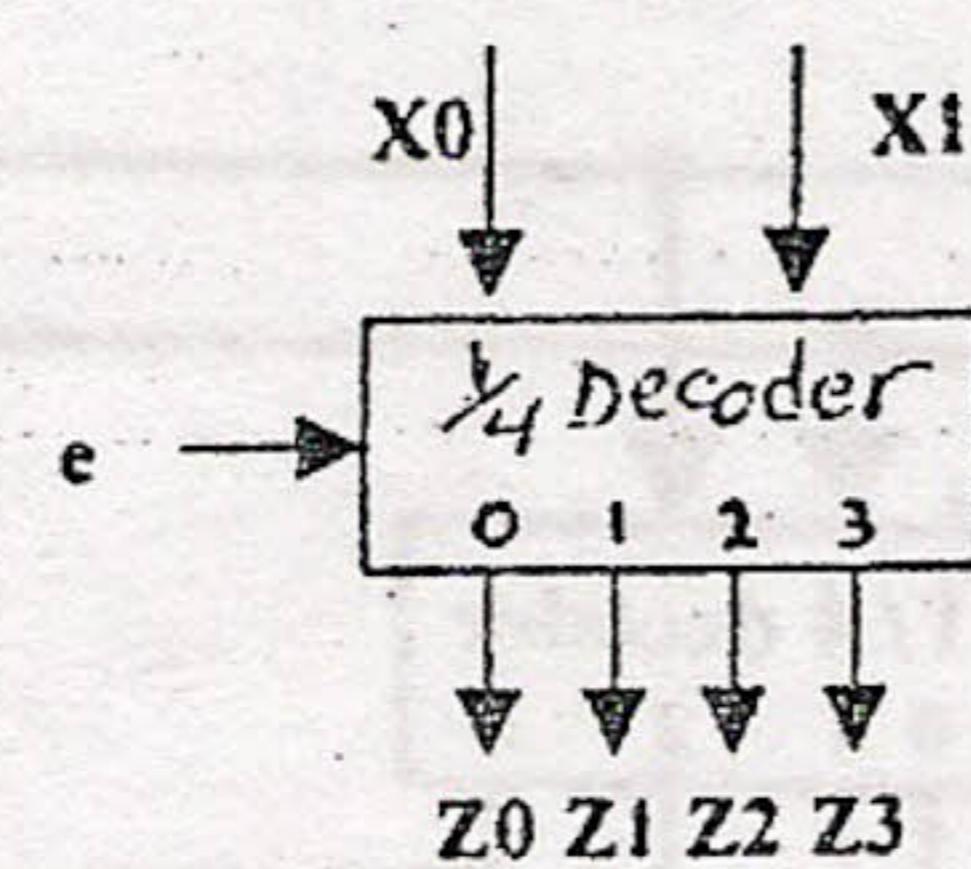
ولی آنچه در این درس منظور است، $\frac{1}{2^n}$ می‌باشد که در آن به ازا هر یک از ترکیبات ورودی تنها یک خروجی فعال و بقیه غیر فعال خواهند بود.



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

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

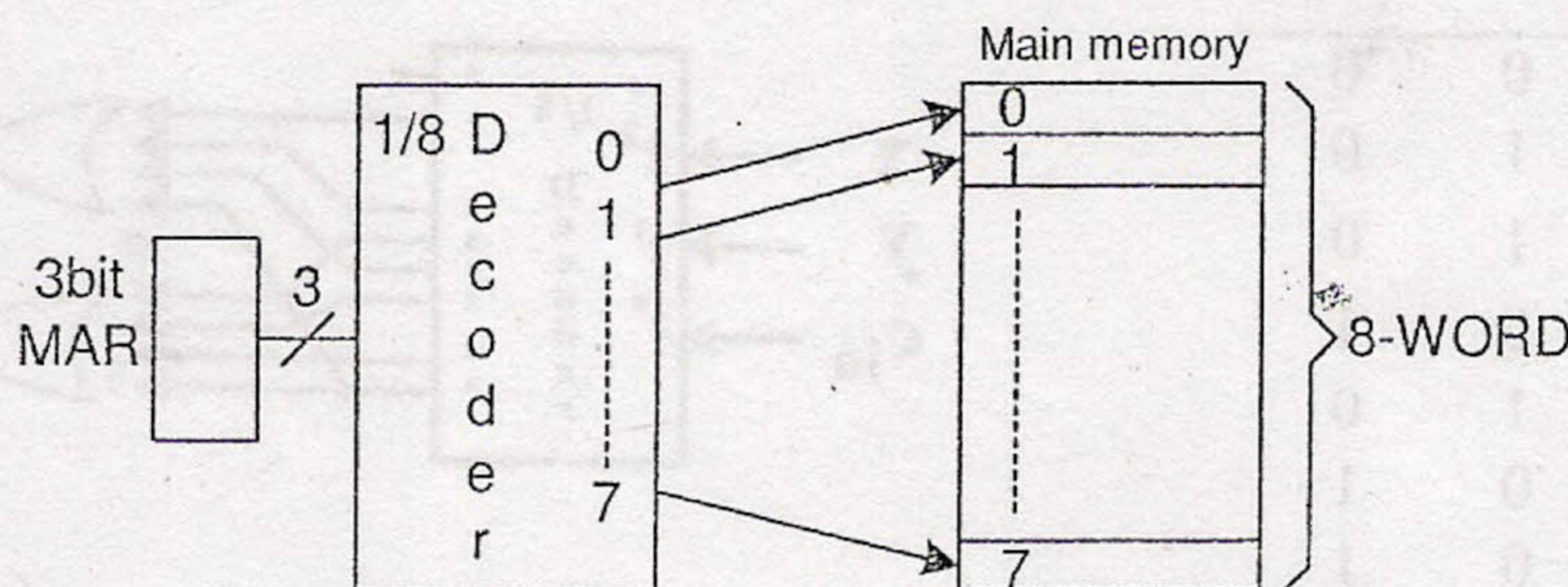
x_0	x_1	z_0	z_1	z_2	z_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



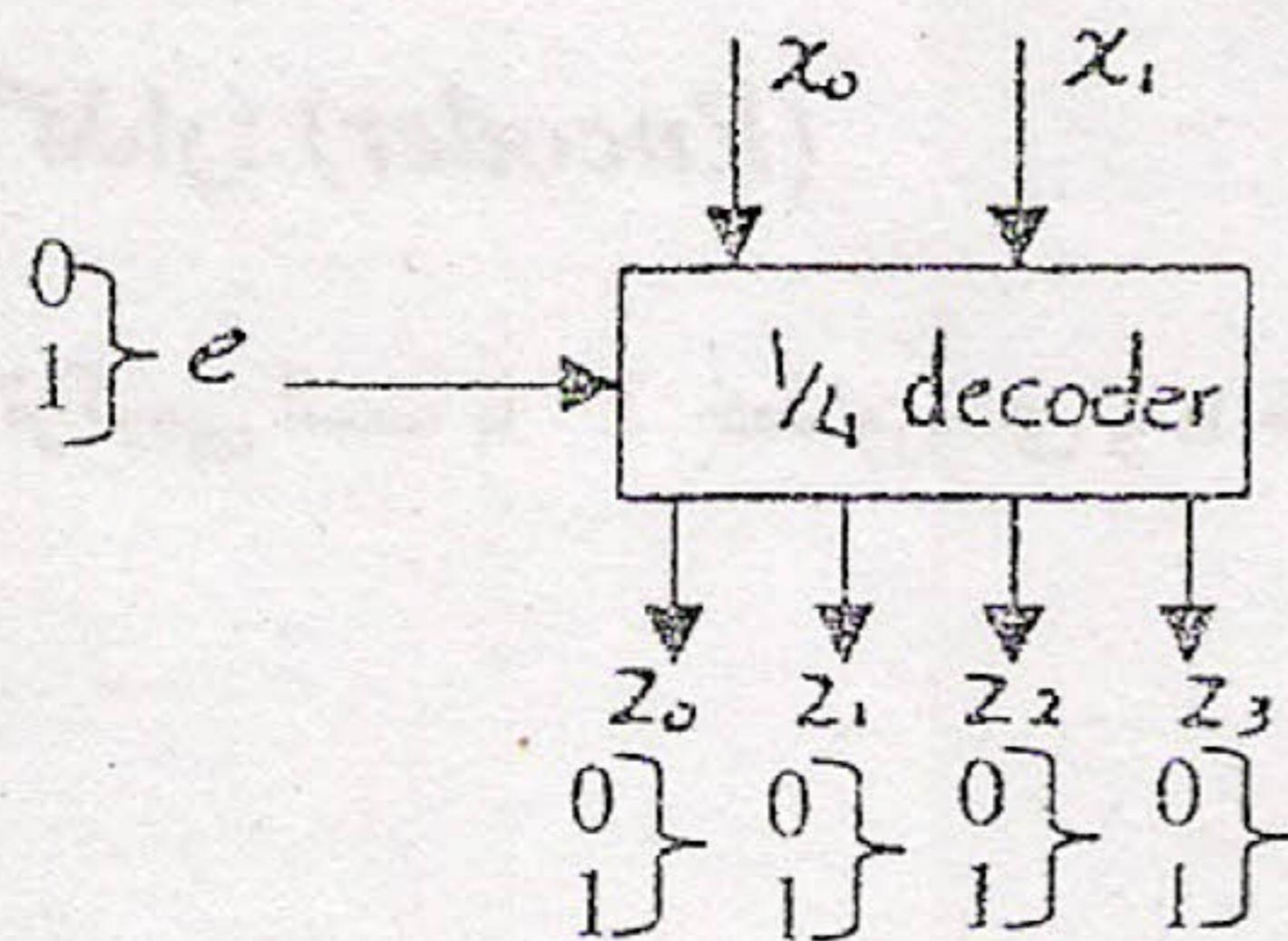
کاربردهای رمزگشایی:

- ۱) برای تبدیل کدها به یکدیگر مانند: Decimal Seven Segment Decoder که ورودی Binary را در خروجی به ارقام ۰-۹ تبدیل می‌کند.
- ۲) برای نشانی دهنده کلمات حافظه اصلی: فرض کنید حافظه اصلی شامل ۸ کلمه باشد.

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



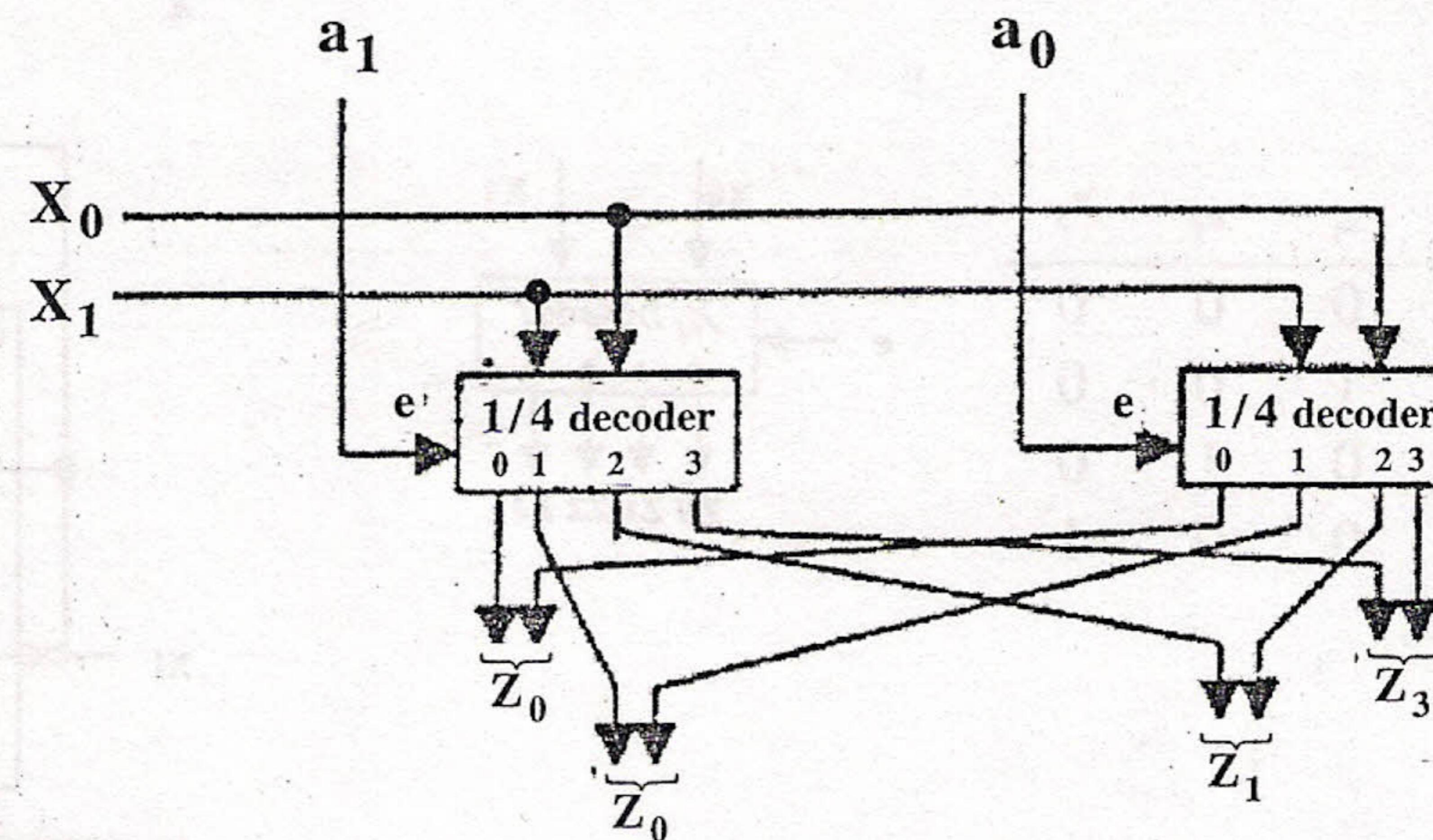
- ۳) در صورتی که خط Enable رمزگشا را به عنوان منبع مشترک تک بیتی تلقی کنیم، آن‌گاه از Decoder می‌توان برای هدایت اطلاعات تک بیتی از یک منبع مشترک به یکی از چندین مقصد متمایز استفاده نمود. چنین مداری را DeMUX گویند.



مثلثاً $\frac{1}{4}$ Decoder است که با دادن آدرس مناسب به x_0 ، می‌توان اطلاعات تک بیتی واقع در خط x_1 را به یکی از چهار مقصد z_3, z_2, z_1, z_0 هدایت نمود.

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

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



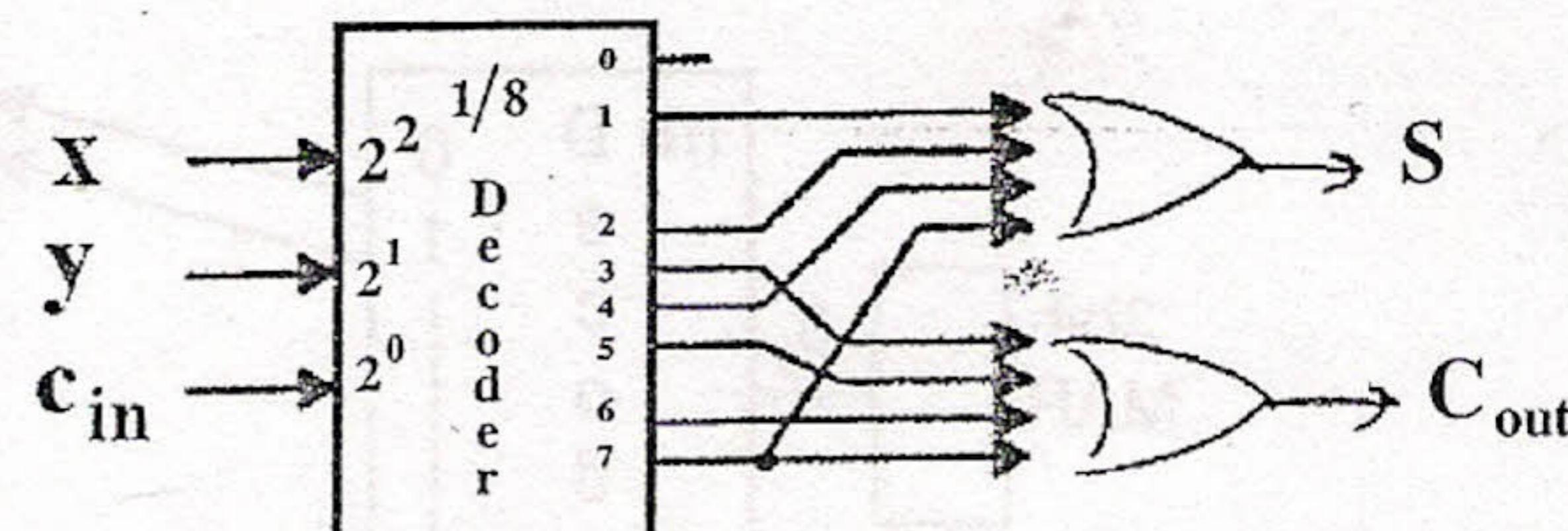
۴) هر مدار ترکیبی با n ورودی و m خروجی را می‌توان به وسیله یک $\frac{1}{2^n}$ Decoder و m عدد OR-Gate طرح نمود.

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

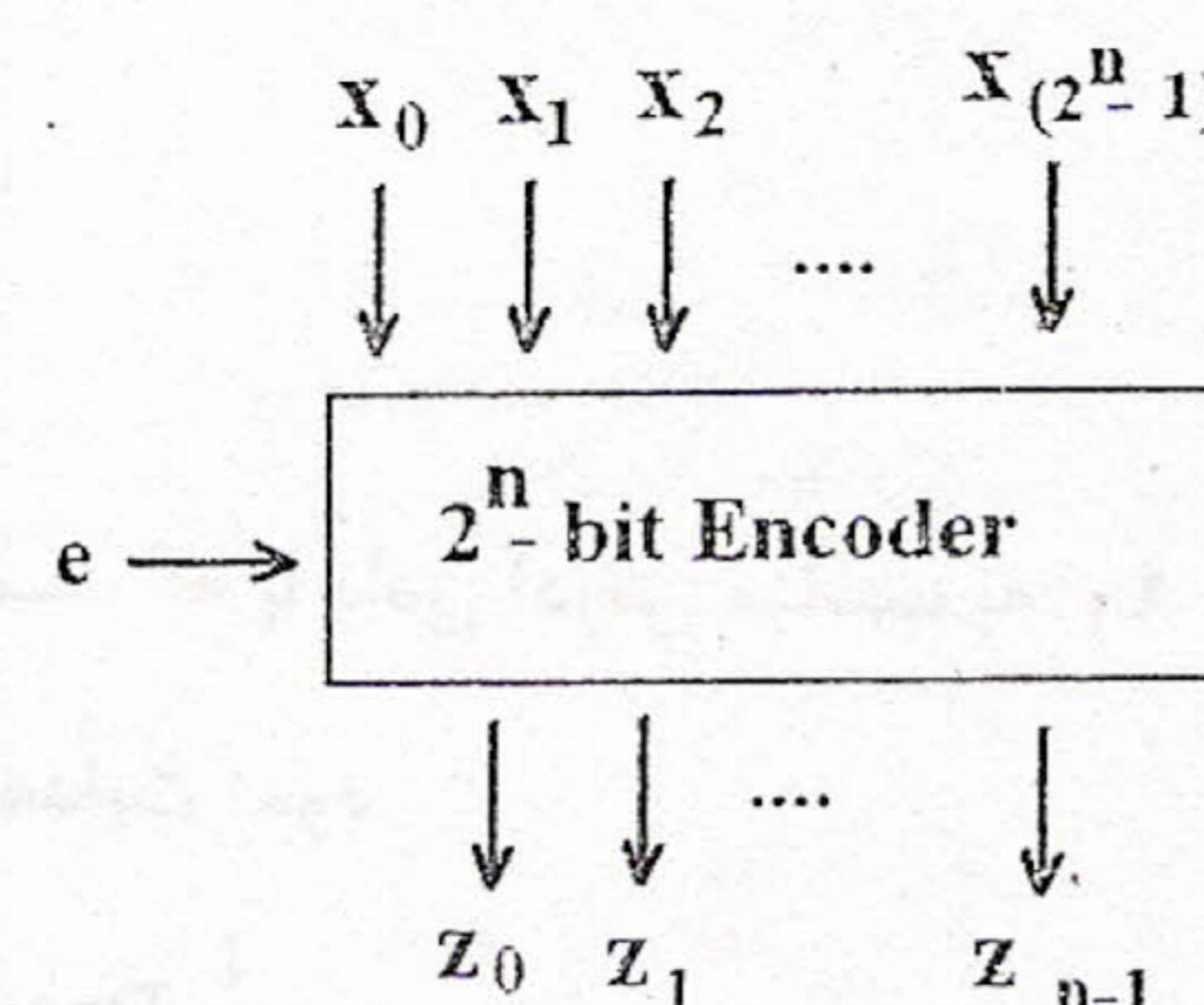


$$S(x, y, C_m) = \sum_1 (1, 2, 4, 7)$$

$$C_{out}(x, y, C_m) = \sum_1 (3, 5, 6, 7)$$

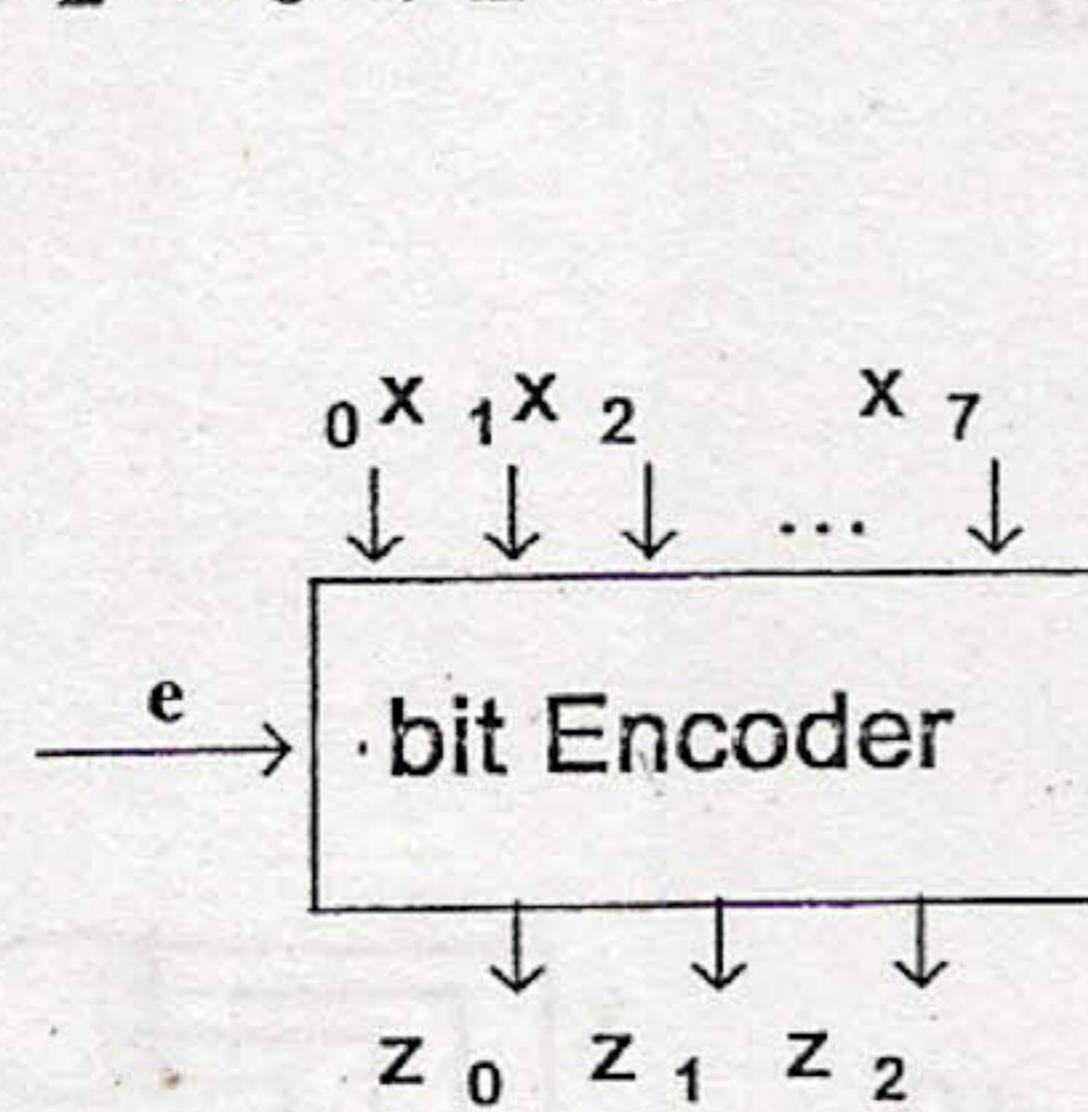
رمزگذار: (Encoder)

مدار ترکیبی است با 2^n خط ورودی و n خروجی که در آن خروجی‌ها آدرس خط ورودی فعال را تولید می‌کند.



مثال: مطلوب است طراحی 8-bit Encoder

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

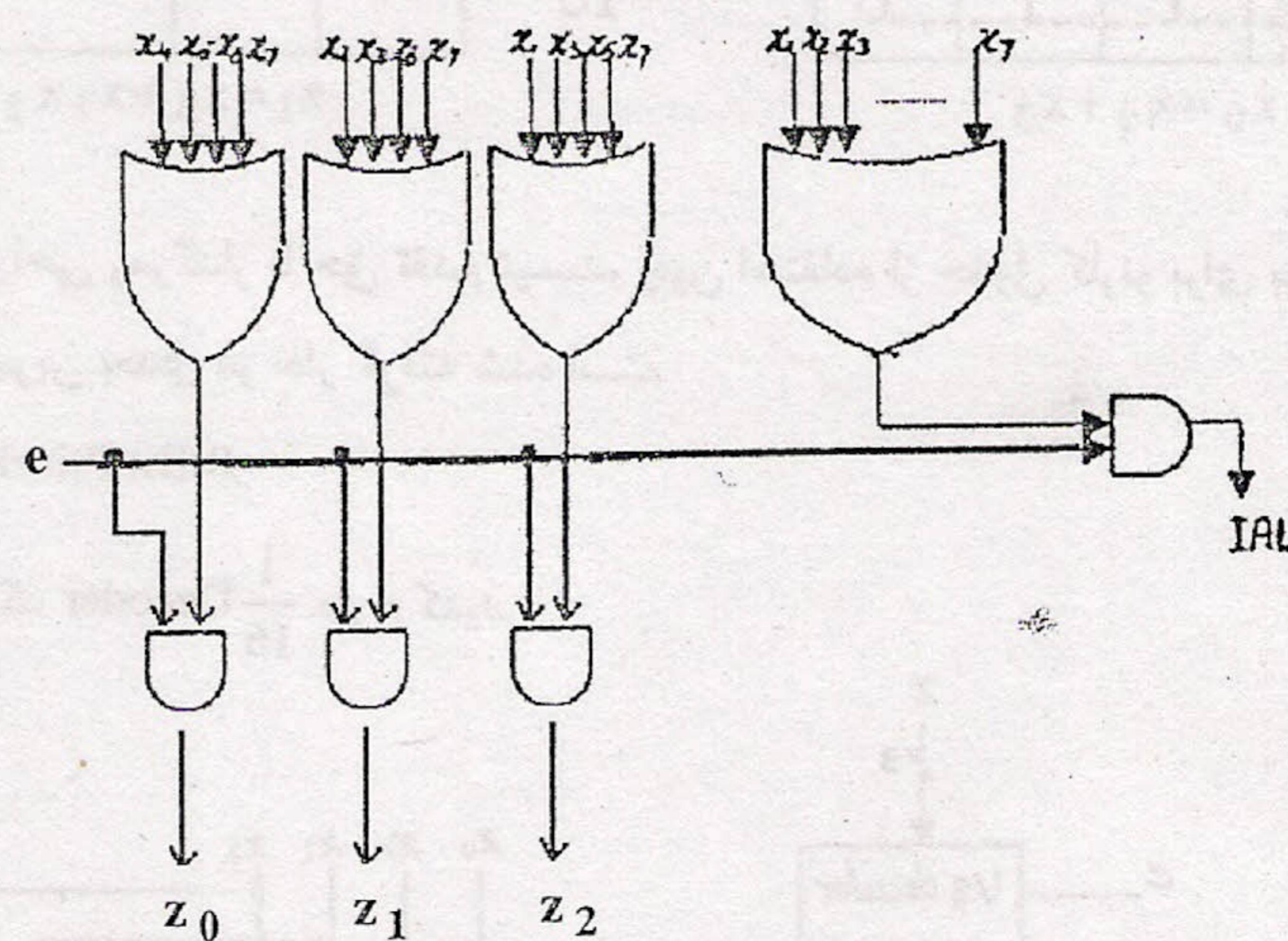


x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	z_0	z_1	z_2
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

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

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

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



طراحی مدار به صورت فوق دو عیب اصلی دارد:

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

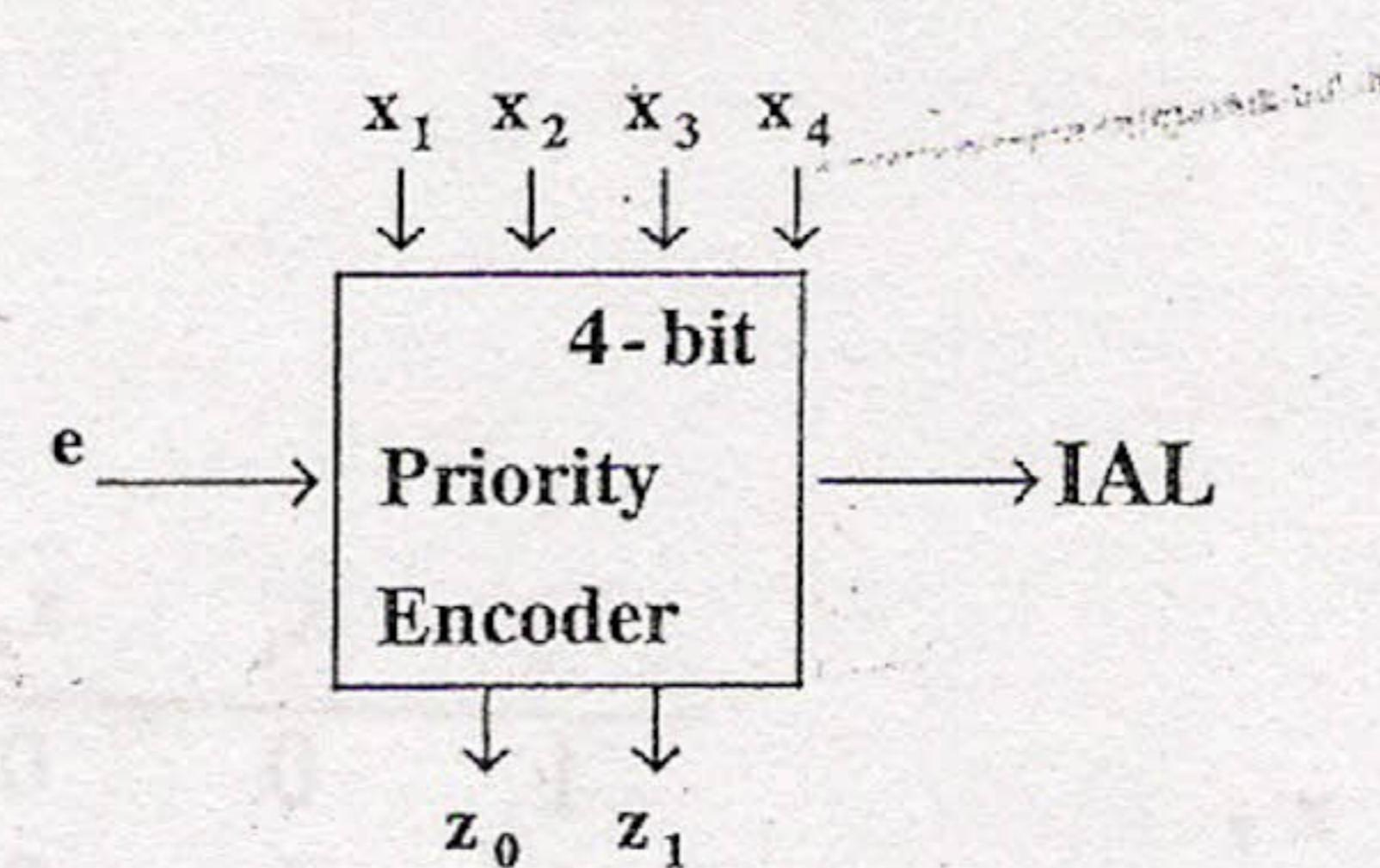
$$\left. \begin{array}{l} x_1 = 1 \\ x_2 = 1 \end{array} \right\} \Rightarrow \text{آدرس نامعتبر که متناظر با خط ورودی فعال } x_3 \text{ می‌باشد فعال خواهد شد. } 011$$

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



x_1x_2

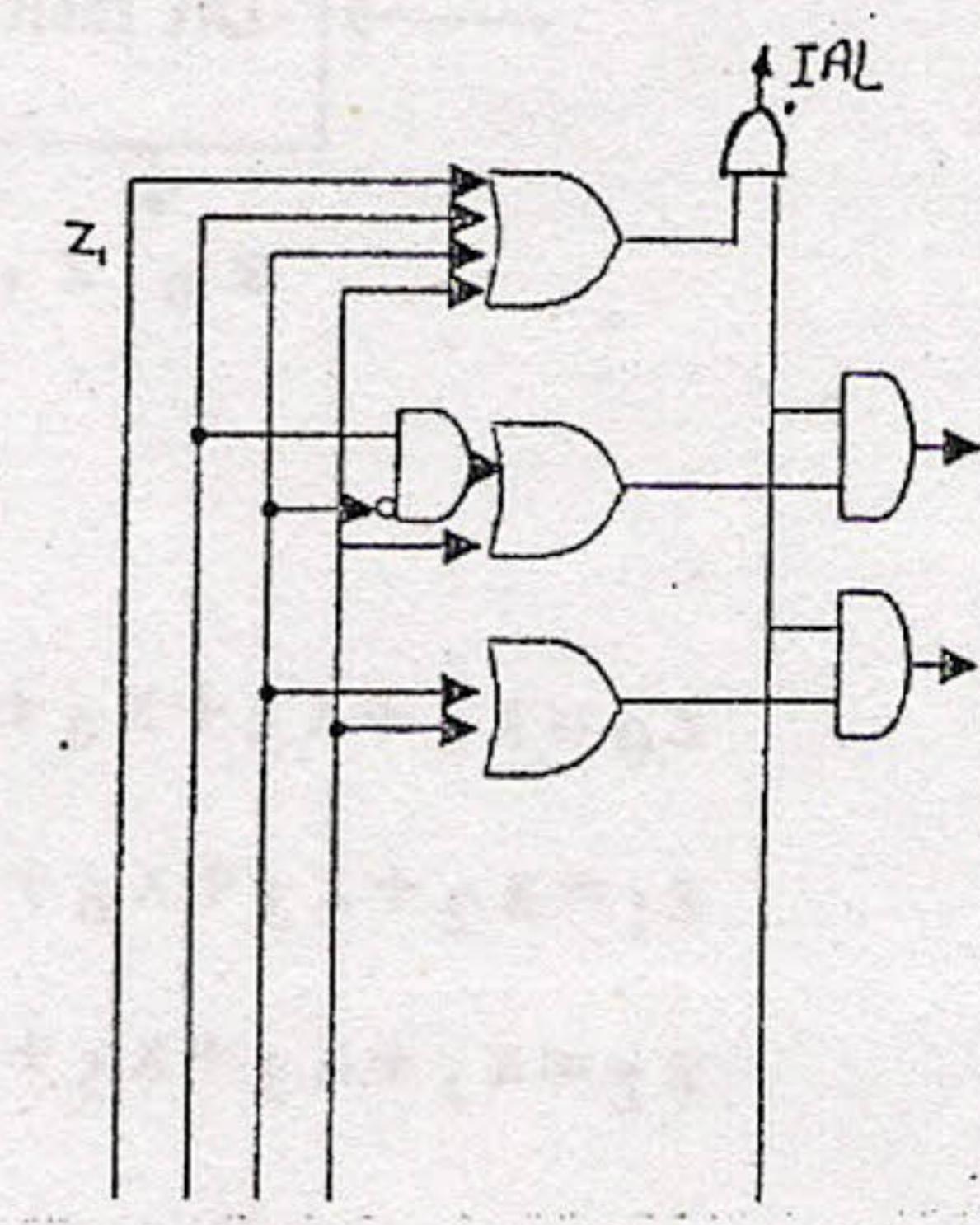
	00	01	11	10
00				
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$z_0 = x_2 + x_3$

x_1x_2

	00	01	11	10
00		1	1	
01	1	1	1	1
11	1	1	1	1
10				

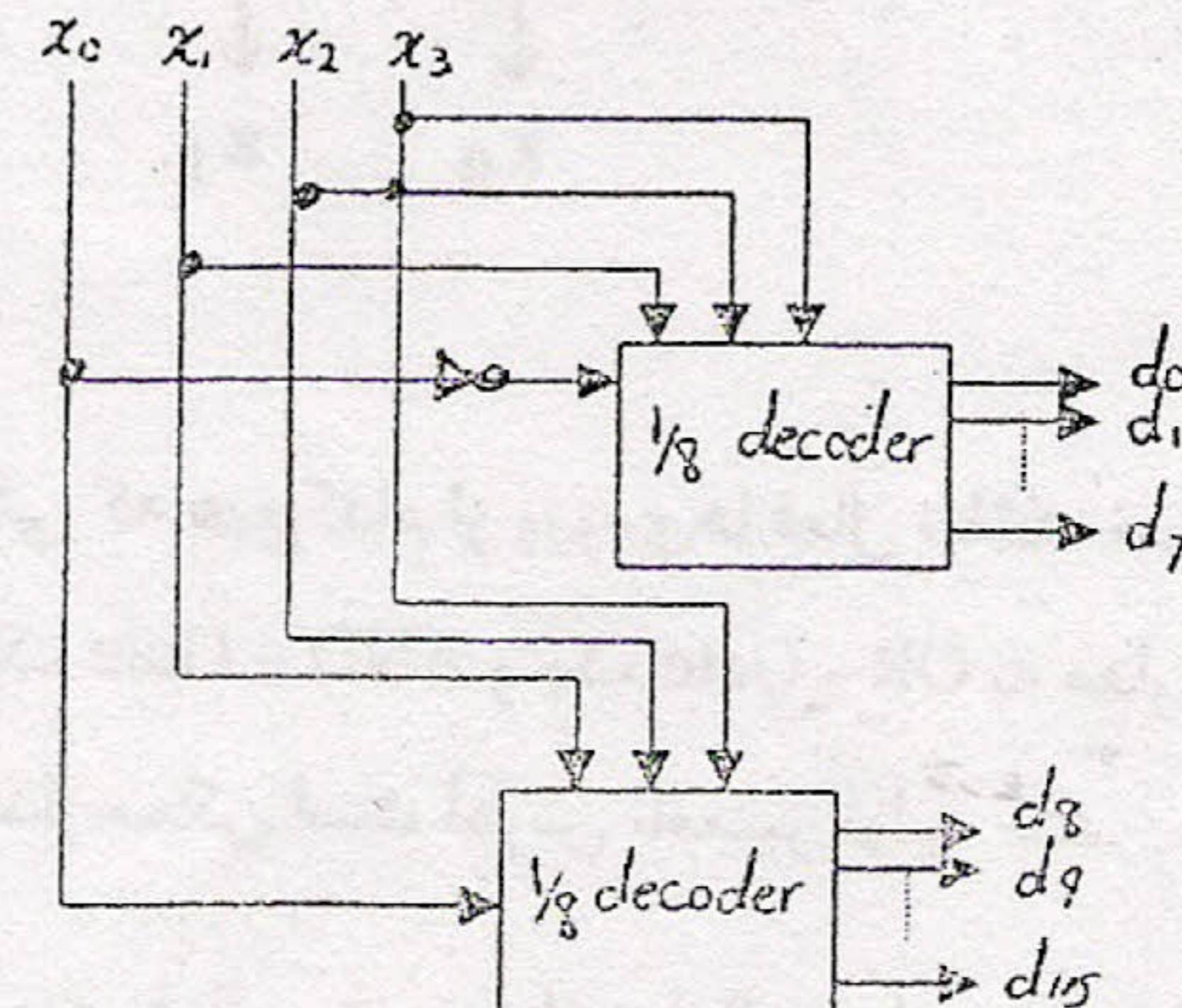
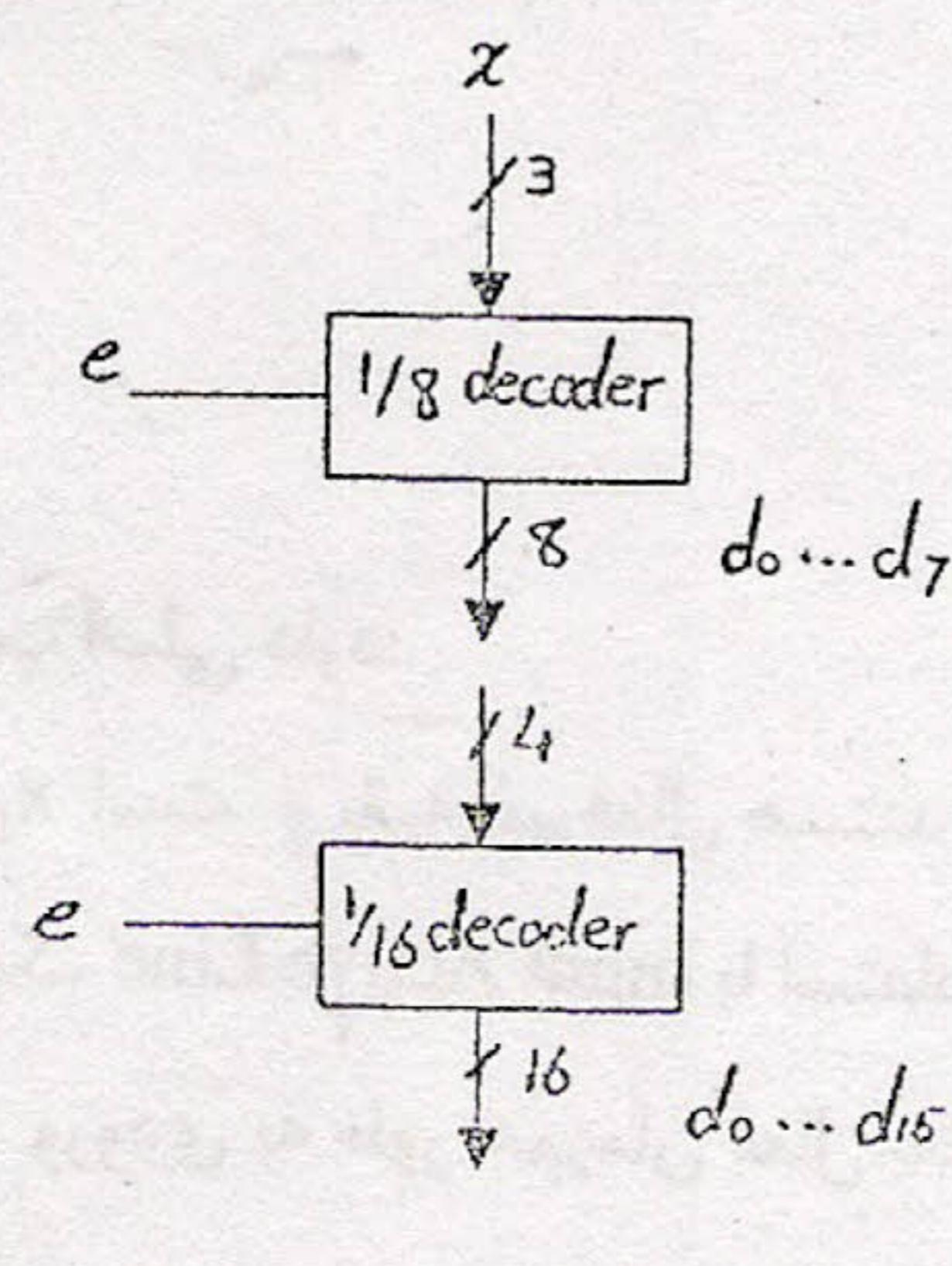
$z_1 = x_3 + x_1 x_2$



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 راه حلی است که برای تمرین بعدی در نظر گرفته شده است.

تمرین: مطلوب است طراحی 8-Bit Priority Encoder

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



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

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

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

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

در صورتی که مبنای مقایسه bit یا بالاترین ارزش مکانی باشد، آن‌گاه بدون تشکیل جدول ارزش می‌توان مدار مقایسه کننده را طرح نمود.

$$A = A_3 A_2 A_1 A_0$$

$$B = B_3 B_2 B_1 B_0$$

متغیر بولی x_i را چنان تعریف می‌کنیم، اگر $A_i = B_i$ آن‌گاه $x_i = 0$ و اگر $A_i \neq B_i$ آن‌گاه $x_i = 1$ باشد. یعنی:

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

$$(A = B) = x_3 x_2 x_1 x_0$$

$$(A < B) = \overline{A_3} B_3 + x_3 \overline{A_2} B_2 + x_3 x_2 \overline{A_1} B_1 + x_3 x_2 x_1 \overline{A_0} B_0$$

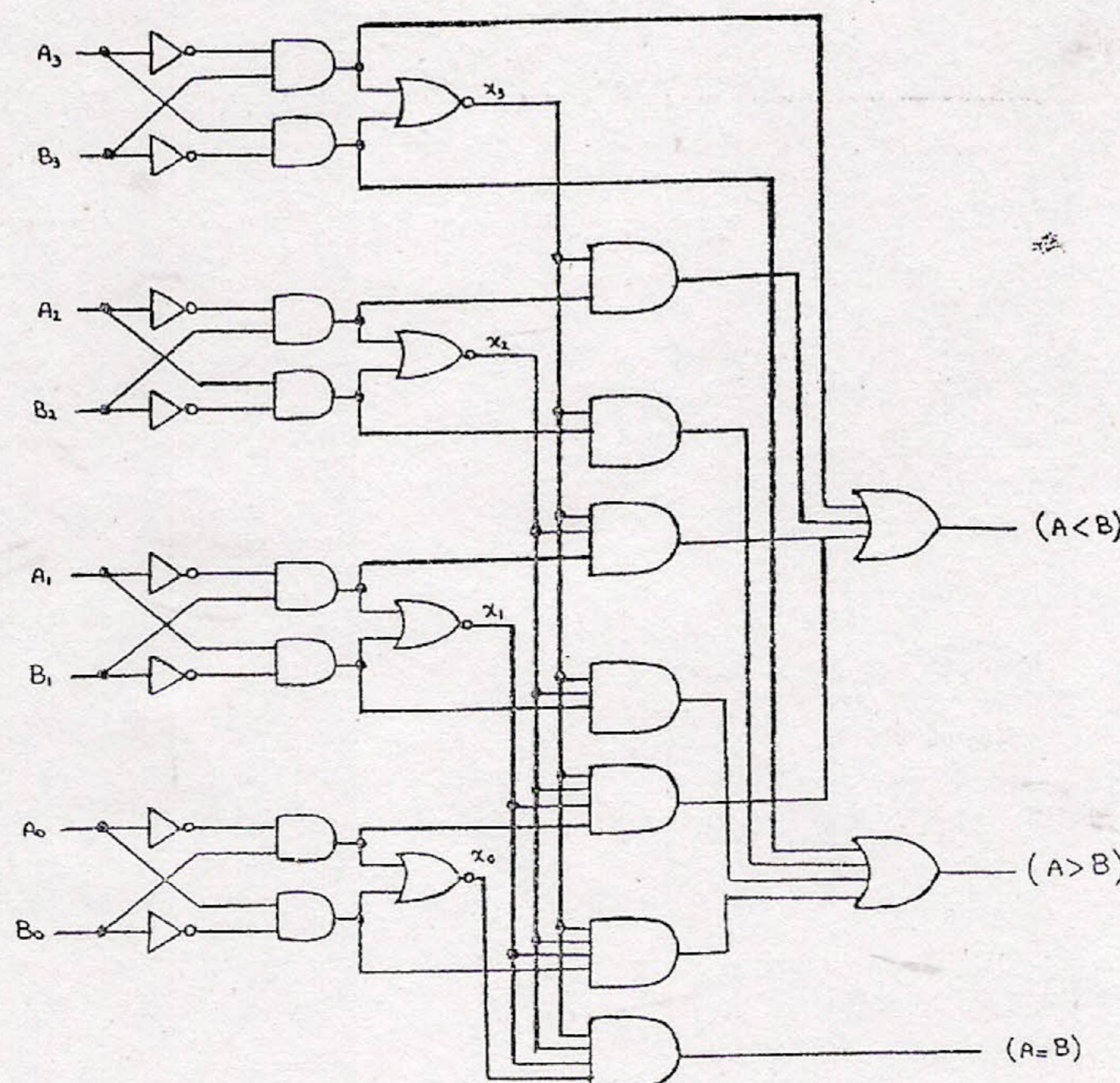
$$(A > B) = \overline{A_3} B_3 + x_3 A_2 \overline{B_2} + x_3 x_2 A_1 \overline{B_1} + x_3 x_2 x_1 A_0 \overline{B_0}$$

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

$$x_i = A_i B_i + \overline{A_i} \overline{B_i} = \overline{A_i \oplus B_i}$$

اثبات:

$$\overline{A_i \oplus B_i} = \overline{(\overline{A_i} B_i)(A_i \overline{B_i})} = (\overline{A_i} + \overline{B_i})(\overline{A_i} + B_i) = (0 + A_i B_i) + (\overline{B_i} A_i + 0) = A_i B_i + \overline{B_i} \overline{A_i}$$



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 min دو سطحی AND - OR تابع را به دست آورید.

(D) فرم min 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	00		1	-
	01			1
	11	-	-	1
	10	1		1

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

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

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

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

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