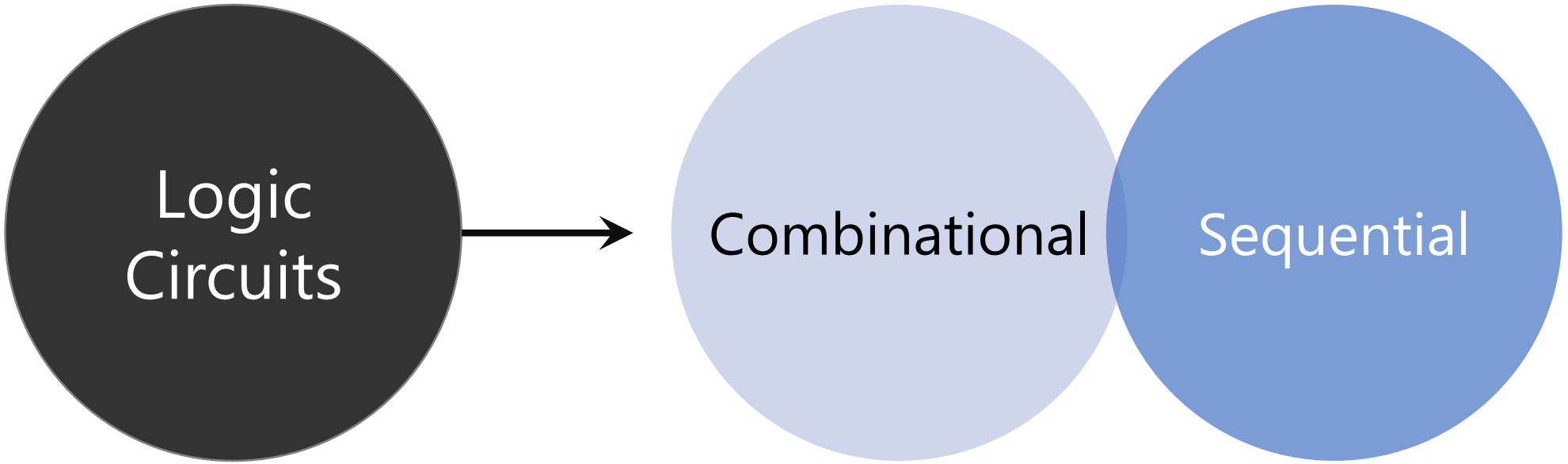

Logic
Circuits
Combinational
Sequential

Chapter 4 Combinational Logic

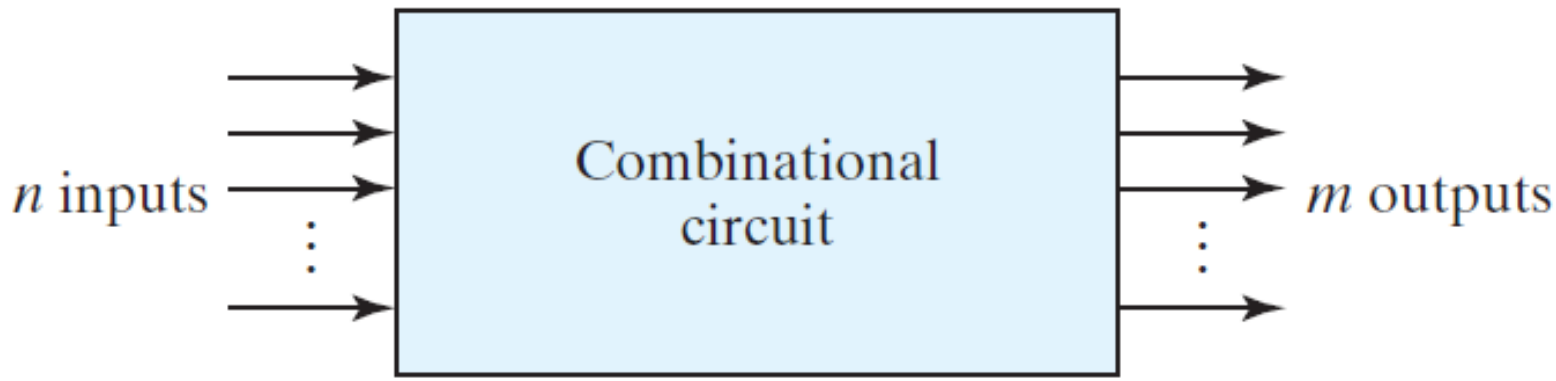


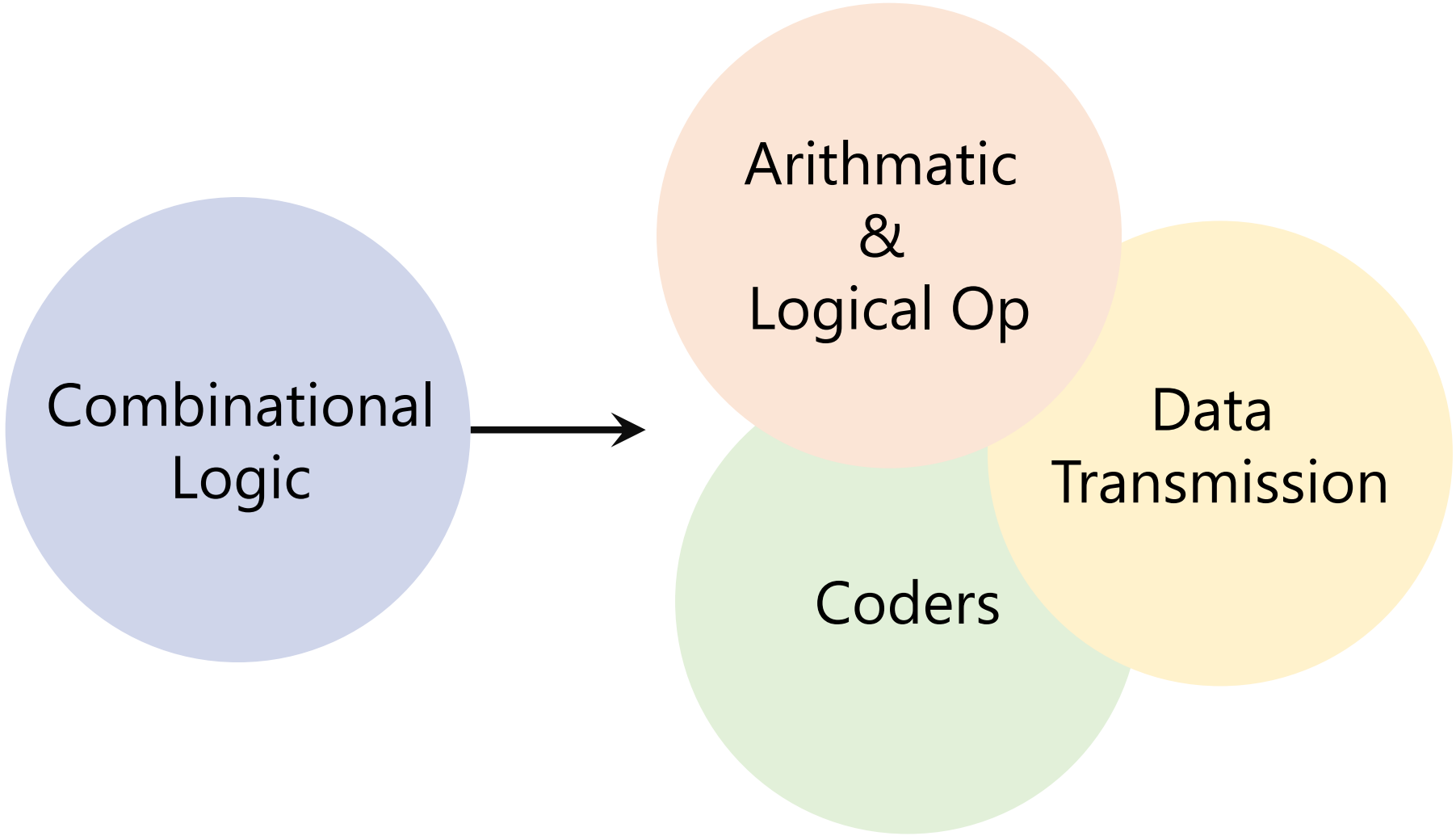
FIGURE 4.1
Block diagram of combinational circuit

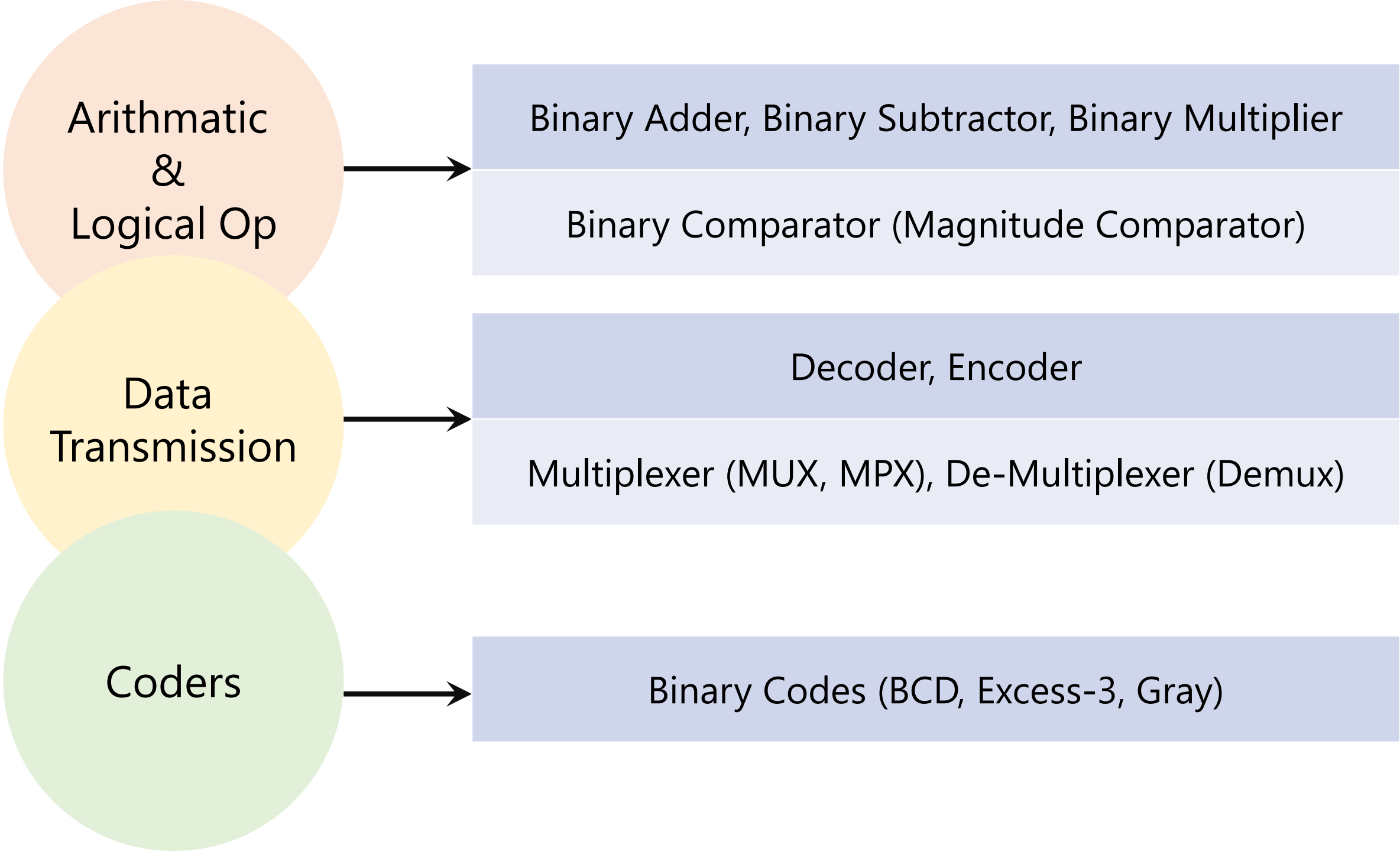
Combinational Logic

aka. Combinational Circuit

Combination of logic gates on the present inputs → the outputs *at any time*!

A combinational circuit performs an operation that can be specified logically by a set of Boolean functions.


Combinational
Logic
Arithmatic
&
Logical Op
Data
Transmission
Coders


Arithmatic
&
Logical Op
Binary Adder, Binary Subtractor, Binary Multiplier
Binary Comparator (Magnitude Comparator)
Data
Transmission
Decoder, Encoder
Multiplexer (MUX, MPX), De-Multiplexer (Demux)
Coders
Binary Codes (BCD, Excess-3, Gray)


Data
Transmission
Decoder, Encoder
Multiplexer (MUX, MPX), De-Multiplexer (Demux)

Binary Decoder

BCD Decoder
Display Decoder

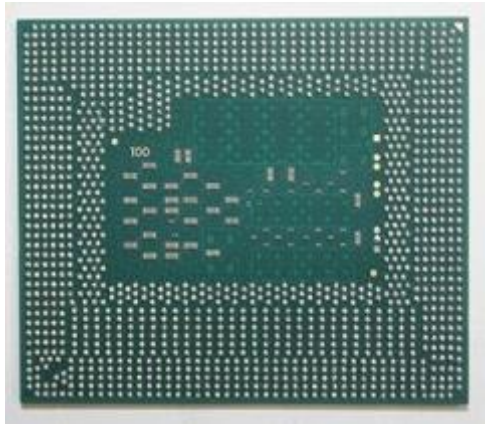
Decoder
Decode Binary to 1-hot

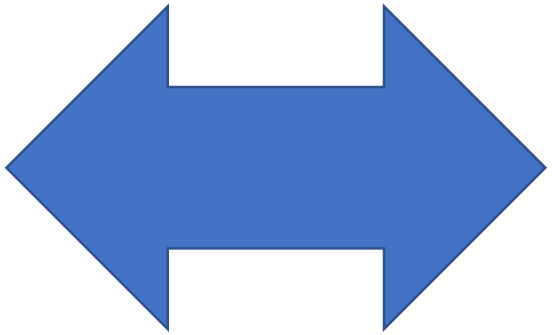
1-hot: a vector of bits with a single 1 and all the others 0

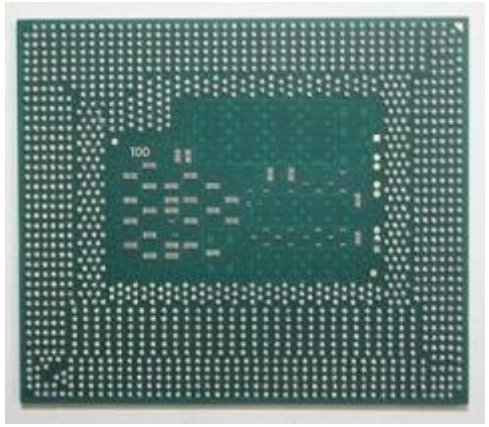
[0010000000]

[0000000100]

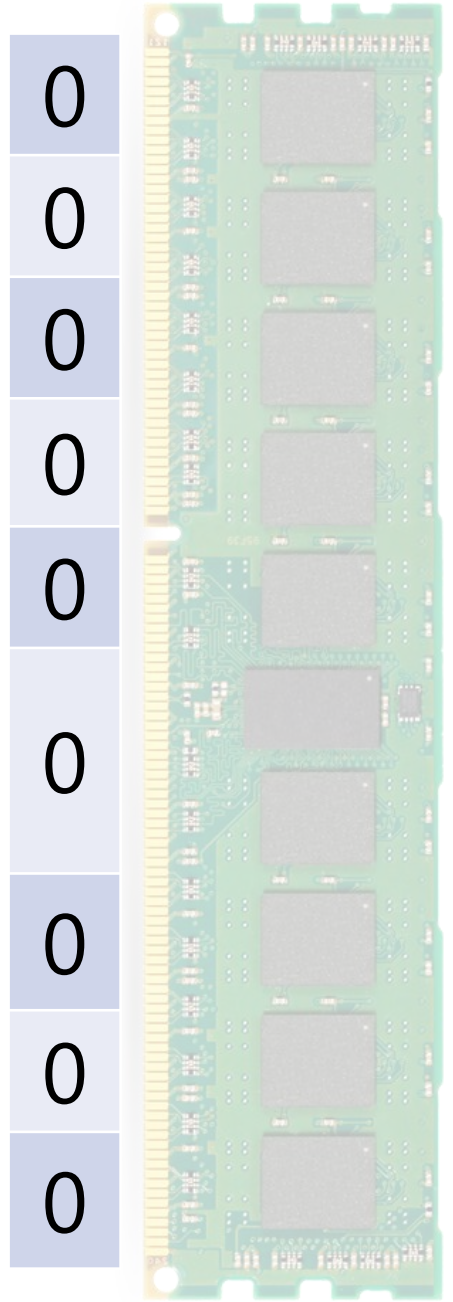
~~[0010010000]~~

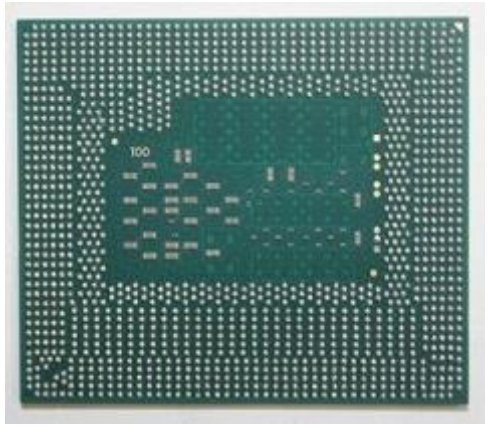




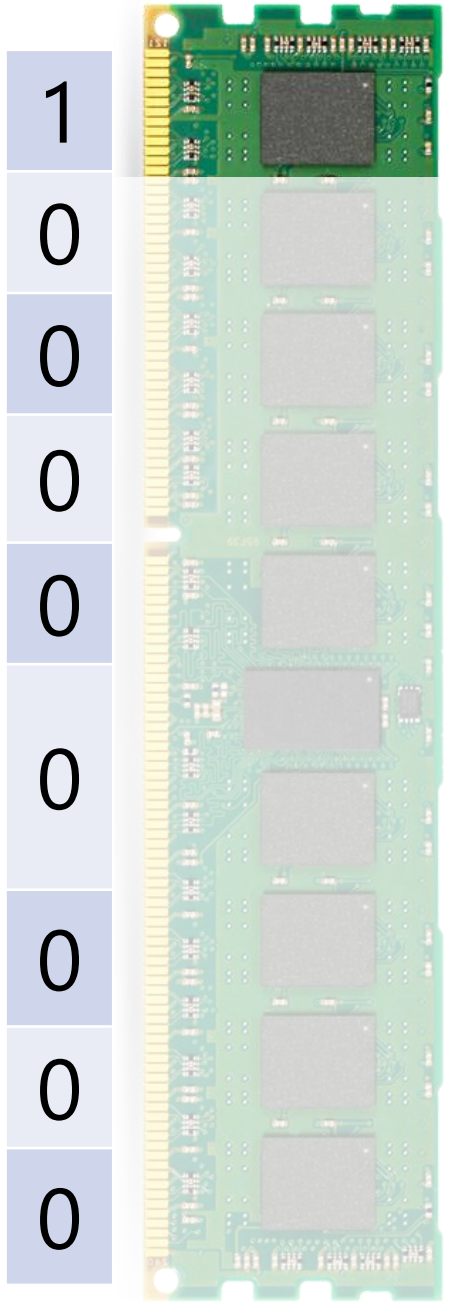


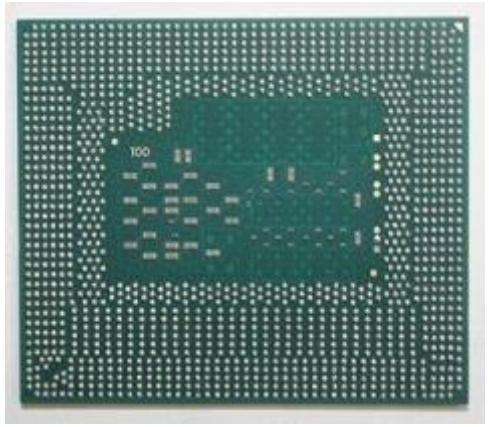
Address



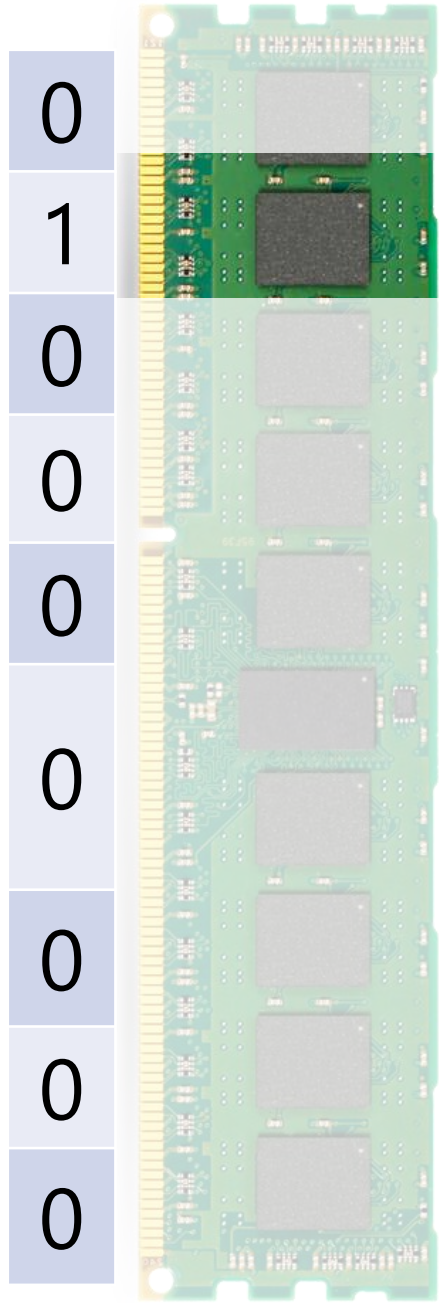


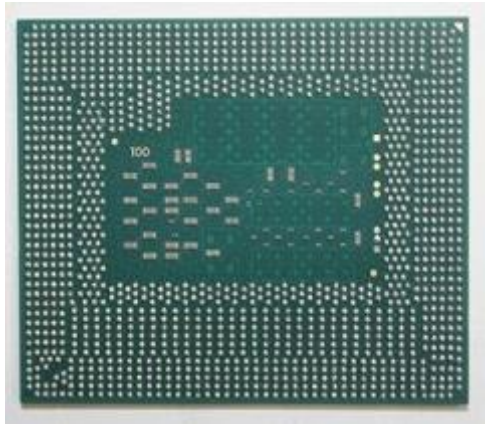
Address



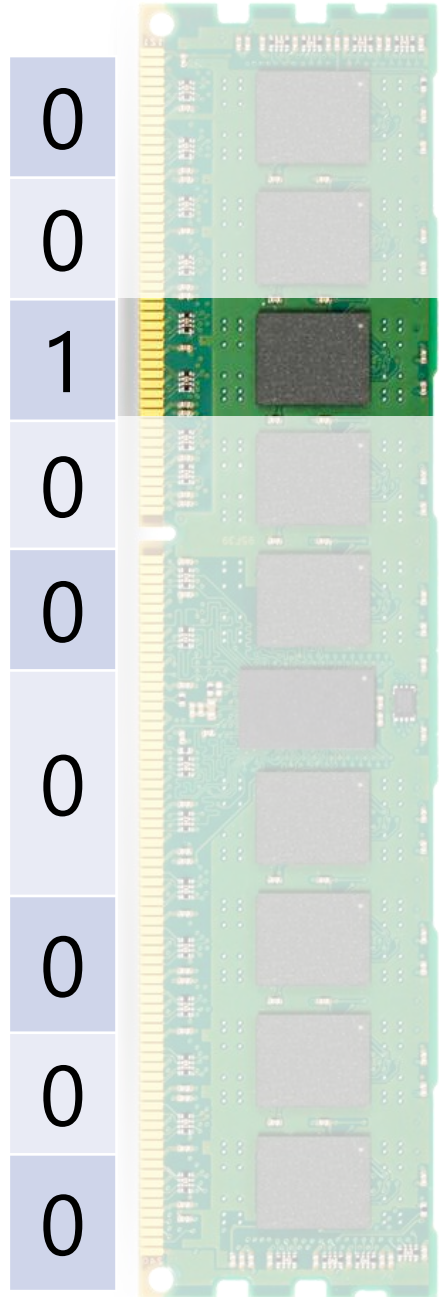


Address





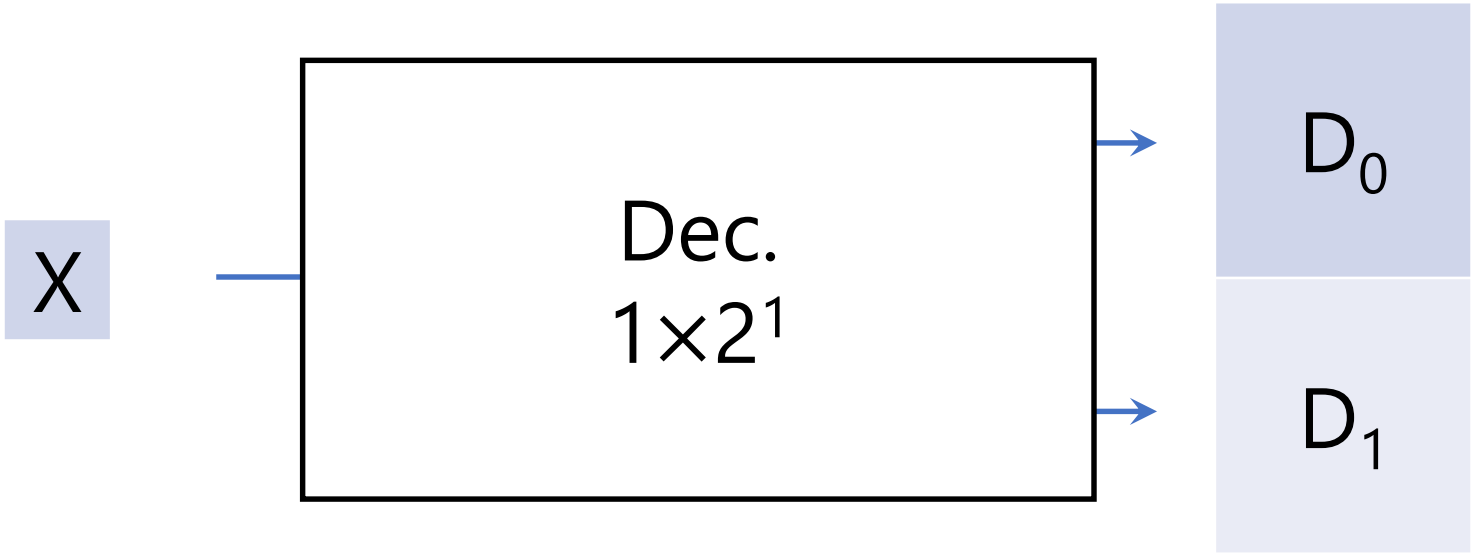
Address




X
D_0
D_1

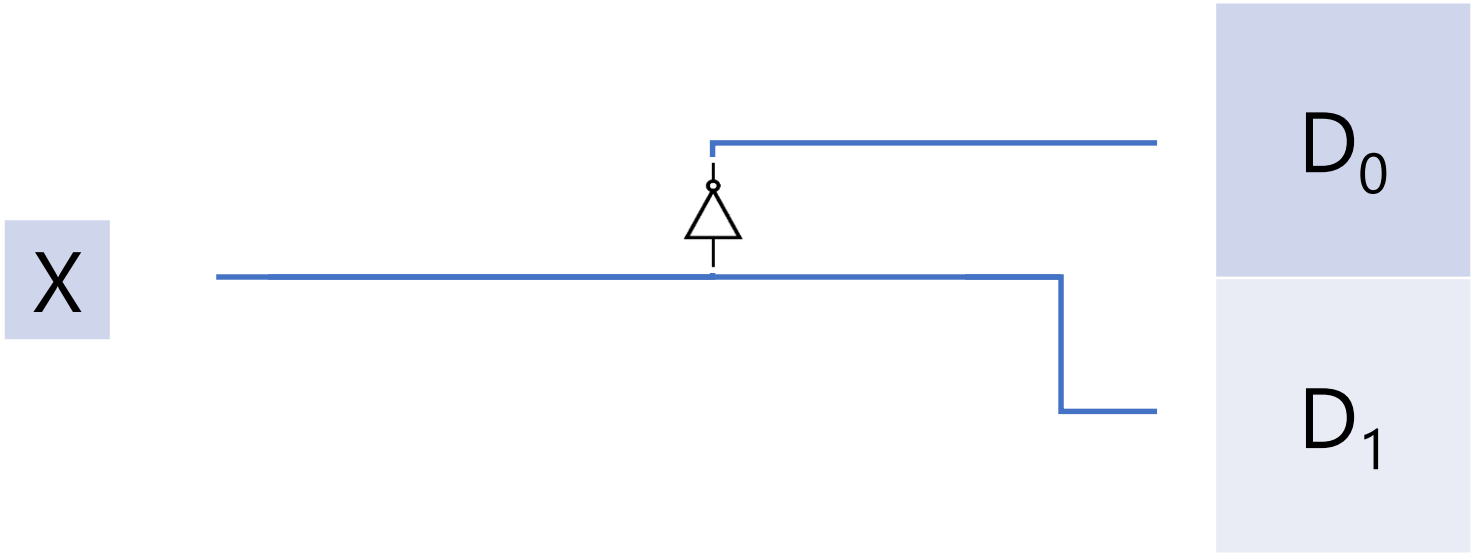

0
$D_0=1$
$D_1=0$

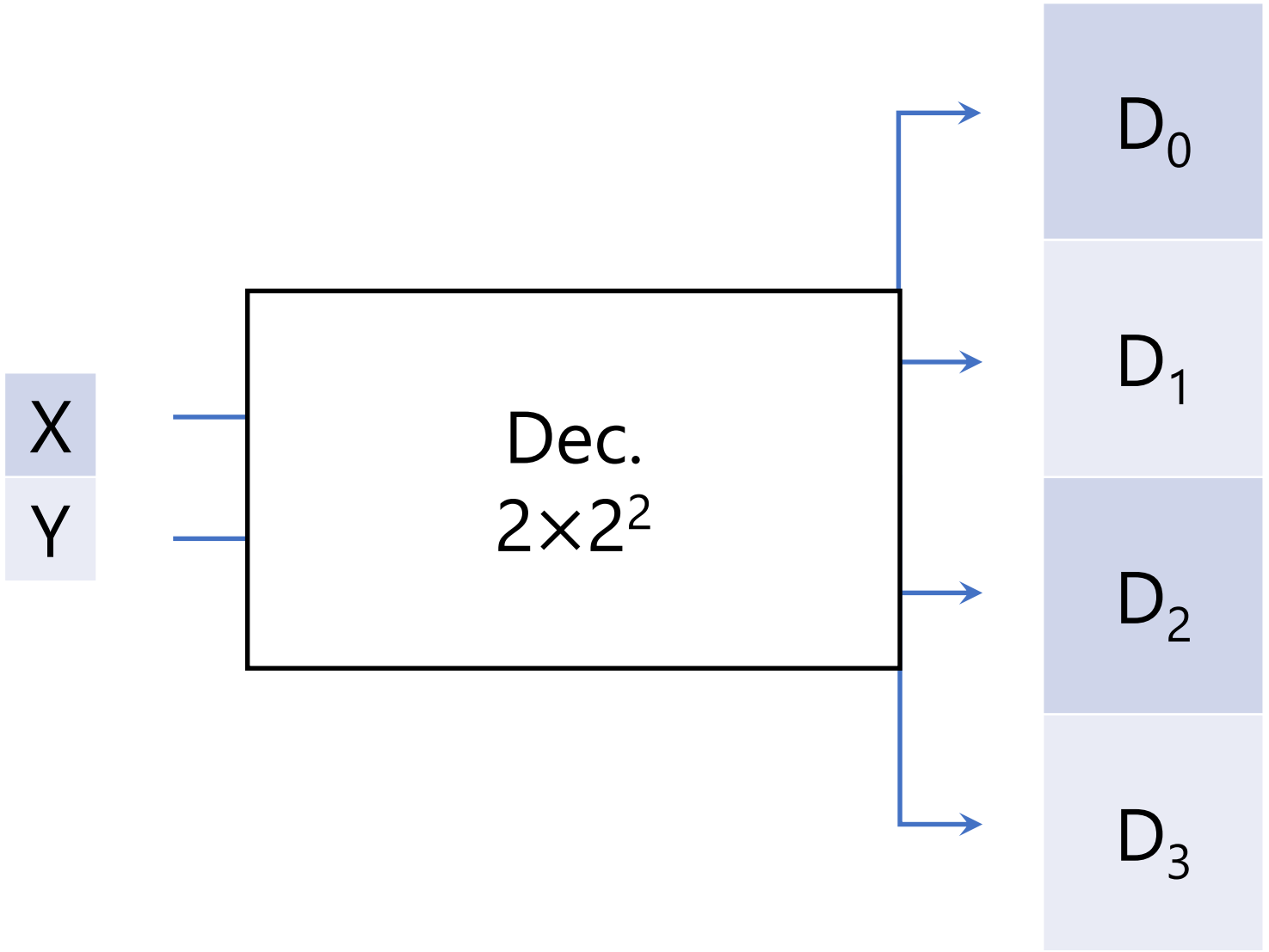

1
$D_0=0$
$D_1=1$

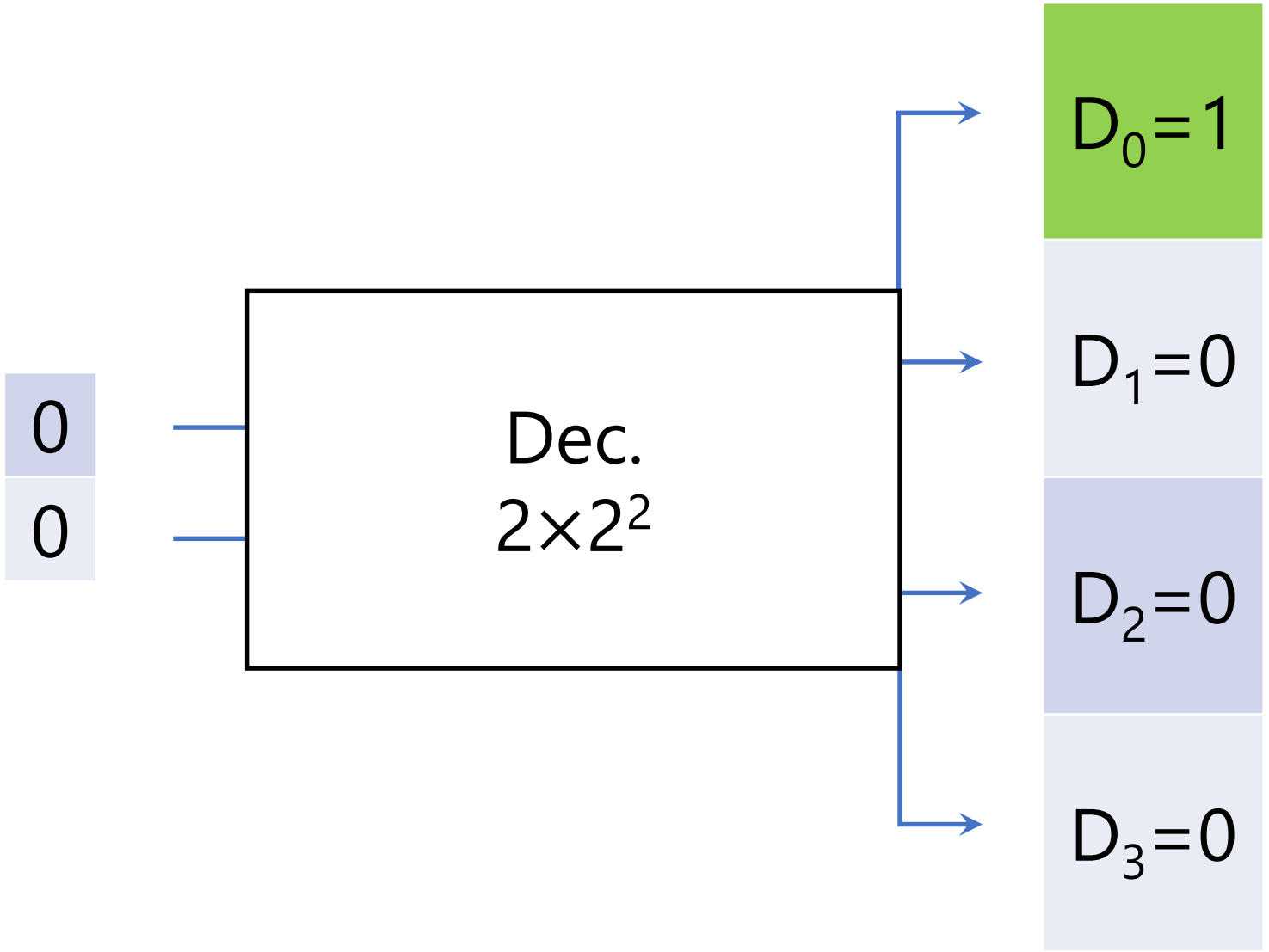

X
Dec.
1×2¹
D0
D1

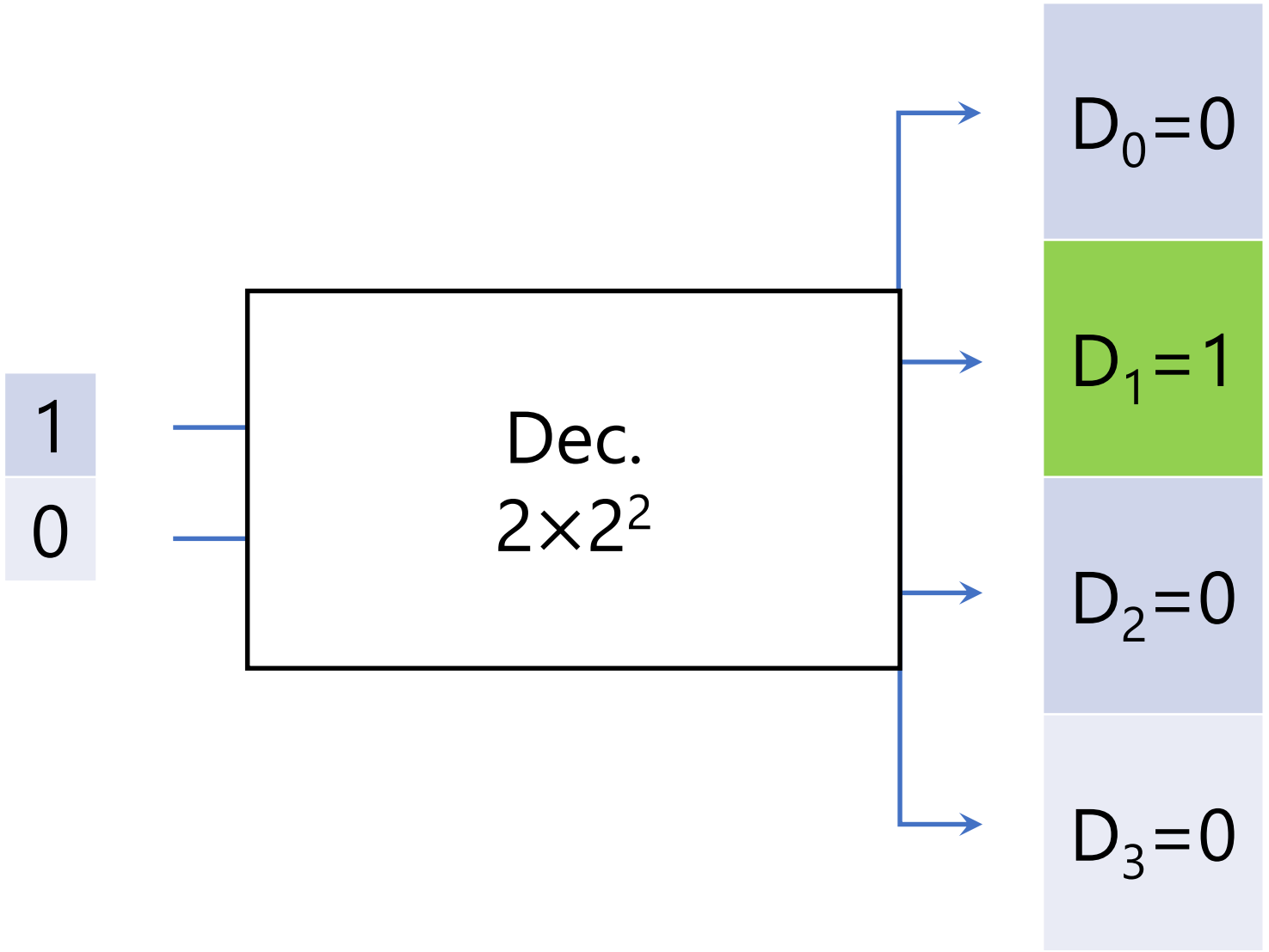
X	D_0	D_1
0	1	0
1	0	1

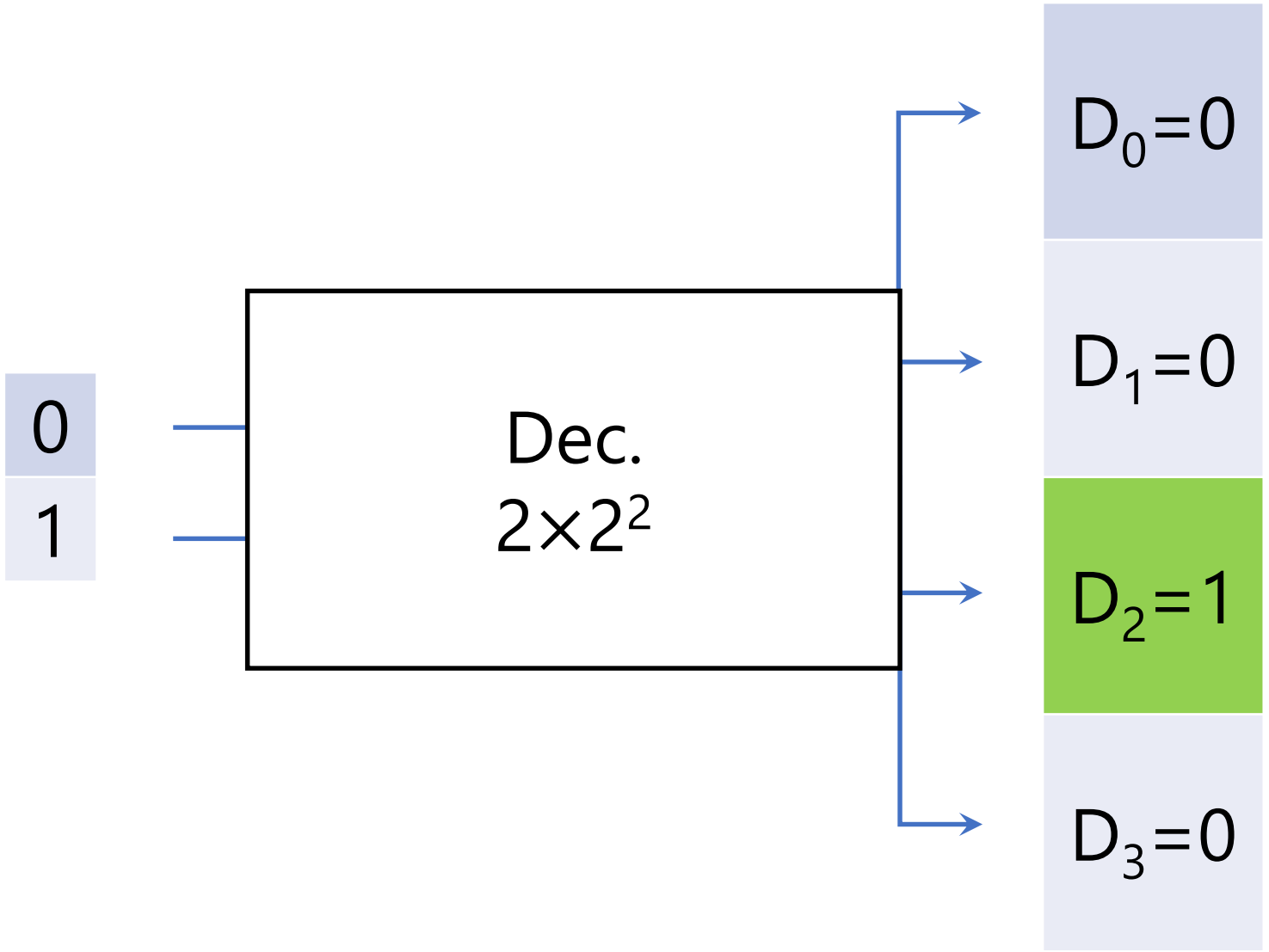
X	$D_0 = m_0$	$D_1 = m_1$
0	1	0
1	0	1

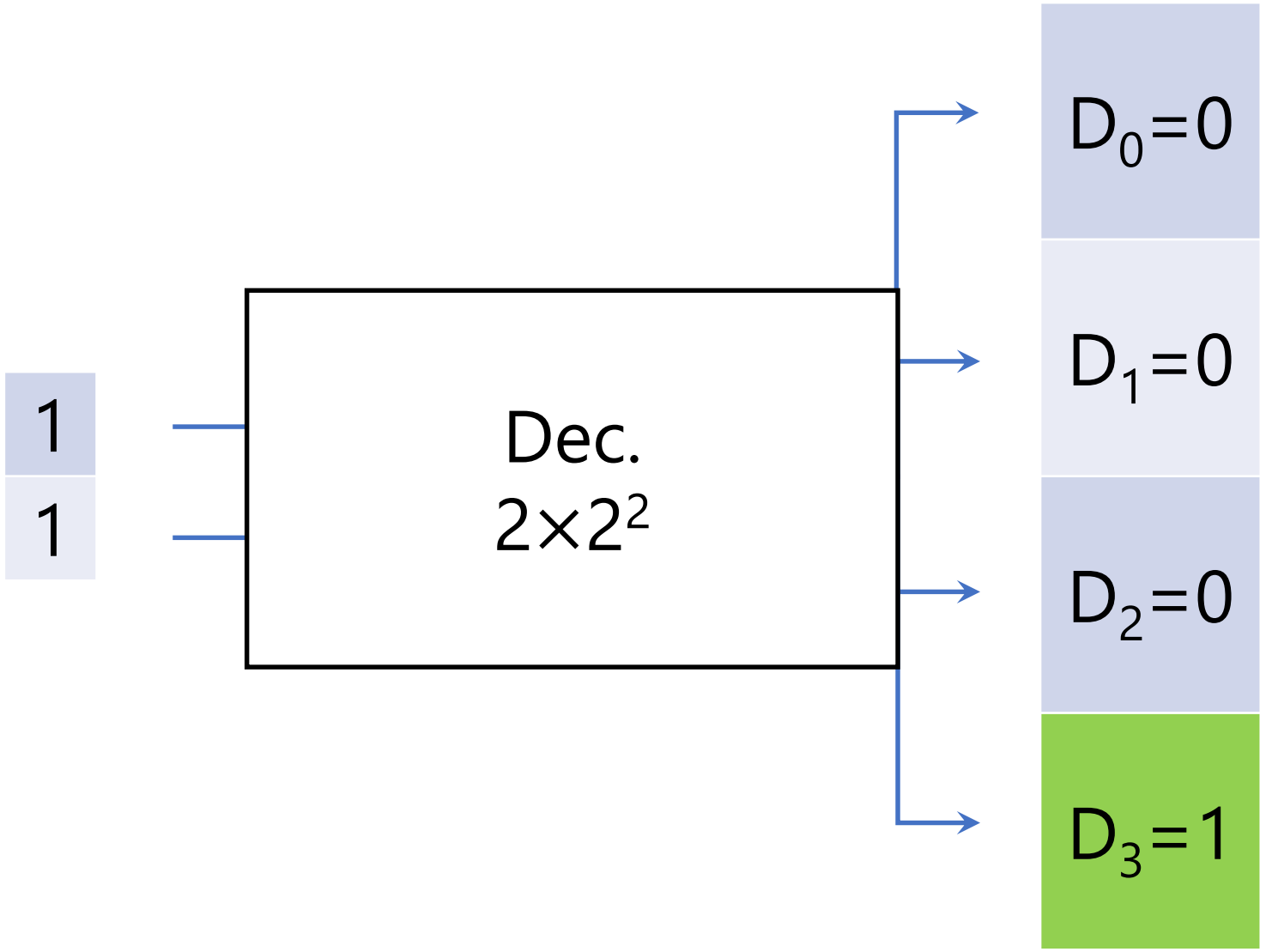

X
D_0
D_1


X
Y
Dec.
2×2²
D0
D1
D2
D3


0
0
Dec.
2×2²
D0=1
D1=0
D2=0
D3=0


1
0
Dec.
2×2²
D0=0
D1=1
D2=0
D3=0


0
1
Dec.
2×2²
D0=0
D1=0
D2=1
D3=0


1
1
Dec.
2×2²
D0=0
D1=0
D2=0
D3=1

Y	X	$D_0=m_0$	$D_1=m_1$	$D_2=m_2$	$D_3=m_3$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

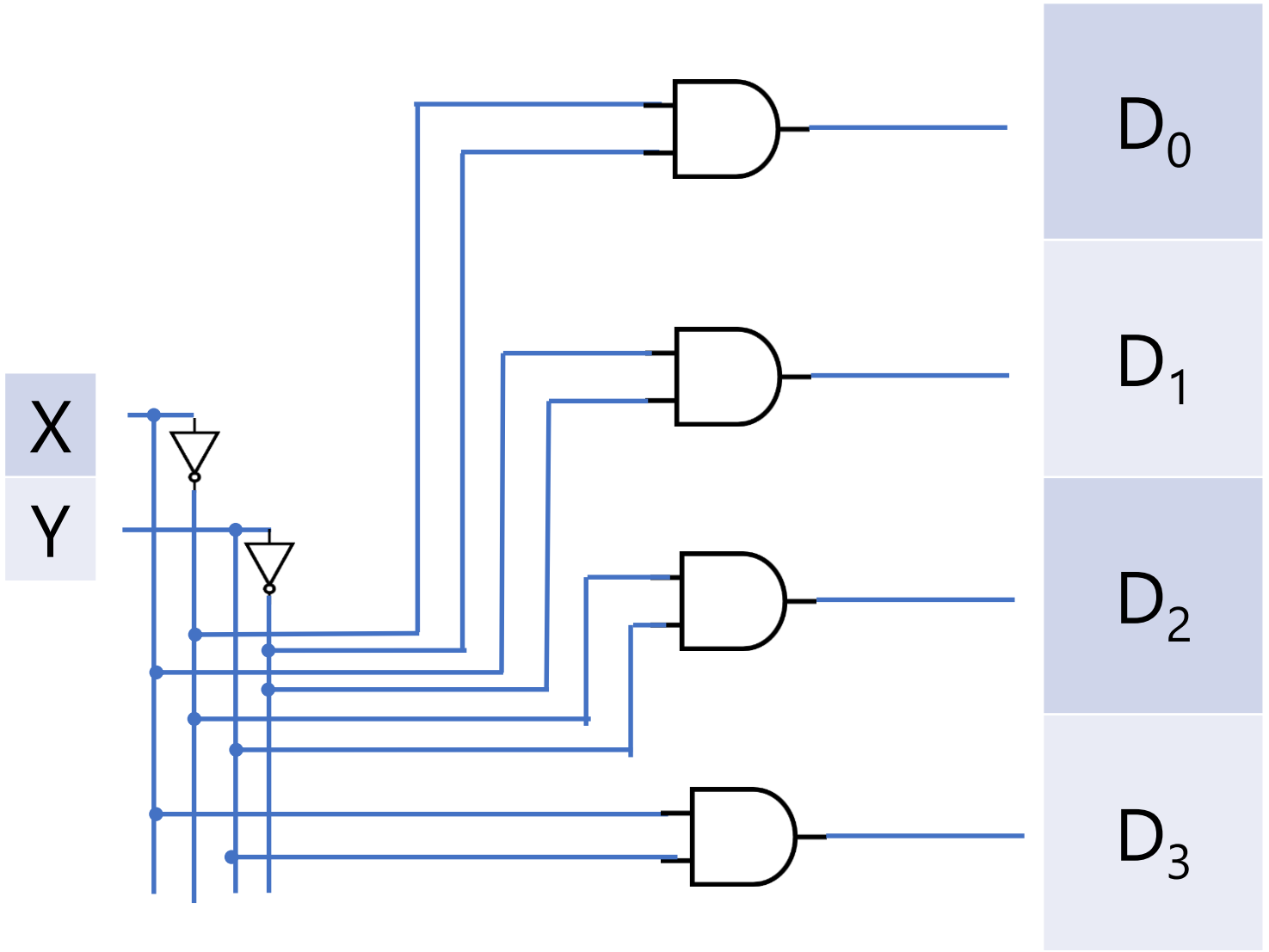

X
Y
D0
D1
D2
D3

Table 4.6
Truth Table of a Three-to-Eight-Line Decoder

Inputs			Outputs							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

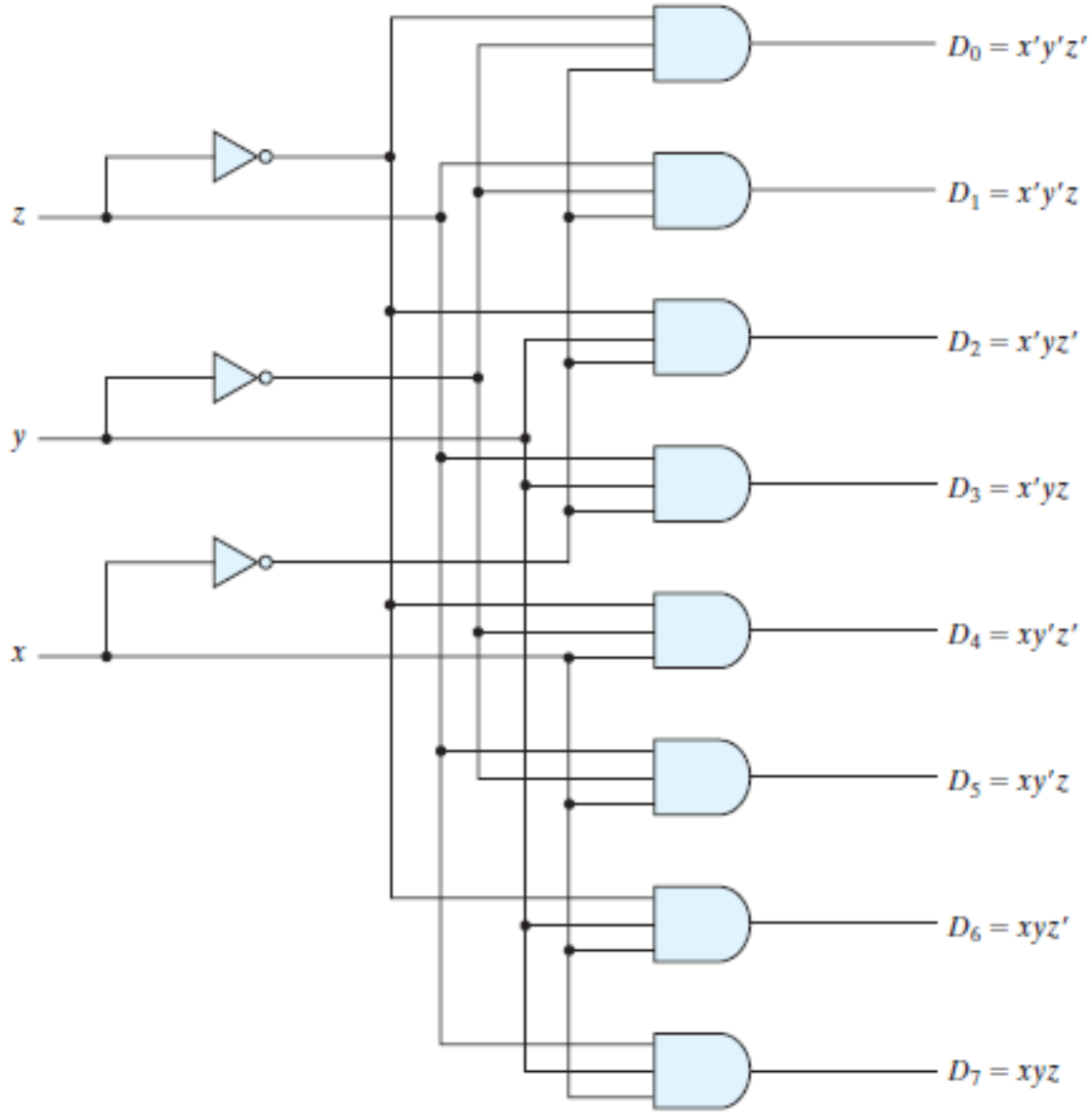


FIGURE 4.18
Three-to-eight-line decoder

Decoder

Decode 4-Bit Binary to 2^4 One-hot

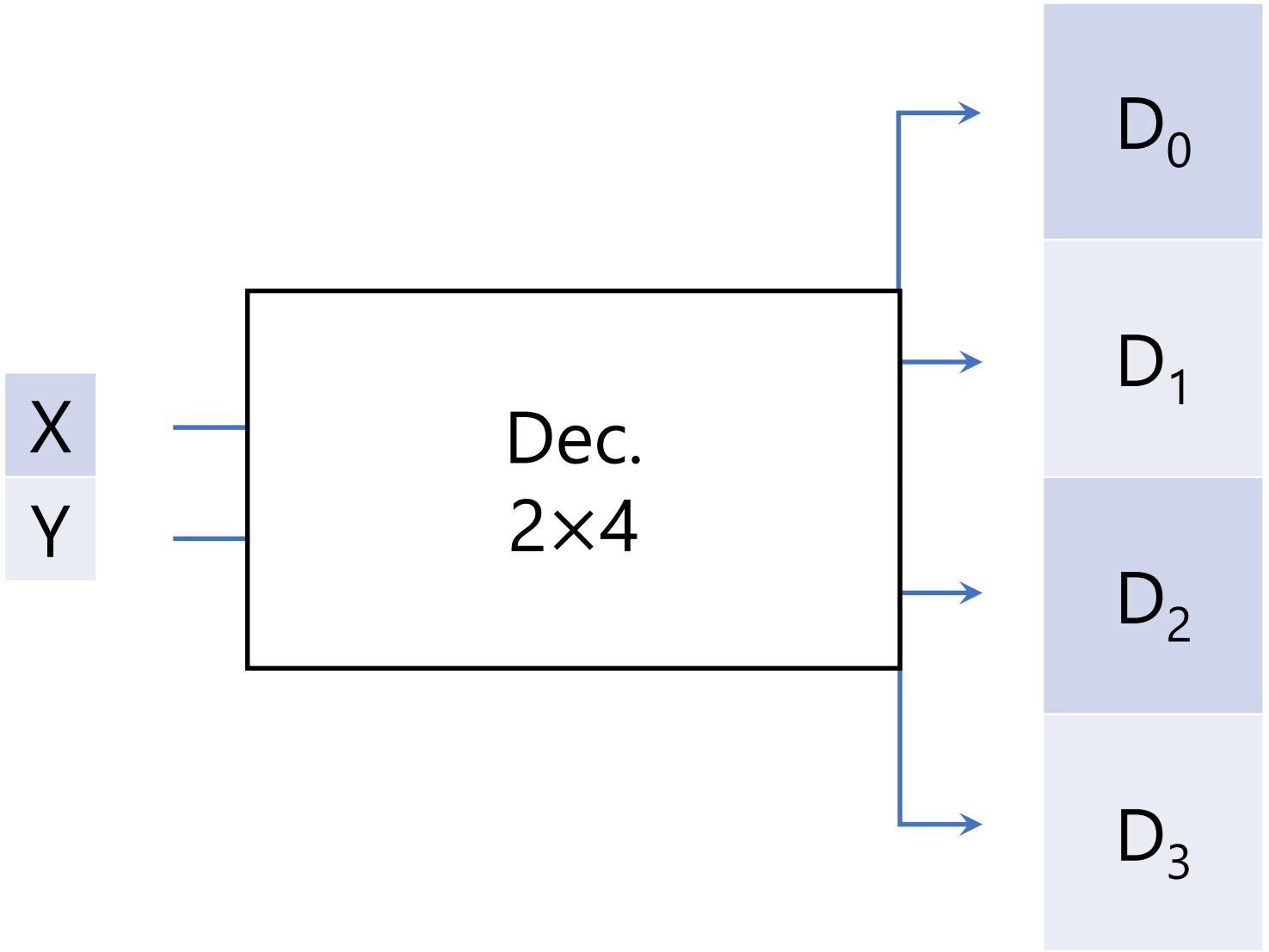
Decoder

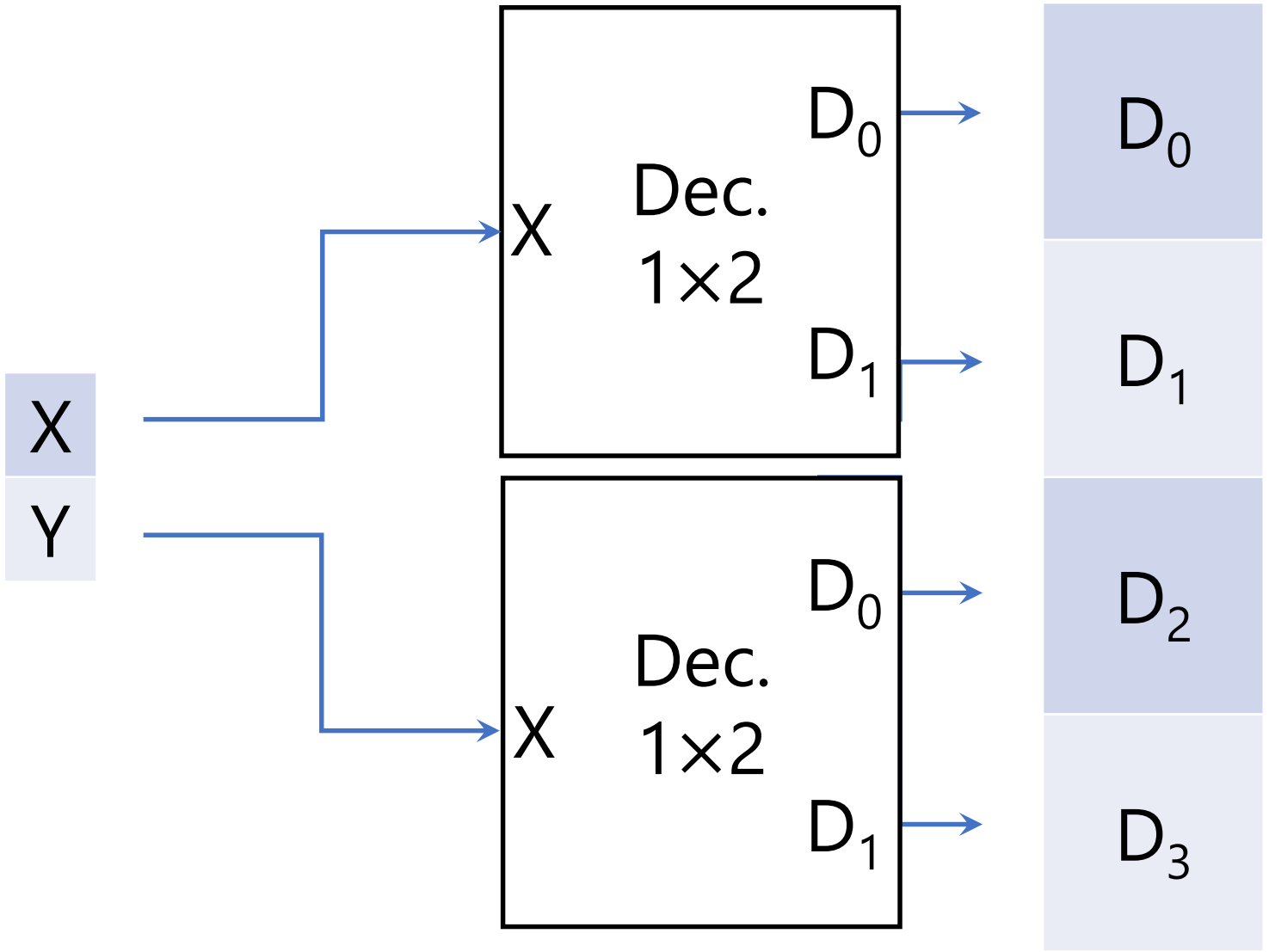
Decode n-Bit Binary to 2^n One-hot

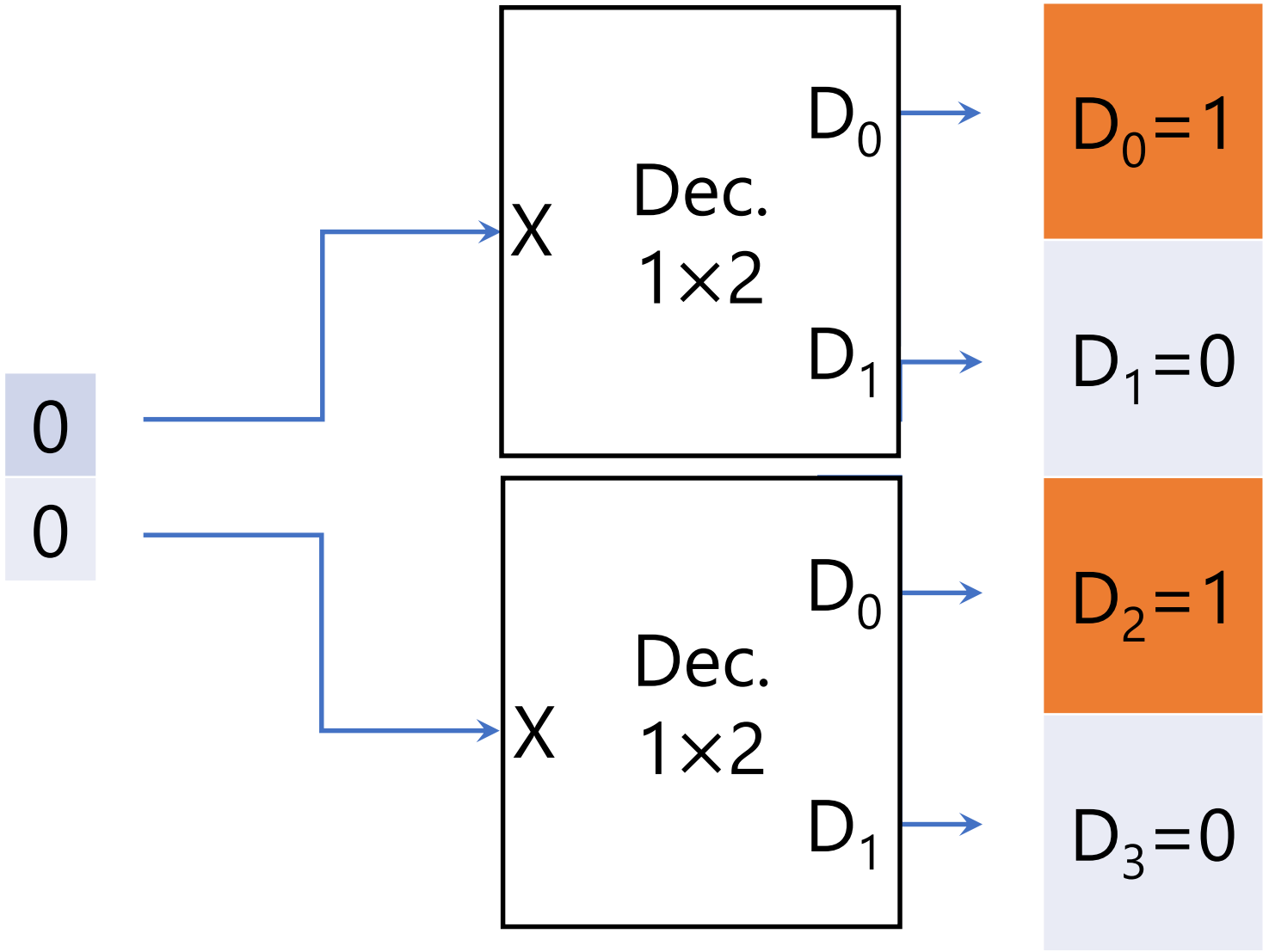
Decoder

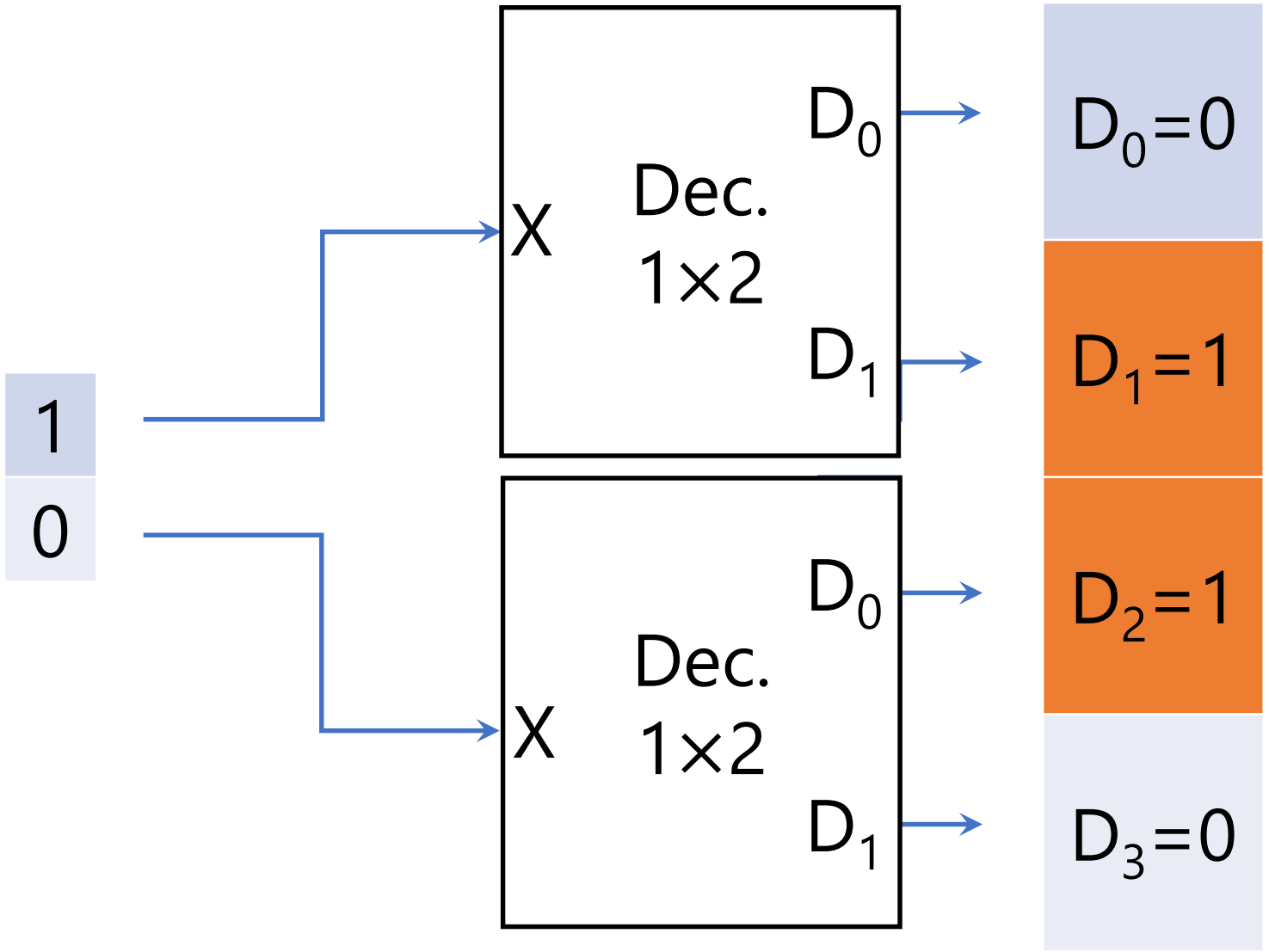
Decode 2-Bit Binary to 2^2 One-hot

Re-Use 1×2^1 Decoder

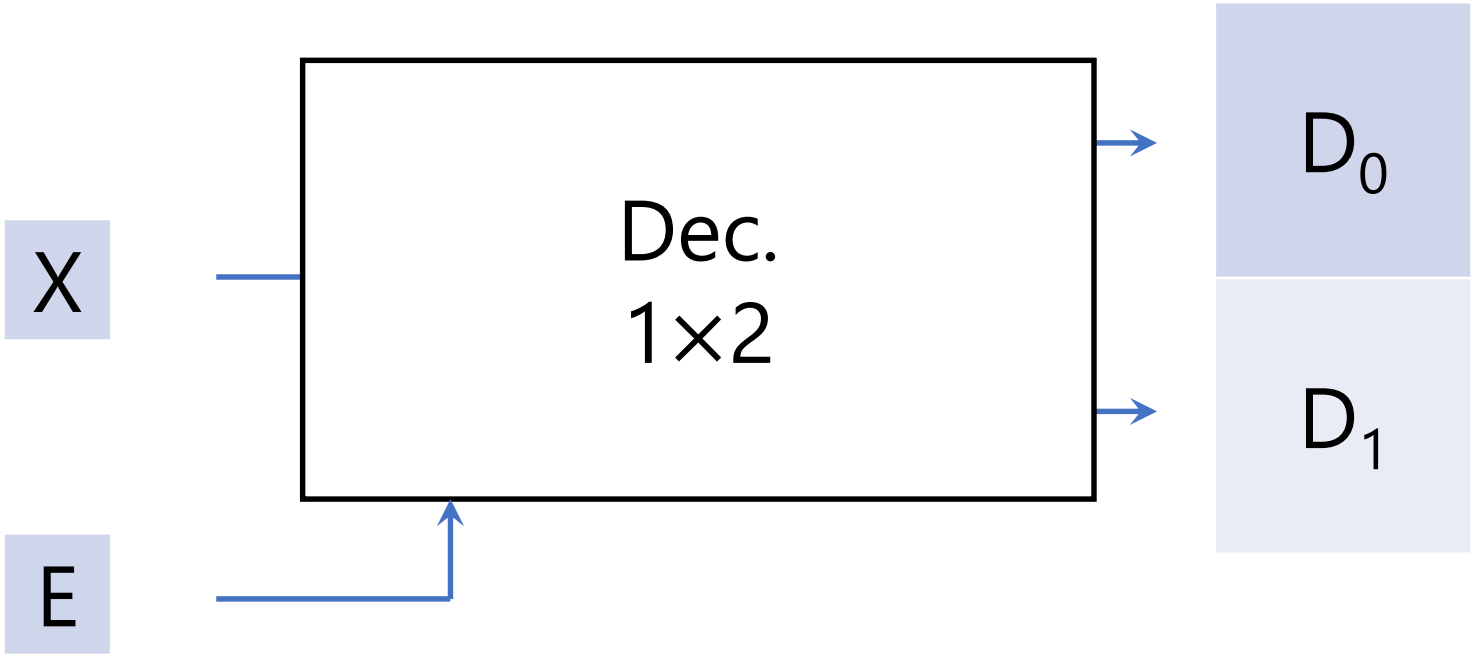

X
Y
Dec.
2×4
D0
D1
D2
D3

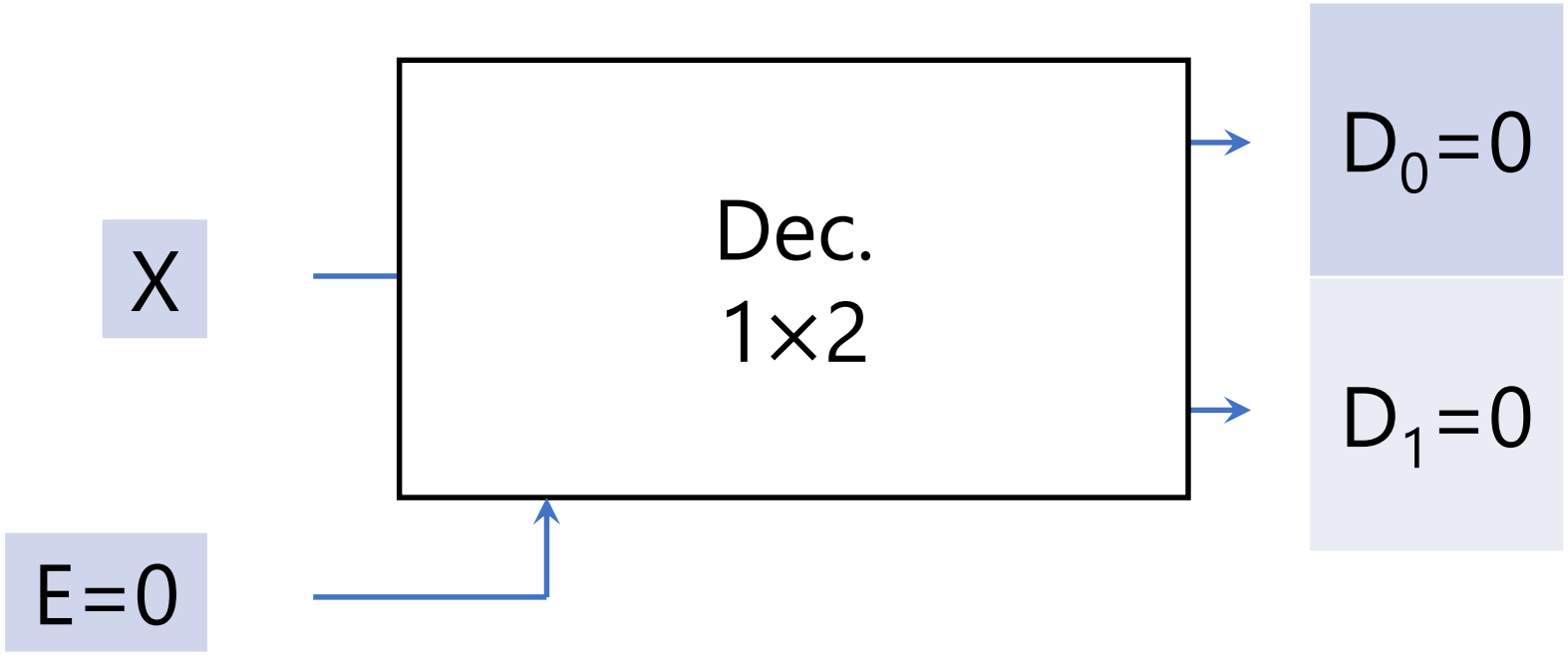

X
Y
X
Dec.
1×2
D_0
D_1
X
Dec.
1×2
D_0
D_1
D_0
D_1
D_2
D_3

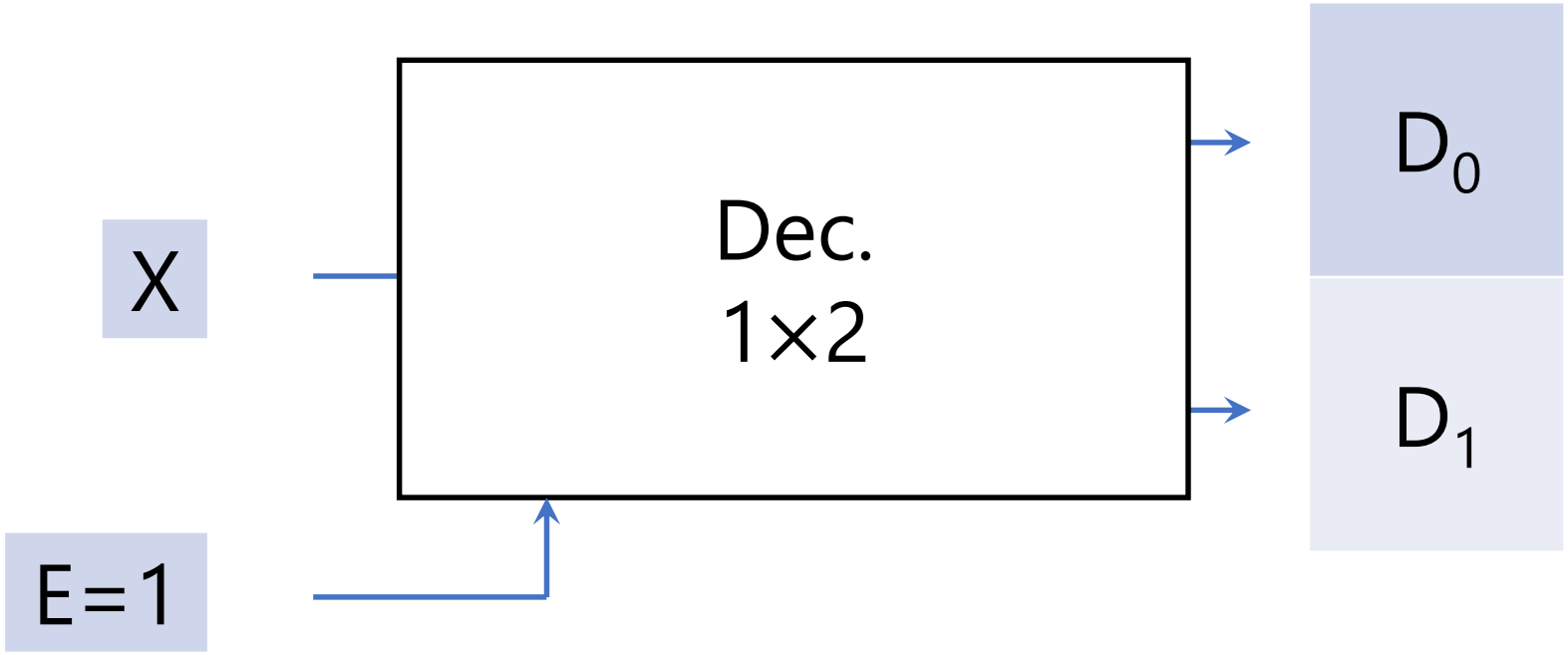

0
0
X
Dec.
1×2
D0
D1
X
Dec.
1×2
D0
D1
D0=1
D1=0
D2=1
D3=0


1
0
X
Dec.
1×2
D_0
D_1
X
Dec.
1×2
D_0
D_1
$D_0=0$
$D_1=1$
$D_2=1$
$D_3=0$

Decoder
Enable input

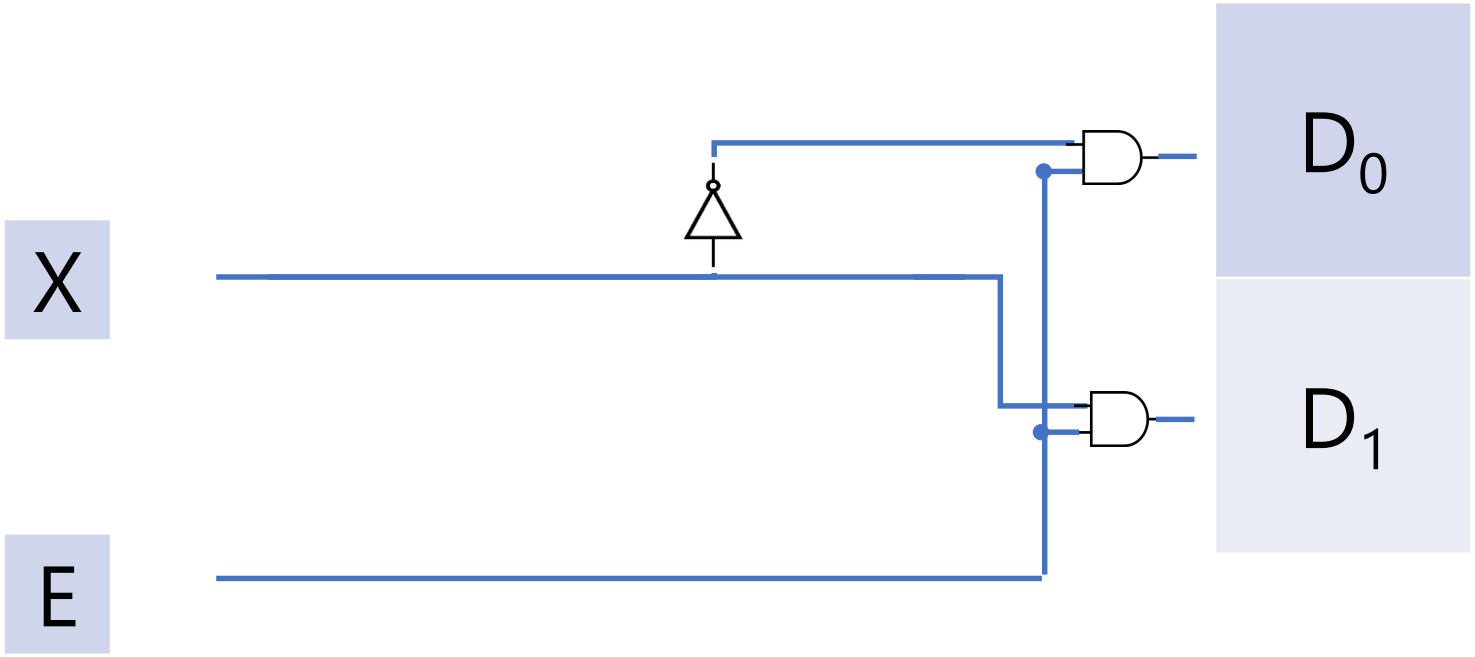

X
E
Dec.
1×2
D0
D1

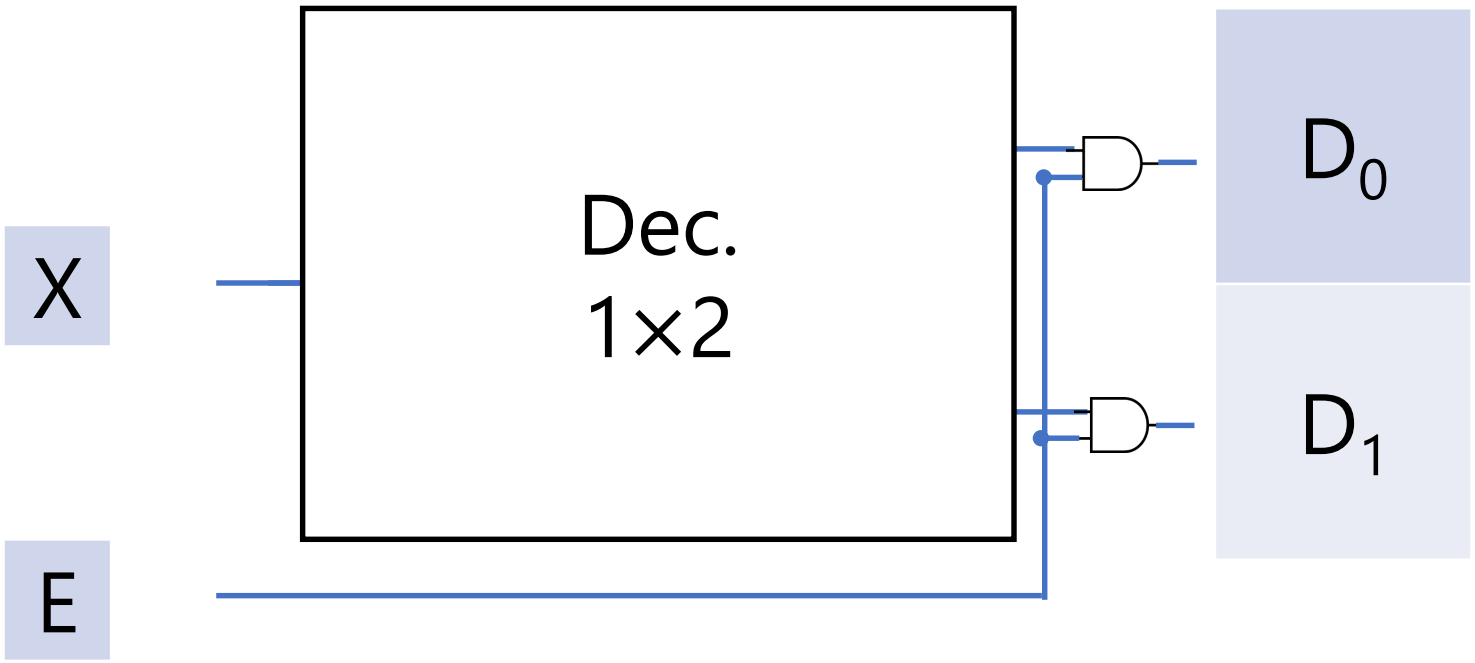

X
E=0
Dec.
1×2
D0=0
D1=0

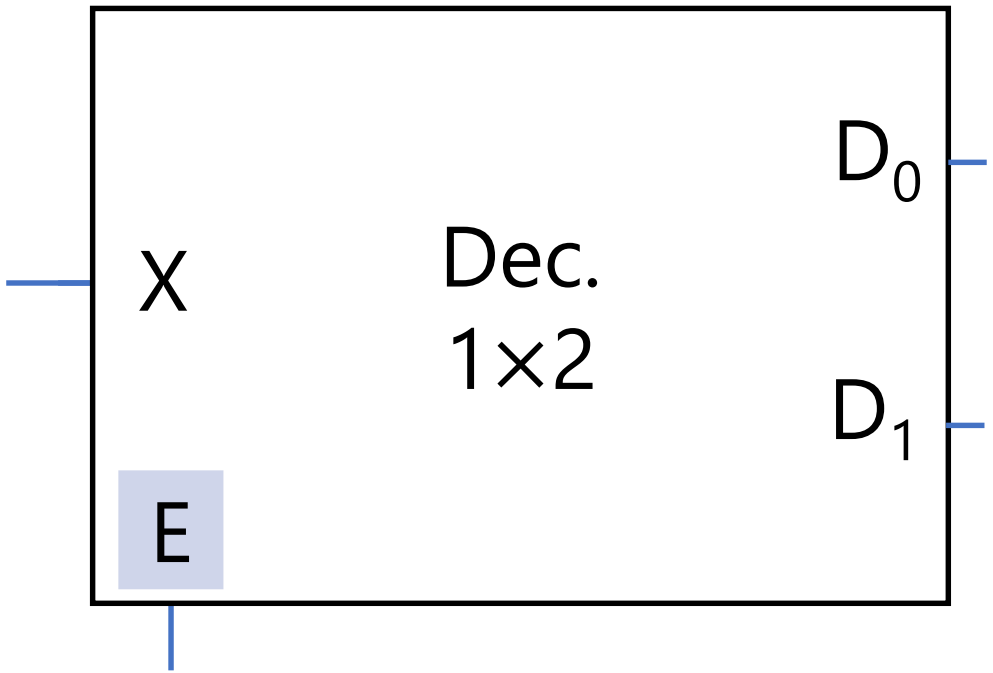

X
E=1
Dec.
1×2
D0
D1

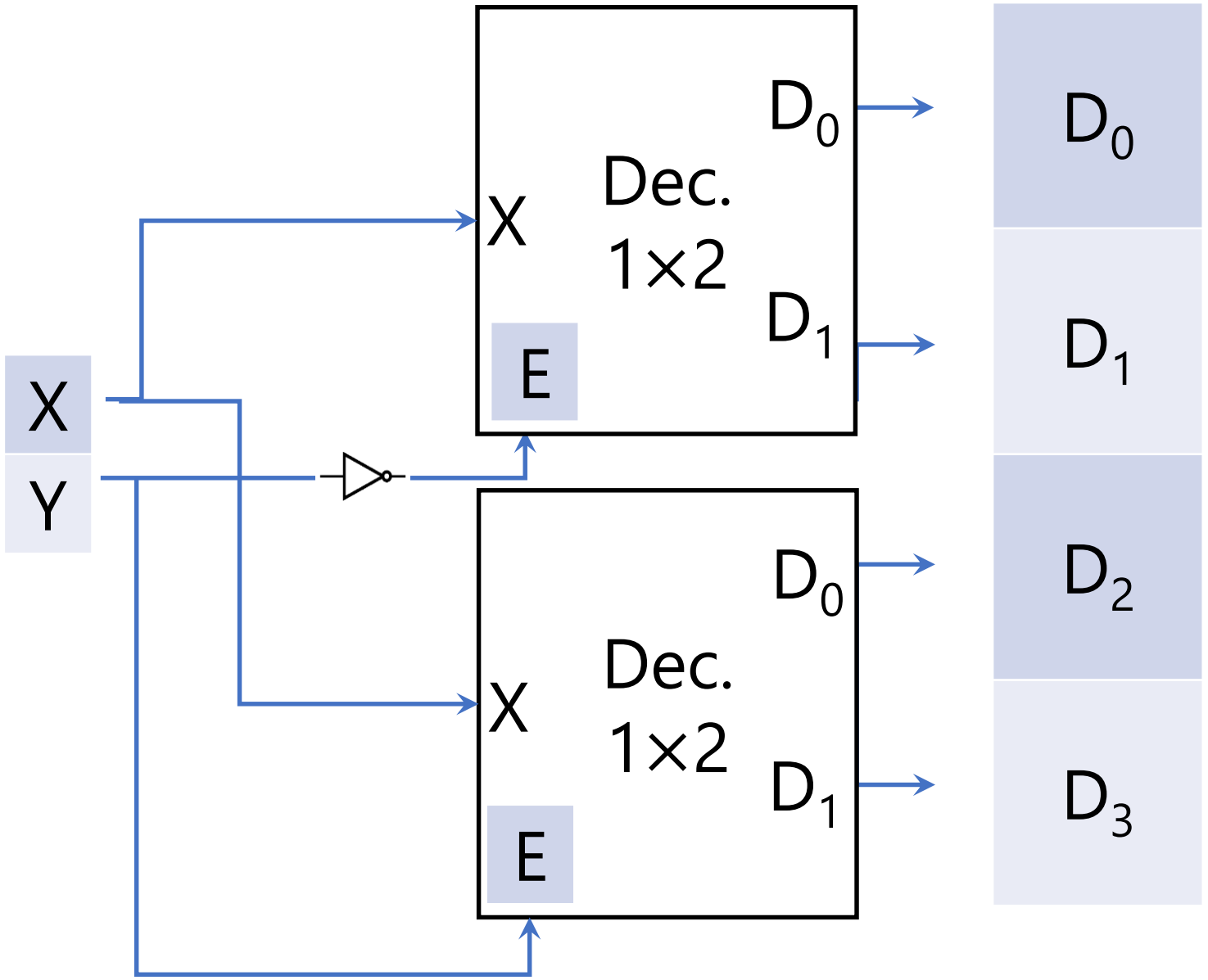
E	X	$D_0=m_2$	$D_1=m_3$
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

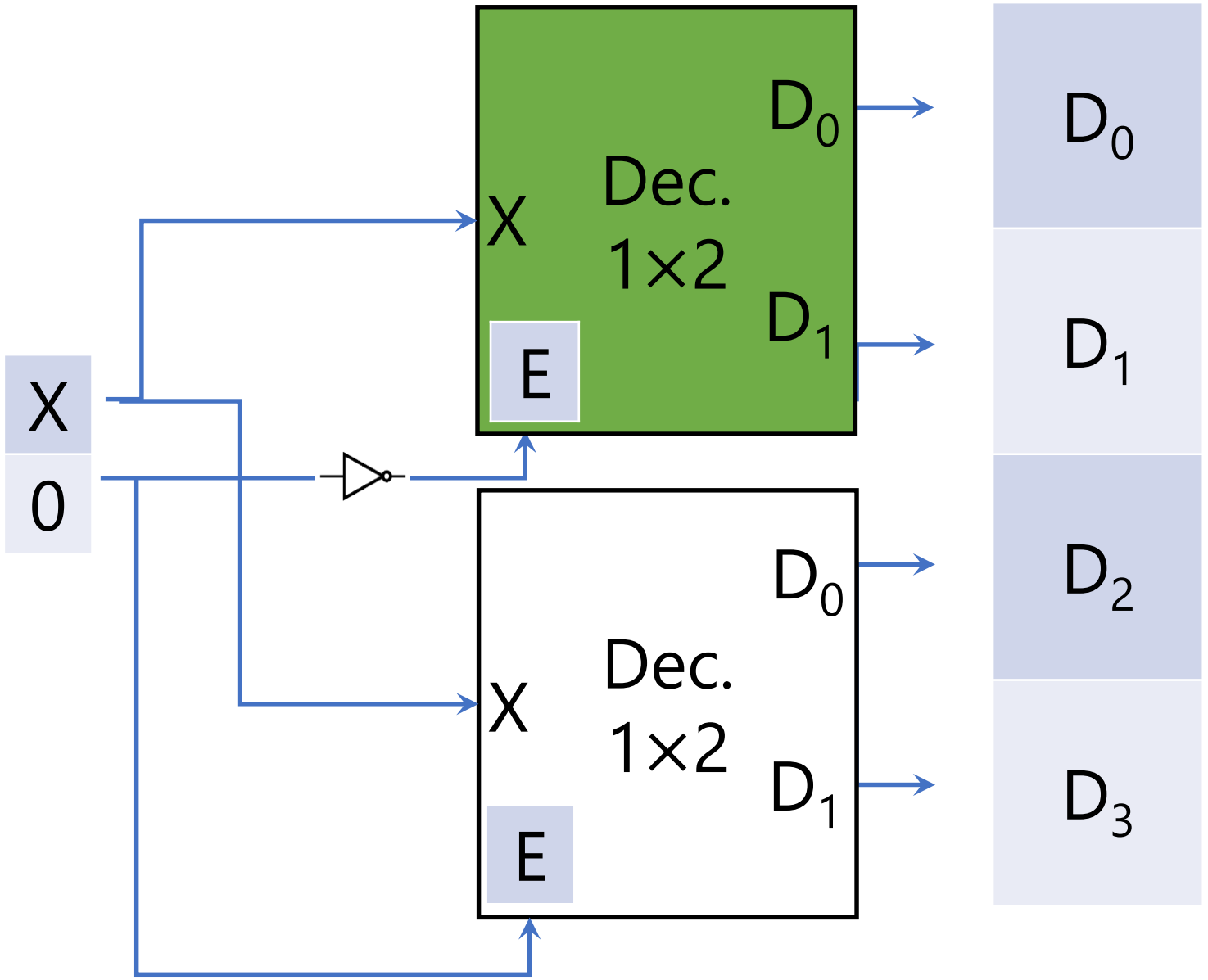
E	X	$D_0=m_0$	$D_1=m_1$
1	0	1	0
1	1	0	1

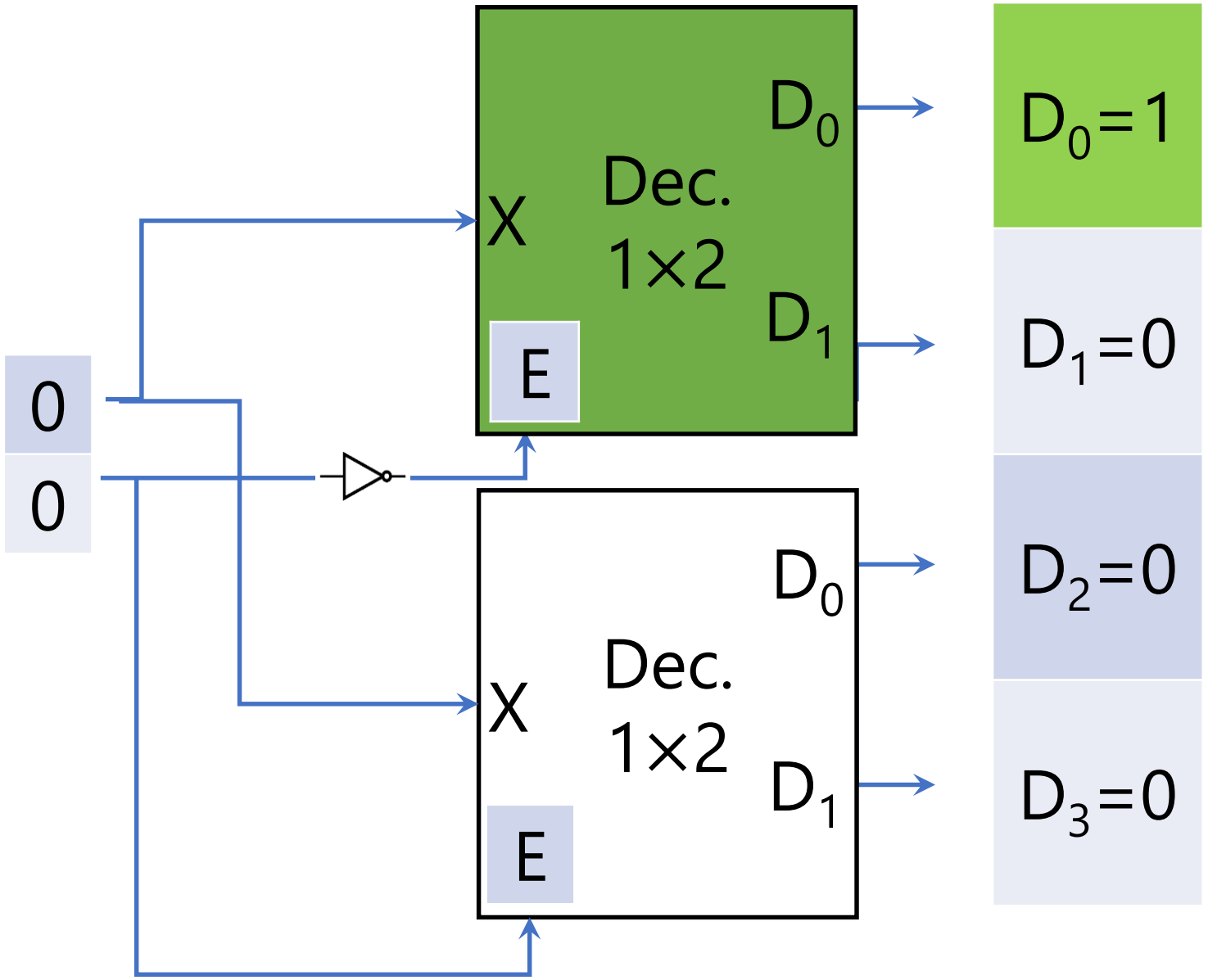

X
E
D_0
D_1

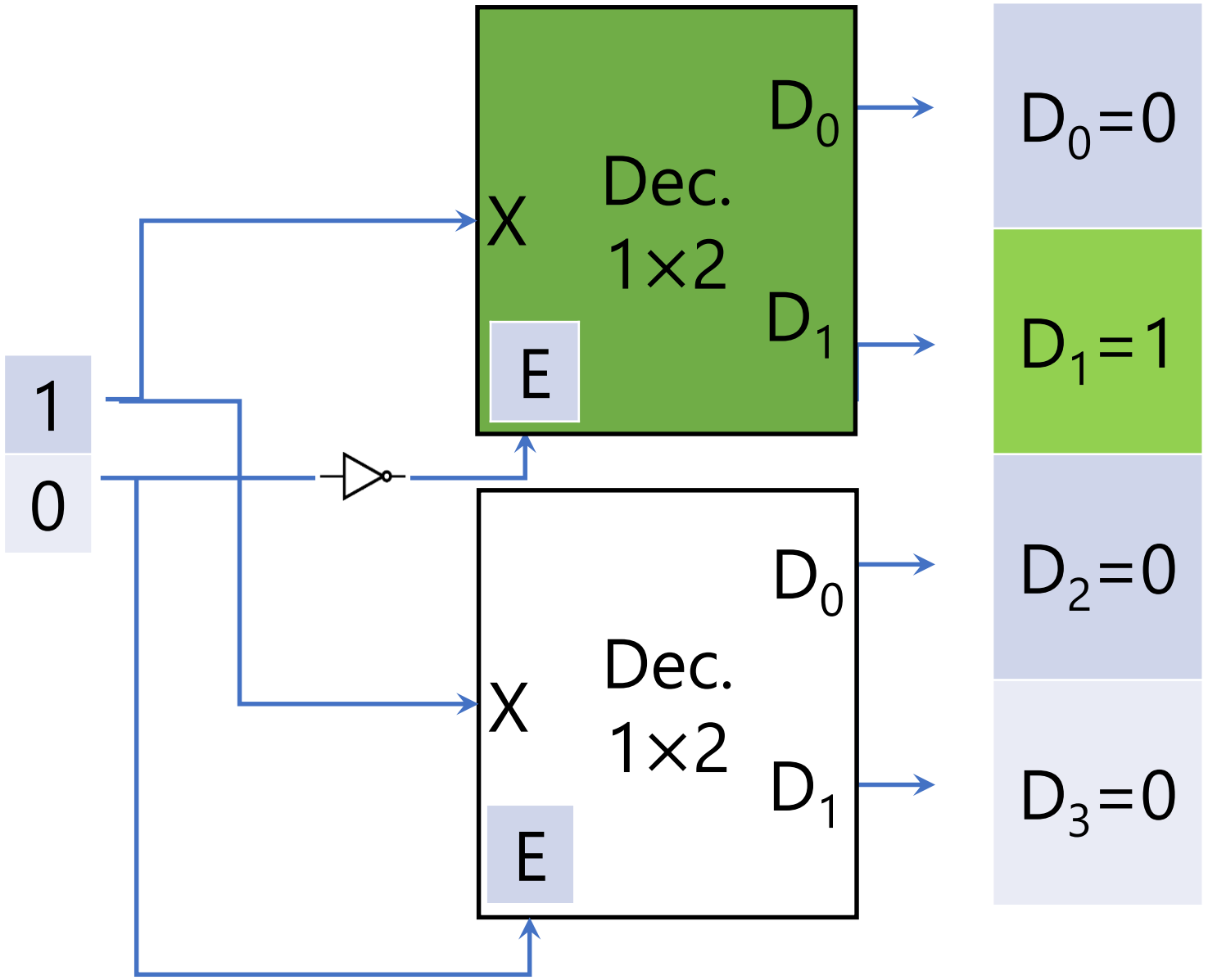

X
E
Dec.
1×2
D0
D1

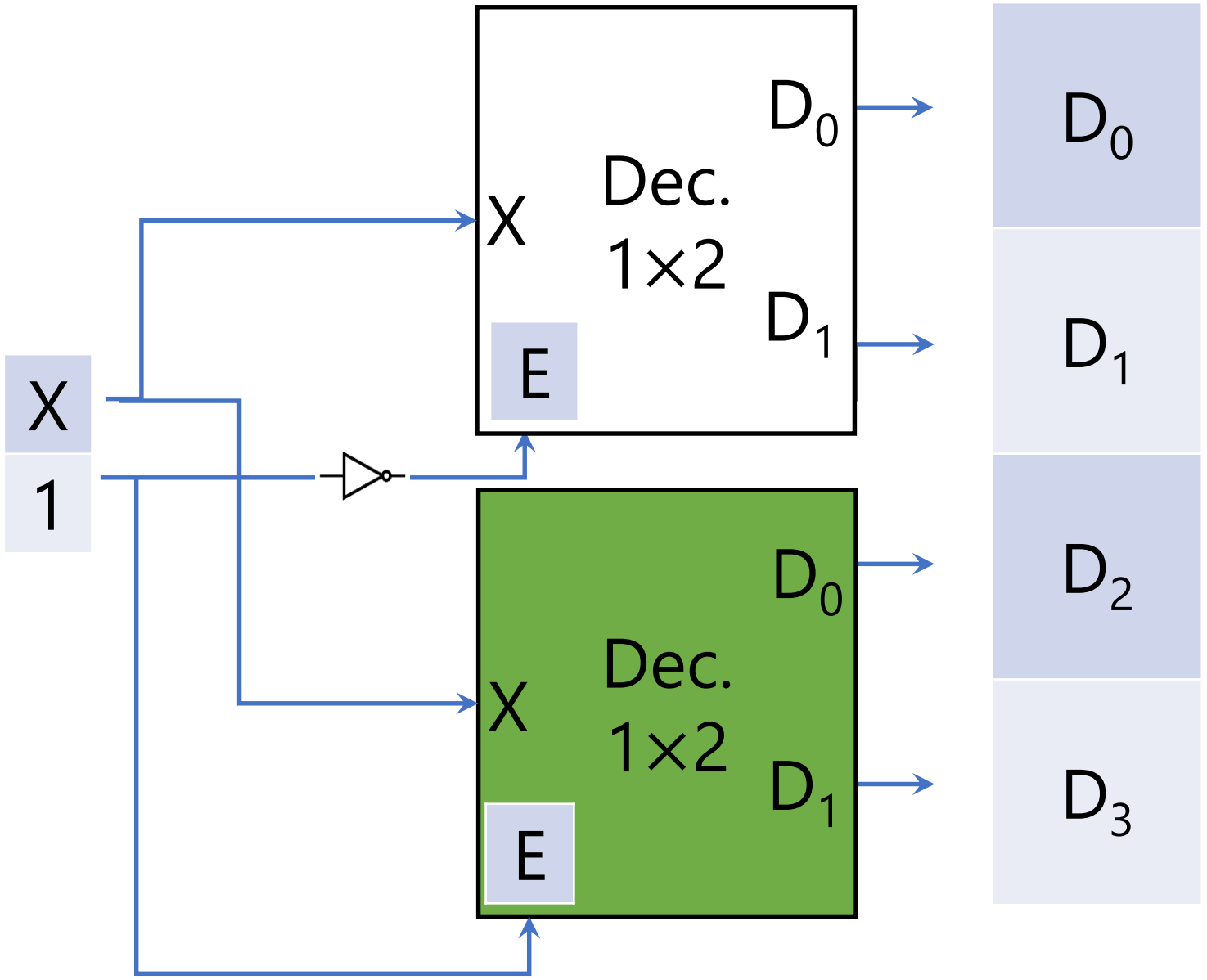

X
Dec.
1×2
E
D0
D1

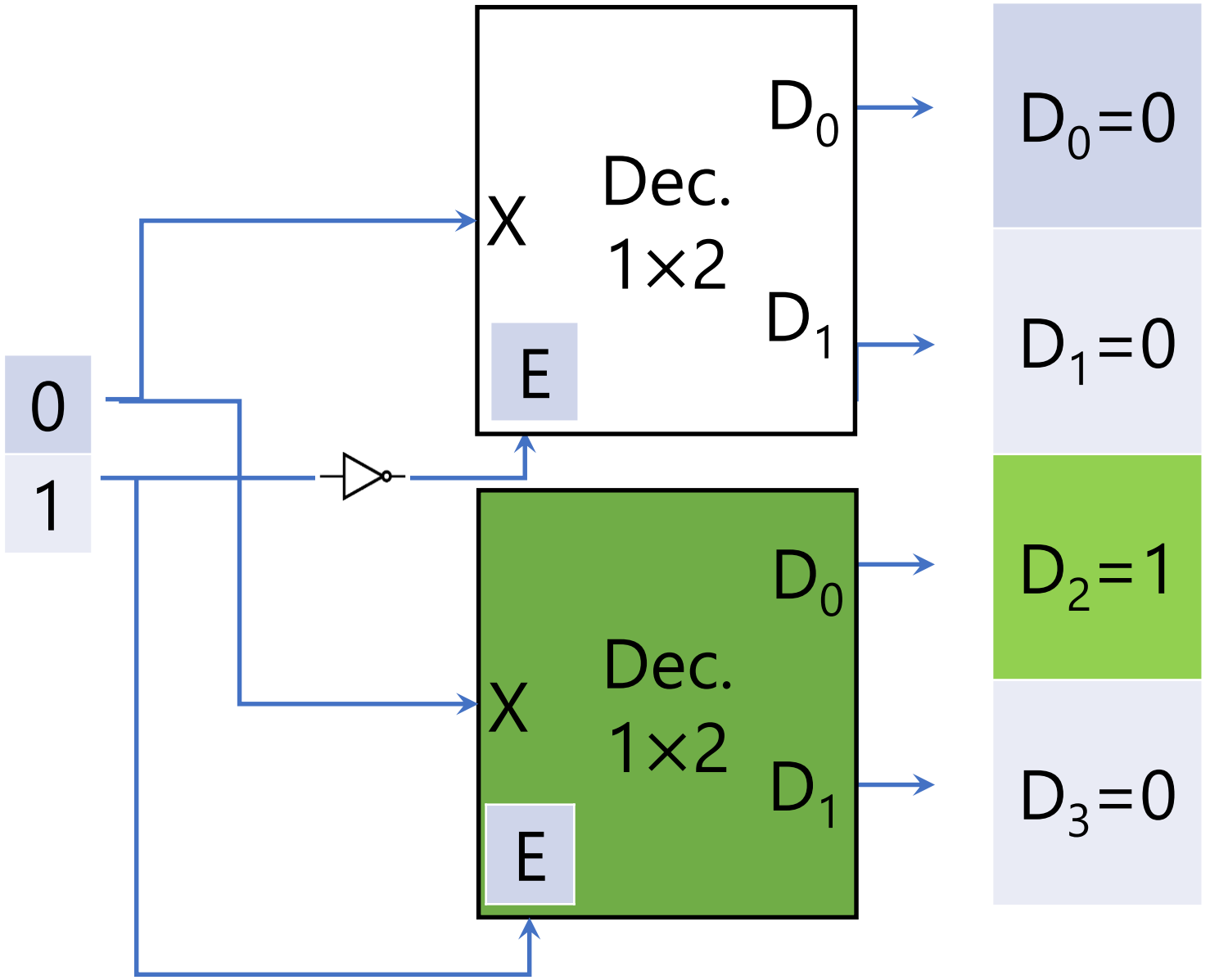

X
Y
Dec.
1×2
X
E
D_0
D_1
Dec.
1×2
X
E
D_0
D_1
D_0
D_1
D_2
D_3

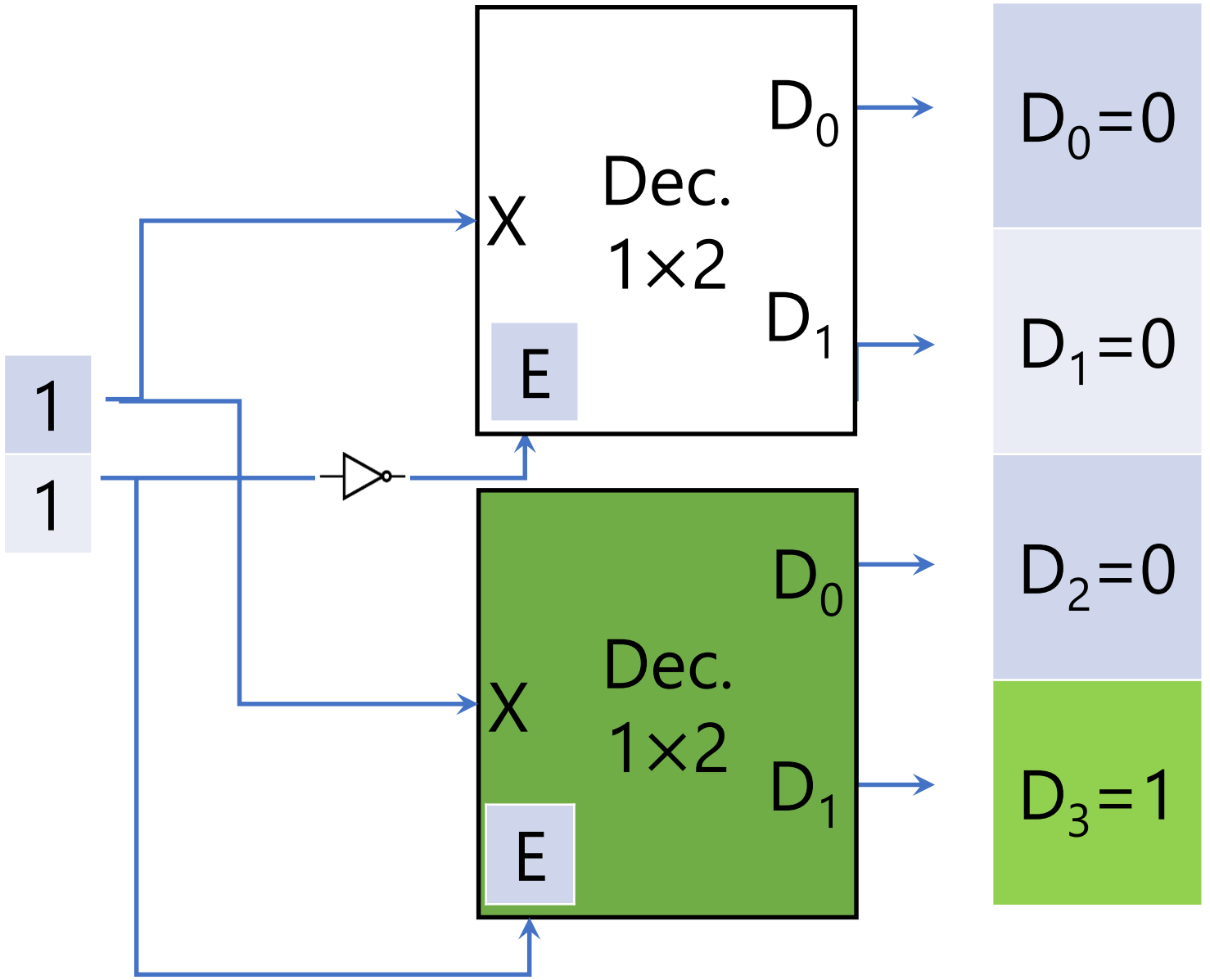

X
0
X
E
Dec.
1×2
D0
D1
X
E
Dec.
1×2
D0
D1
D0
D1
D2
D3

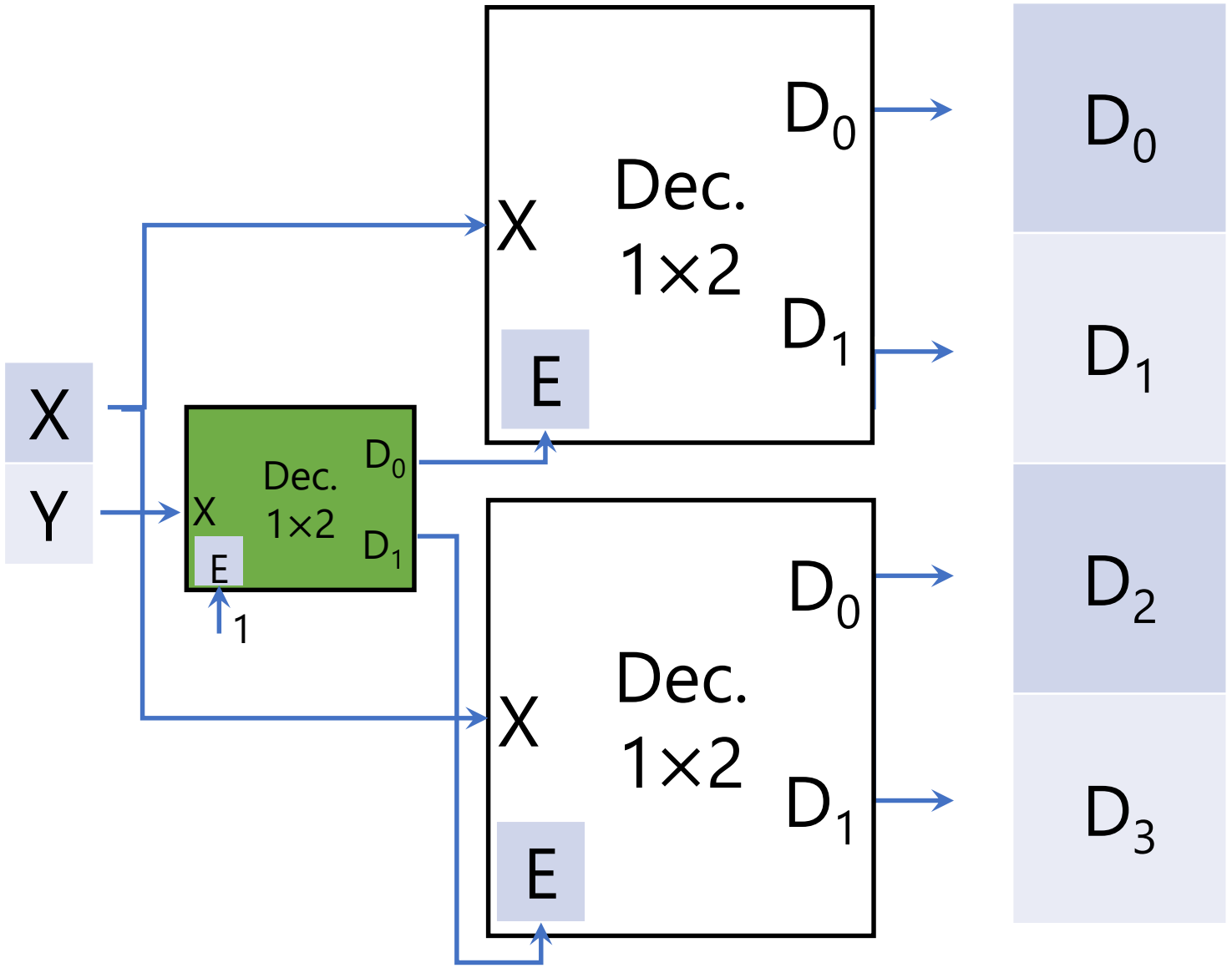

0
0
X
Dec.
1×2
E
D0
D1
X
Dec.
1×2
E
D0
D1
D0=1
D1=0
D2=0
D3=0

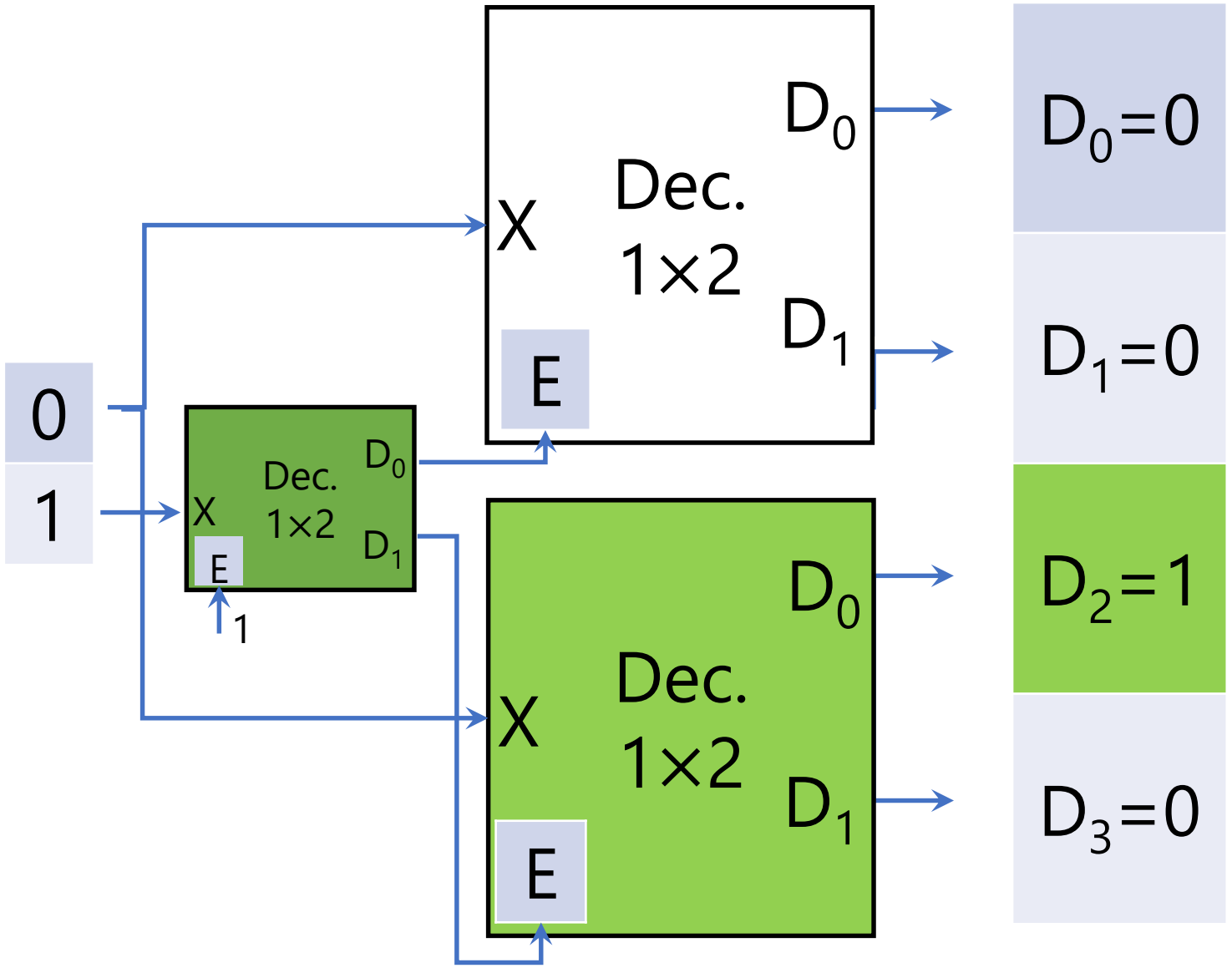

1
0
X
Dec.
1×2
E
D0
D1
X
Dec.
1×2
E
D0
D1
D0=0
D1=1
D2=0
D3=0


X
1
X
E
Dec.
1×2
D0
D1
X
E
Dec.
1×2
D0
D1
D0
D1
D2
D3


0
1
X
Dec.
1×2
E
D0
D1
X
Dec.
1×2
E
D0
D1
D0=0
D1=0
D2=1
D3=0


1
1
X
Dec.
1×2
E
D0
D1
X
Dec.
1×2
E
D0
D1
D0=0
D1=0
D2=0
D3=1

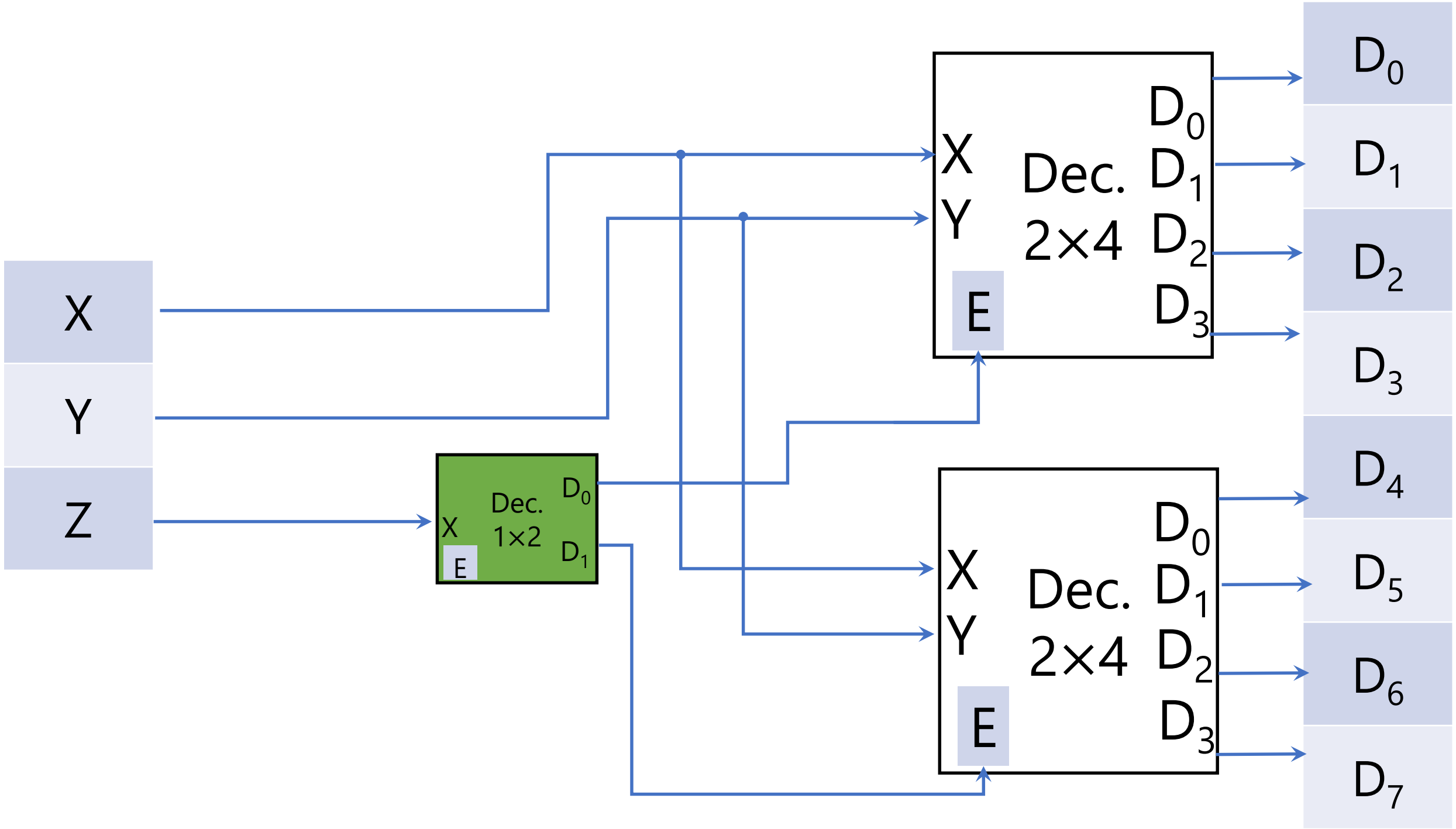

X
Y
Dec.
1×2
X
E
D0
D1
1
Dec.
1×2
X
E
D0
D1
Dec.
1×2
X
E
D0
D1
D0
D1
D2
D3

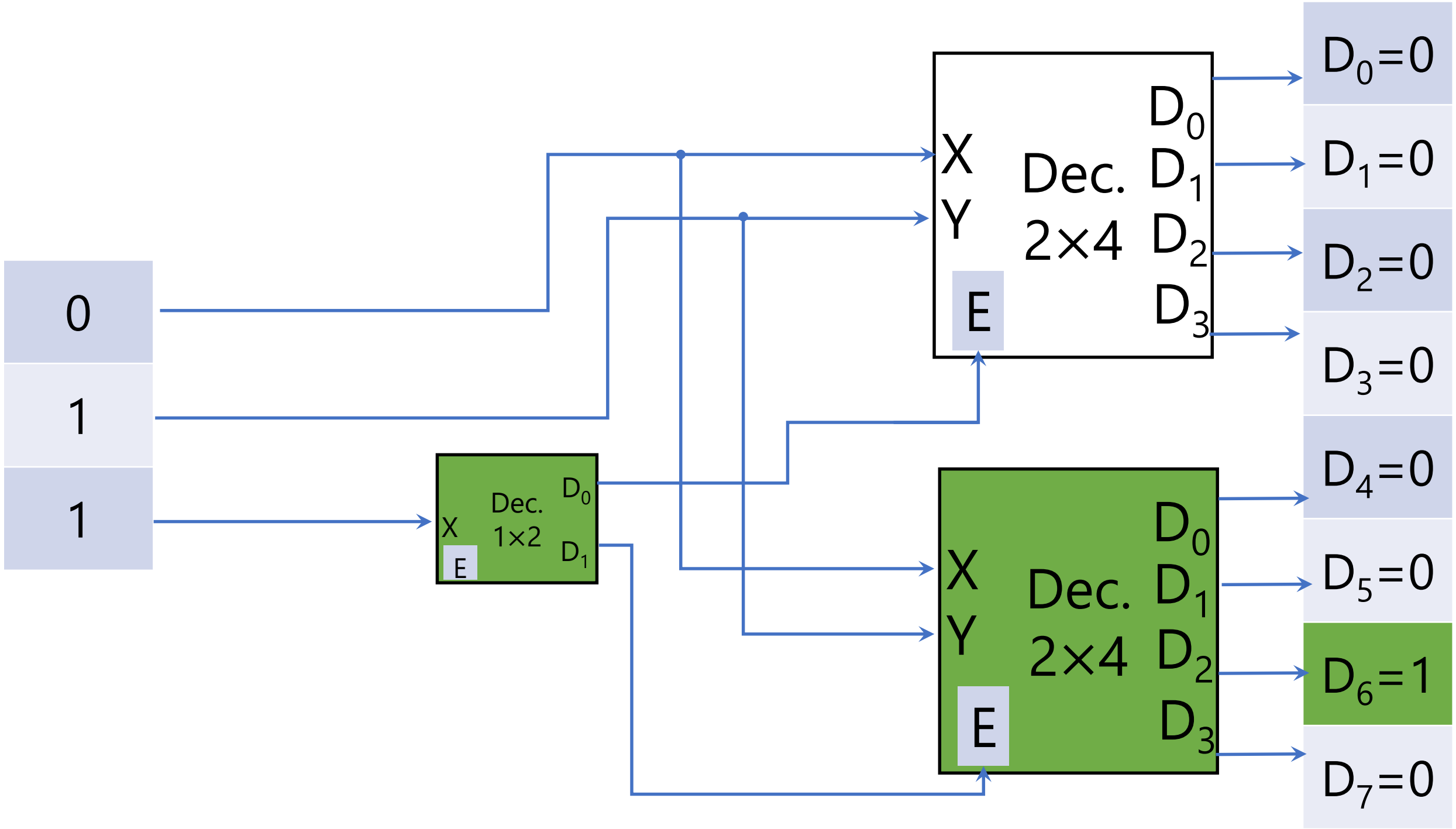

0
1
X
Dec.
1×2
D0
D1
E
1
D0=0
D1=0
D2=1
D3=0

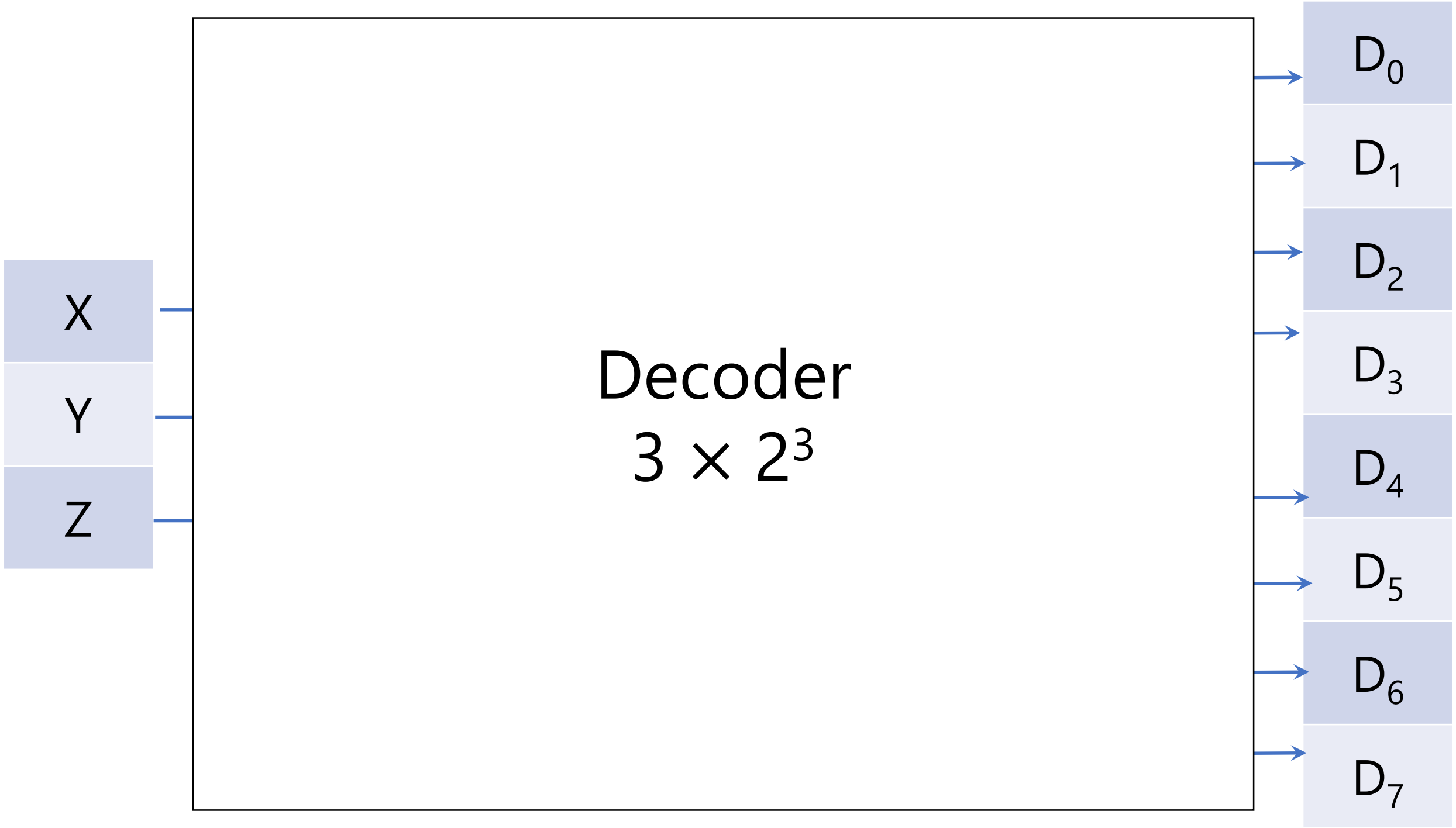
Decoder

Decode 3-Bit Binary to 2^3 One-hot

Re-Use 2×2^2 Decoder


X
Y
Z
Dec.
1×2
X
E
D0
D1
X
Y
Dec.
2×4
E
D0
D1
D2
D3
X
Y
Dec.
2×4
E
D0
D1
D2
D3
D0
D1
D2
D3
D4
D5
D6
D7


0
1
1
X
Y
E
Dec.
2×4
D0
D1
D2
D3
X
E
Dec.
1×2
D0
D1
X
Y
E
Dec.
2×4
D0
D1
D2
D3
D0=0
D1=0
D2=0
D3=0
D4=0
D5=0
D6=1
D7=0


X
Y
Z
Decoder
3 × 2³
D0
D1
D2
D3
D4
D5
D6
D7

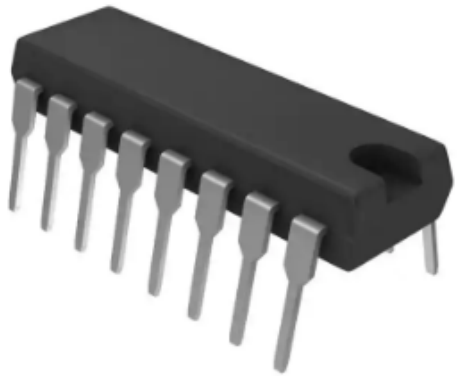
Decoder

Decode 4-Bit Binary to 2^4 One-hot

Re-Use 1×2^1 Decoder

Re-Use 2×2^2 Decoder

Re-Use 3×2^3 Decoder



SN74LS138N

Datasheet

Digi-Key Part Number	296-1639-5-ND
Manufacturer	Texas Instruments
Manufacturer Product Number	SN74LS138N
Supplier	Texas Instruments
Description	IC 3-8 LINE DECODER/DEMUX 16-DIP
Manufacturer Standard Lead Time	6 Weeks
	Decoder/Demultiplexer 1 x 3:8 16-PDIP

Customer Reference

Media & Downloads

RESOURCE TYPE	LINK
Datasheets	SN54LS138, SN54S138, SN74LS138, SN74S138A
Featured Product	Logic Solutions Analog Solutions
PCN Design/Specification	Material Set 30/Mar/2017
EDA / CAD Models	SN74LS138N by SnapEDA SN74LS138N by Ultra Librarian

Price and Procurement

4,043 In Stock
Can ship immediately

QUANTITY

Quantity

Add to Cart

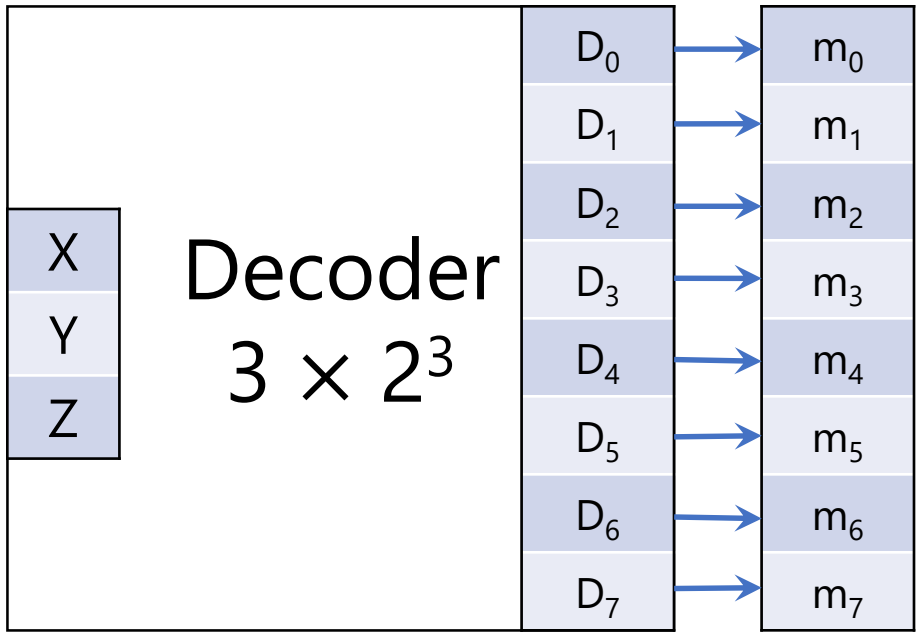
Add to BOM | Add to Favorites

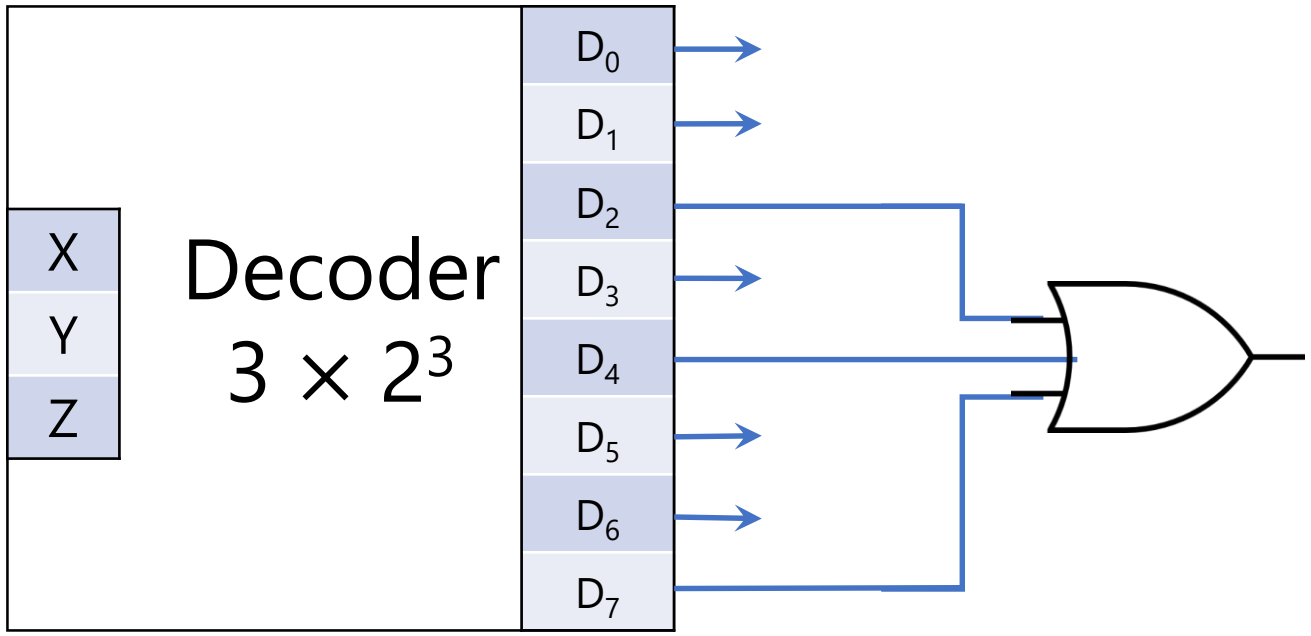
Tube

QTY	UNIT PRICE	EXT PRICE
1	$1.27000	$1.27
10	$1.12000	$11.20
25	$1.05280	$26.32
100	$0.85920	$85.92

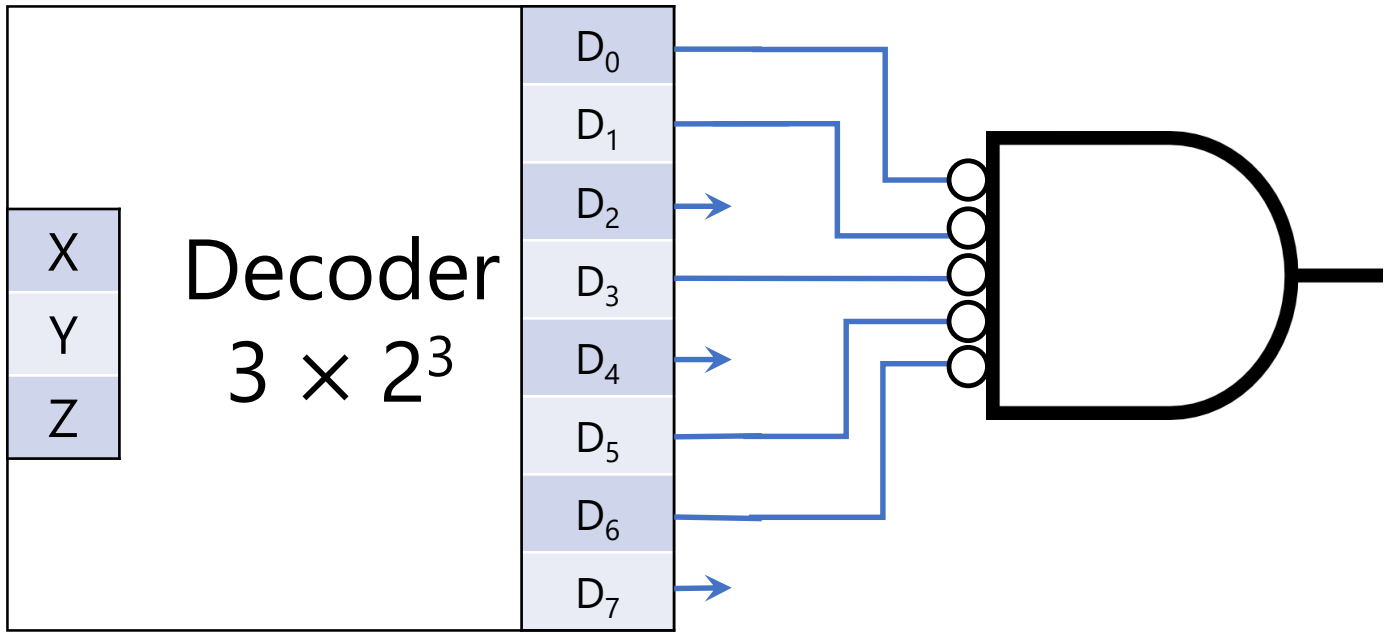
Decoder
Boolean Function

$$F_{SoP} = \sum m(\ldots)$$
$$F_{PoS} = \prod M(\ldots)$$


X
Y
Z
Decoder
3 × 2^3
D0
D1
D2
D3
D4
D5
D6
D7
m0
m1
m2
m3
m4
m5
m6
m7



$$F_{SoP} = \sum m(2,4,7)$$



$$F_{PoS} = \prod M(0,1,3,5,6)$$

Decoder
Full Adder

$$S = \sum m(1,2,4,7)$$
$$C = \sum m(3,5,6,7)$$

C_p	Y	X	$C=\sum m(3,5,6,7)$	$S=\sum m(1,2,4,7)$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

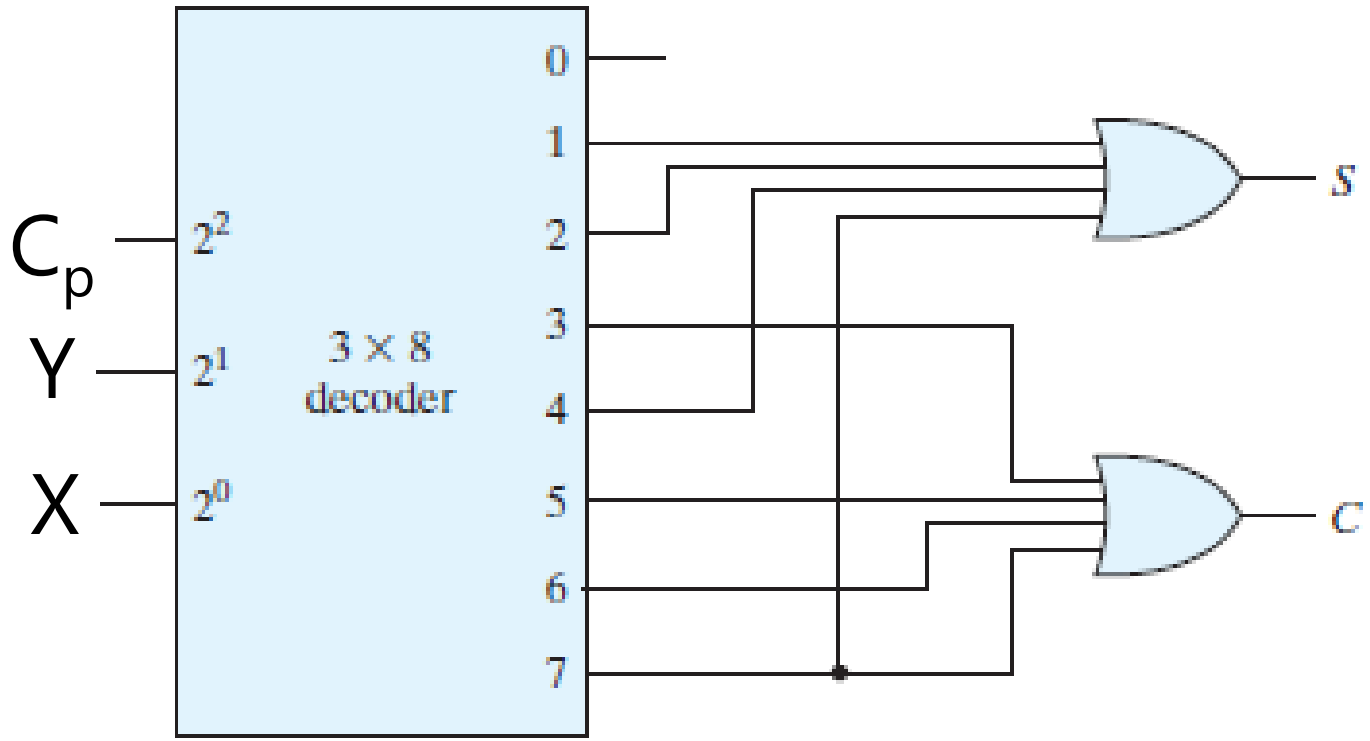
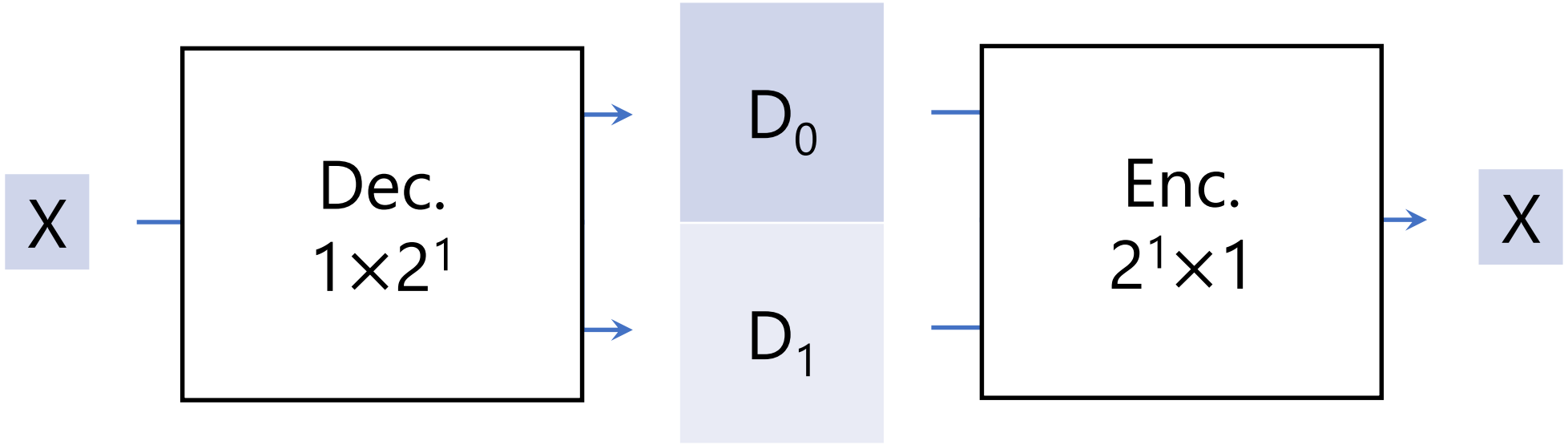
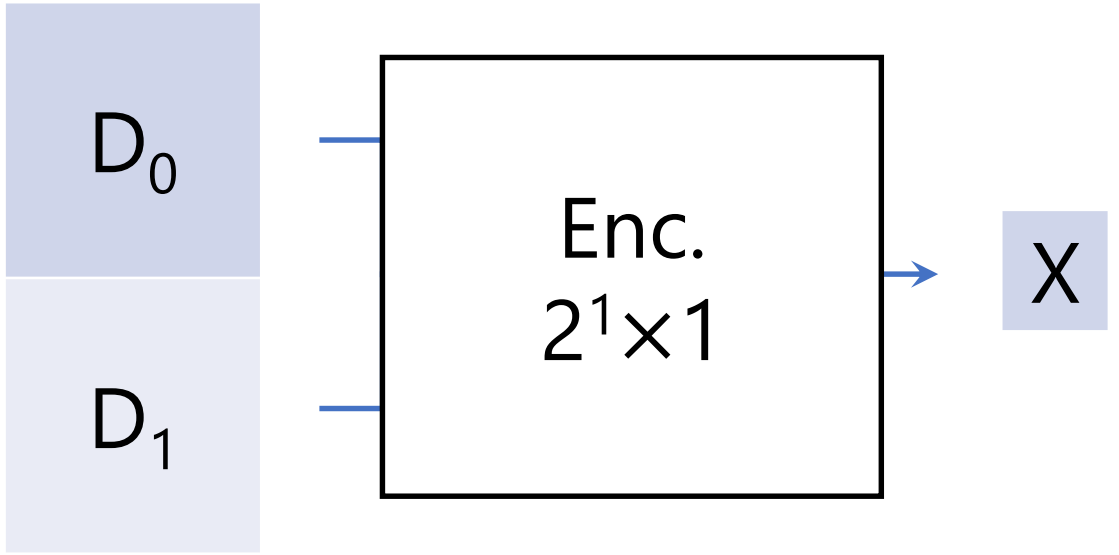


FIGURE 4.21
Implementation of a full adder with a decoder

Encoder

Encoder
1-hot to Binary

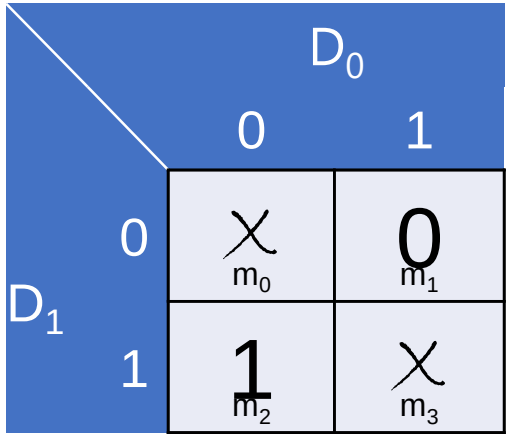

X
Dec.
1×2¹
D0
D1
Enc.
2¹×1
X


D_0
D_1
Enc.
$2^1 \times 1$
X

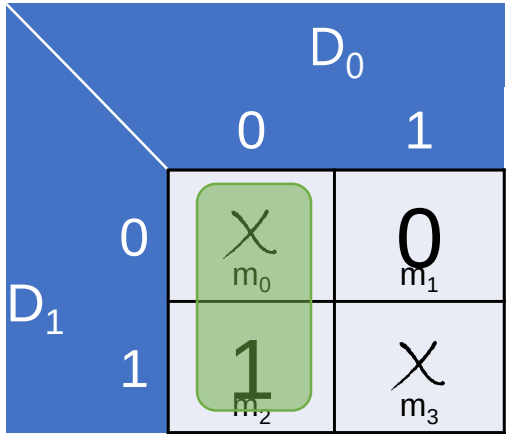
D_1	D_0	F_1
0	0	X
0	1	0
1	0	1
1	1	X

X: Don’t Care Conditions

D_1	D_0	F_1
0	0	X
0	1	0
1	0	1
1	1	X

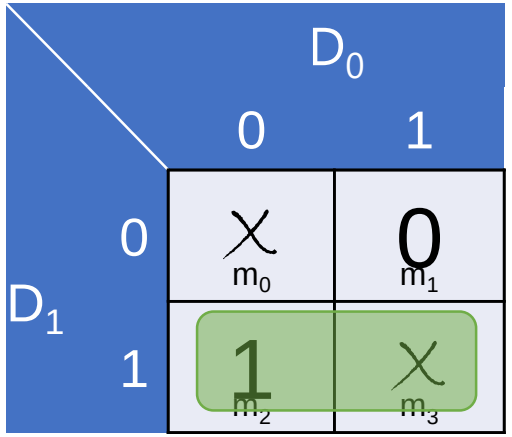


D_1	D_0	F_1
0	0	X
0	1	0
1	0	1
1	1	X



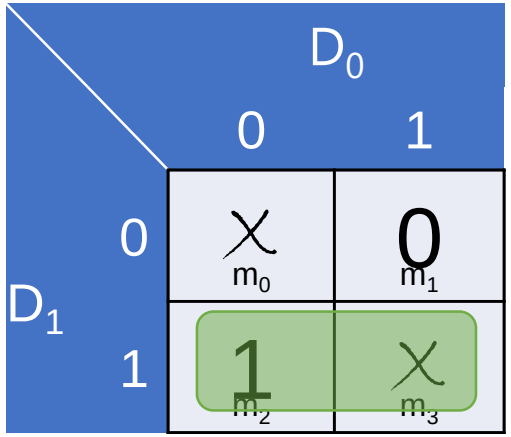
$$F_1 = D'_0$$

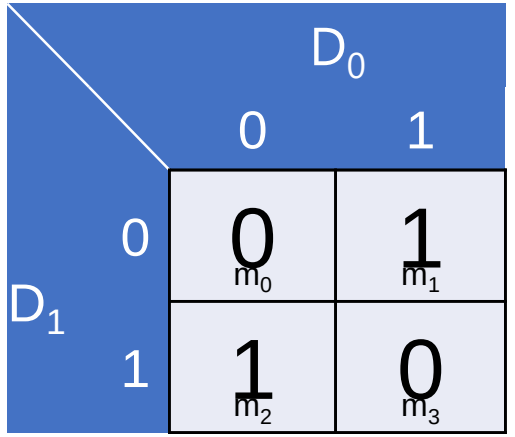
D_1	D_0	F_1
0	0	X
0	1	0
1	0	1
1	1	X



$$F_1 = D_1$$

D_1	D_0	F_1	V
0	0	X	0
0	1	0	1
1	0	1	1
1	1	X	0

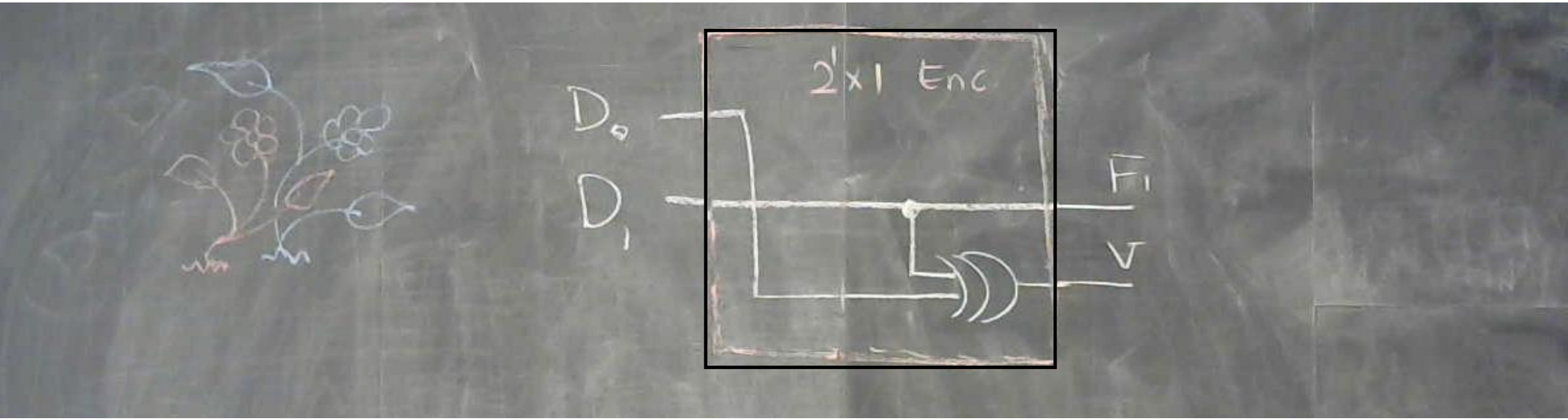


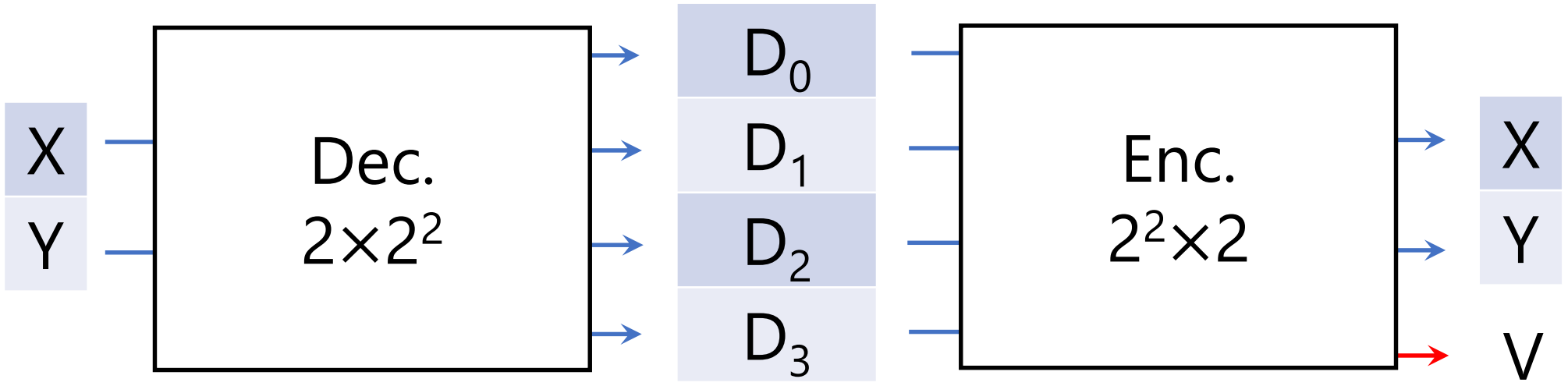


$F_1 = D_1$

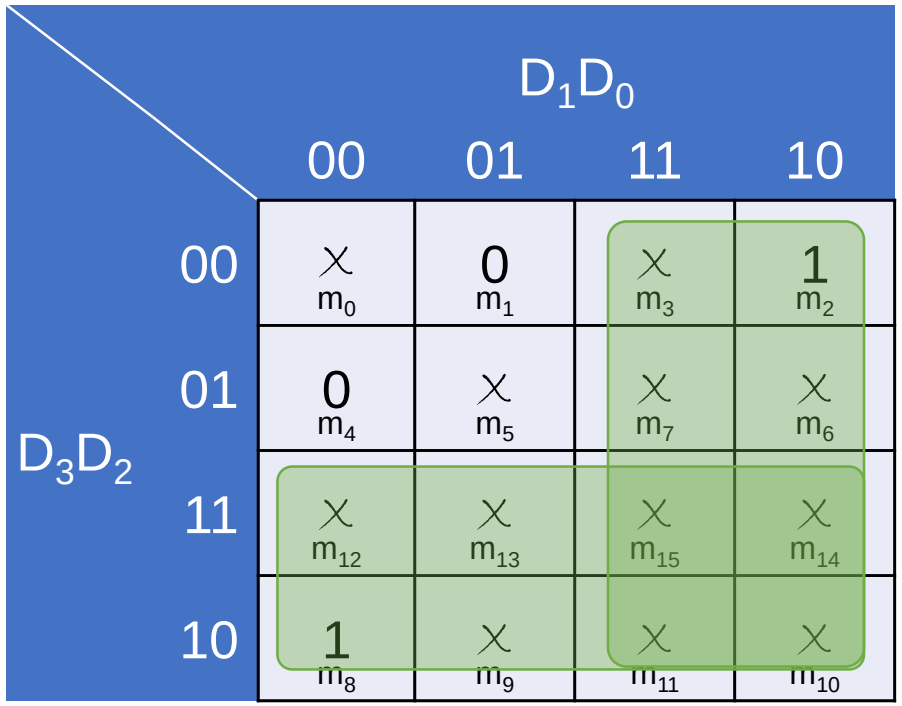
$V = D_0D'_1 + D'_0D_1$

$= D_0 \oplus D_1$

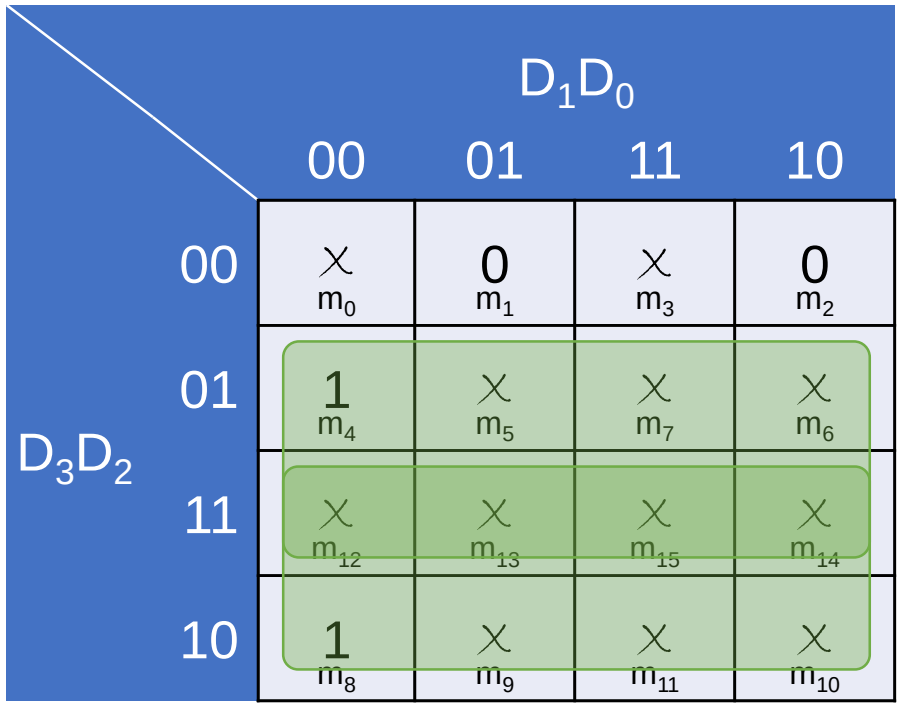

2'x1 Enc
D0
D1
F1
V


X
Y
Dec.
2×2²
D0
D1
D2
D3
Enc.
2²×2
X
Y
V

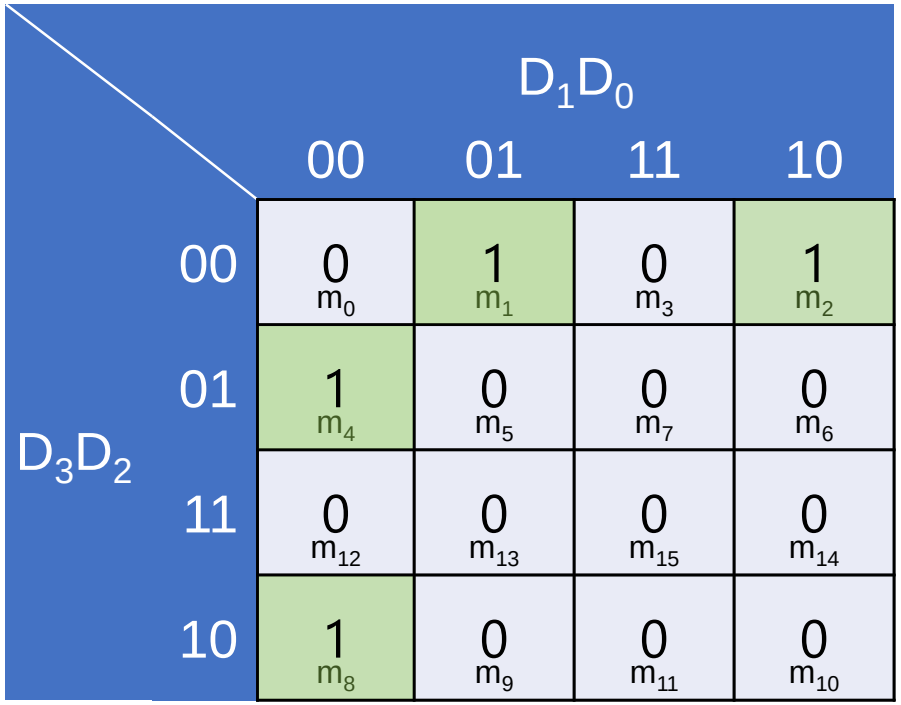
D_3	D_2	D_1	D_0	$F_2=Y$	$F_1=X$	$F_3=V$
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	X	X	0
0	1	0	0	1	0	1
0	1	0	1	X	X	0
0	1	1	0	X	X	0
0	1	1	1	X	X	0
1	0	0	0	1	1	1
1	0	0	1	X	X	0
1	0	1	0	X	X	0
1	0	1	1	X	X	0
1	1	0	0	X	X	0
1	1	0	1	X	X	0
1	1	1	0	X	X	0
1	1	1	1	X	X	0


D_1D_0
00
01
11
10
D_3D_2
00
01
11
10
X
m_0
0
m_1
X
m_3
1
m_2
0
m_4
X
m_5
X
m_7
X
m_6
X
m_{12}
X
m_{13}
X
m_{15}
X
m_{14}
1
m_8
X
m_9
X
m_{11}
X
m_{10}

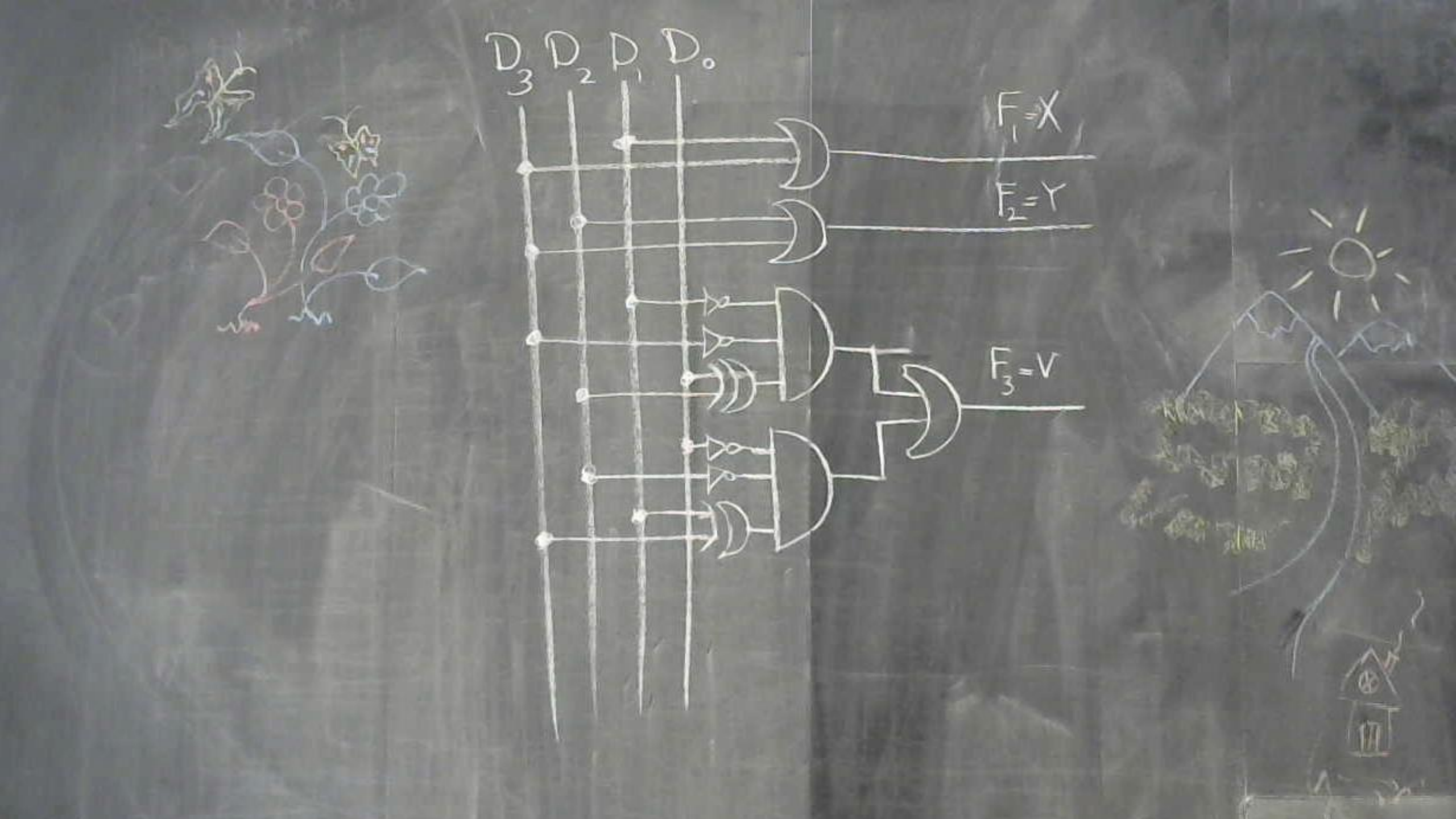
$$F_1 = X = D_1 + D_3$$


D1D0
00
01
11
10
D3D2
00
01
11
10
X
m0
0
m1
X
m3
0
m2
1
m4
X
m5
X
m7
X
m6
X
m12
X
m13
X
m15
X
m14
1
m8
X
m9
X
m11
X
m10

$$F_2 = Y = D_2 + D_3$$



$$F_3 = V = D'_3D'_2D'_1D_0 + D'_3D_2D'_1D'_0 + D'_3D'_2D_1D'_0 + D_3D'_2D'_1D'_0$$
$$= D'_3D'_1(D'_2D_0 + D_2D'_0) + D'_2D'_0(D'_3D_1 + D_3D'_1)$$
$$= D'_3D'_1(D_2 \oplus D_0) + D'_2D'_0(D_3 \oplus D_1)$$


D3 D2 D1 D0
F1=X
F2=Y
F3=V

Priority Encoder

D_3	D_2	D_1	D_0	$F_2=Y$	$F_1=X$	$F_3=V$
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	X	X	0
0	1	0	0	1	0	1
0	1	0	1	X	X	0
0	1	1	0	X	X	0
0	1	1	1	X	X	0
1	0	0	0	1	1	1
1	0	0	1	X	X	0
1	0	1	0	X	X	0
1	0	1	1	X	X	0
1	1	0	0	X	X	0
1	1	0	1	X	X	0
1	1	1	0	X	X	0
1	1	1	1	X	X	0

D_3	D_2	D_1	D_0	$F_2=Y$	$F_1=X$	$F_3=V$
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	X	X	0
0	1	0	0	1	0	1
0	1	0	1	X	X	0
0	1	1	0	X	X	0
0	1	1	1	X	X	0
1	0	0	0	1	1	1
1	0	0	1	X	X	0
1	0	1	0	X	X	0
1	0	1	1	X	X	0
1	1	0	0	X	X	0
1	1	0	1	X	X	0
1	1	1	0	X	X	0
1	1	1	1	X	X	0

D_3	D_2	D_1	D_0	$F_2=Y$	$F_1=X$	$F_3=V$
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	0	1	1
0	1	0	0	1	0	1
0	1	0	1	X	X	0
0	1	1	0	X	X	0
0	1	1	1	X	X	0
1	0	0	0	1	1	1
1	0	0	1	X	X	0
1	0	1	0	X	X	0
1	0	1	1	X	X	0
1	1	0	0	X	X	0
1	1	0	1	X	X	0
1	1	1	0	X	X	0
1	1	1	1	X	X	0

D_3	D_2	D_1	D_0	$F_2=Y$	$F_1=X$	$F_3=V$
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	~~1~~	0	1	1
0	1	0	0	1	0	1
0	1	0	~~1~~	X	X	0
0	1	1	0	X	X	0
0	1	1	1	X	X	0
1	0	0	0	1	1	1
1	0	0	1	X	X	0
1	0	1	0	X	X	0
1	0	1	1	X	X	0
1	1	0	0	X	X	0
1	1	0	1	X	X	0
1	1	1	0	X	X	0
1	1	1	1	X	X	0

D_3	D_2	D_1	D_0	$F_2=Y$	$F_1=X$	$F_3=V$
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	~~1~~	0	1	1
0	1	0	0	1	0	1
0	1	0	~~1~~	1	0	1
0	1	1	0	X	X	0
0	1	1	1	X	X	0
1	0	0	0	1	1	1
1	0	0	1	X	X	0
1	0	1	0	X	X	0
1	0	1	1	X	X	0
1	1	0	0	X	X	0
1	1	0	1	X	X	0
1	1	1	0	X	X	0
1	1	1	1	X	X	0

D_3	D_2	D_1	D_0	$F_2=Y$	$F_1=X$	$F_3=V$
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	~~1~~	0	1	1
0	1	0	0	1	0	1
0	1	0	~~1~~	1	0	1
0	1	~~1~~	0	1	0	1
0	1	1	1	X	X	0
1	0	0	0	1	1	1
1	0	0	1	X	X	0
1	0	1	0	X	X	0
1	0	1	1	X	X	0
1	1	0	0	X	X	0
1	1	0	1	X	X	0
1	1	1	0	X	X	0
1	1	1	1	X	X	0

D_3	D_2	D_1	D_0	F_2=Y	F_1=X	F_3=V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	~~1~~	0	1	1
0	1	0	0	1	0	1
0	1	0	~~1~~	1	0	1
0	1	~~1~~	0	1	0	1
0	1	~~1~~	~~1~~	1	0	1
1	0	0	0	1	1	1
1	0	0	1	X	X	0
1	0	1	0	X	X	0
1	0	1	1	X	X	0
1	1	0	0	X	X	0
1	1	0	1	X	X	0
1	1	1	0	X	X	0
1	1	1	1	X	X	0

D_3	D_2	D_1	D_0	$F_2=Y$	$F_1=X$	$F_3=V$
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	~~1~~	0	1	1
0	1	0	0	1	0	1
0	1	0	~~1~~	1	0	1
0	1	~~1~~	0	1	0	1
0	1	~~1~~	~~1~~	1	0	1
1	0	0	0	1	1	1
1	0	0	~~1~~	1	1	1
1	0	~~1~~	0	1	1	1
1	0	~~1~~	~~1~~	1	1	1
1	~~1~~	0	0	1	1	1
1	~~1~~	0	1	1	1	1
1	~~1~~	~~1~~	0	1	1	1
1	~~1~~	~~1~~	~~1~~	1	1	1

D_3	D_2	D_1	D_0	$F_2=Y$	$F_1=X$	$F_3=V$
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
0	1	X	X	1	0	1
0	1	X	X	1	0	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1
1	X	X	X	1	1	1
1	X	X	X	1	1	1
1	X	X	X	1	1	1
1	X	X	X	1	1	1
1	X	X	X	1	1	1
1	X	X	X	1	1	1
1	X	X	X	1	1	1

D_3	D_2	D_1	D_0	F_2=Y	F_1=X	F_3=V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

$$x = D_2 + D_3$$
$$y = D_3 + D_1 D_2'$$
$$V = D_0 + D_1 + D_2 + D_3$$

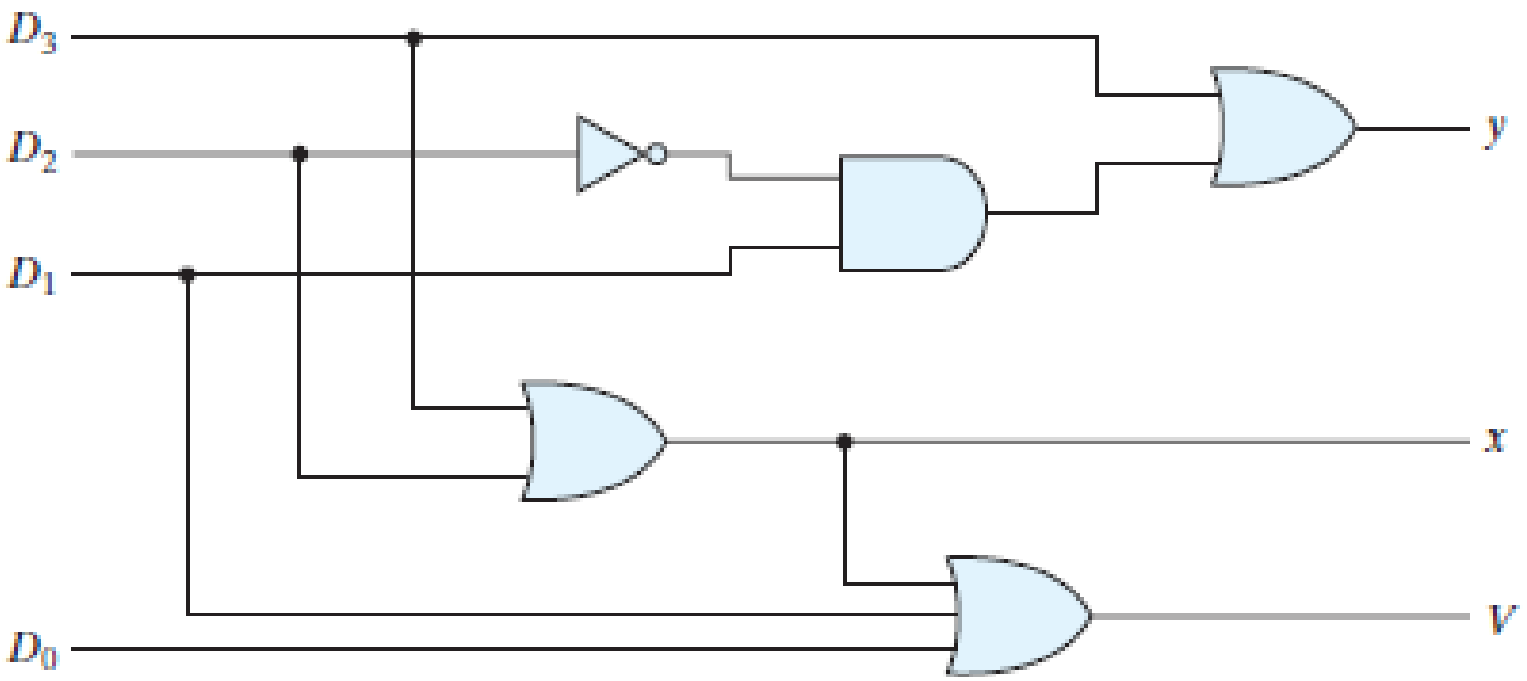
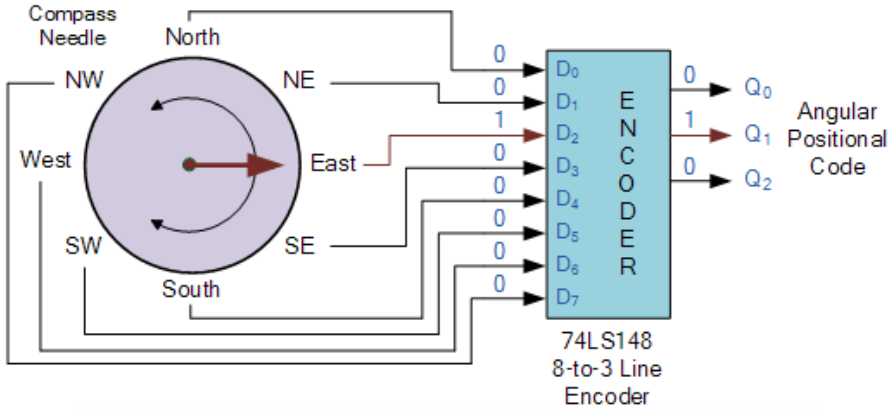


FIGURE 4.23
Four-Input priority encoder

Priority Encoder Applications

Positional Encoders

Priority Encoder Navigation



Compass Direction	Binary Output		
	Q_0	Q_1	Q_2
North	0	0	0
North-East	0	0	1
East	0	1	0
South-East	0	1	1
South	1	0	0
South-West	1	0	1
West	1	1	0
North-West	1	1	1

https://www.electronics-tutorials.ws/

Keyboard Encoders