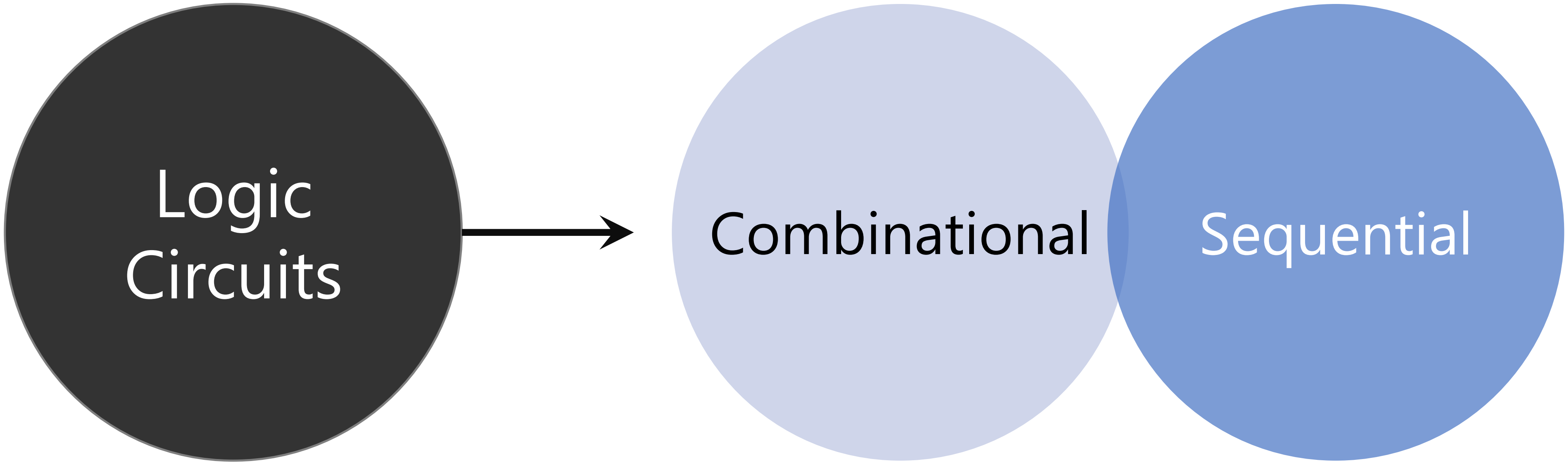

Logic Circuits
Combinational
Sequential

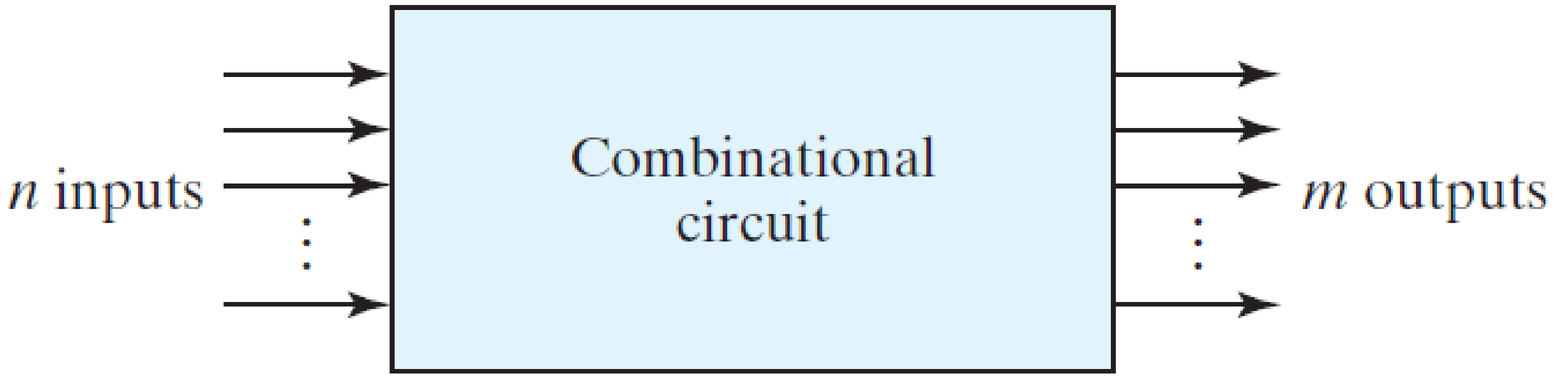


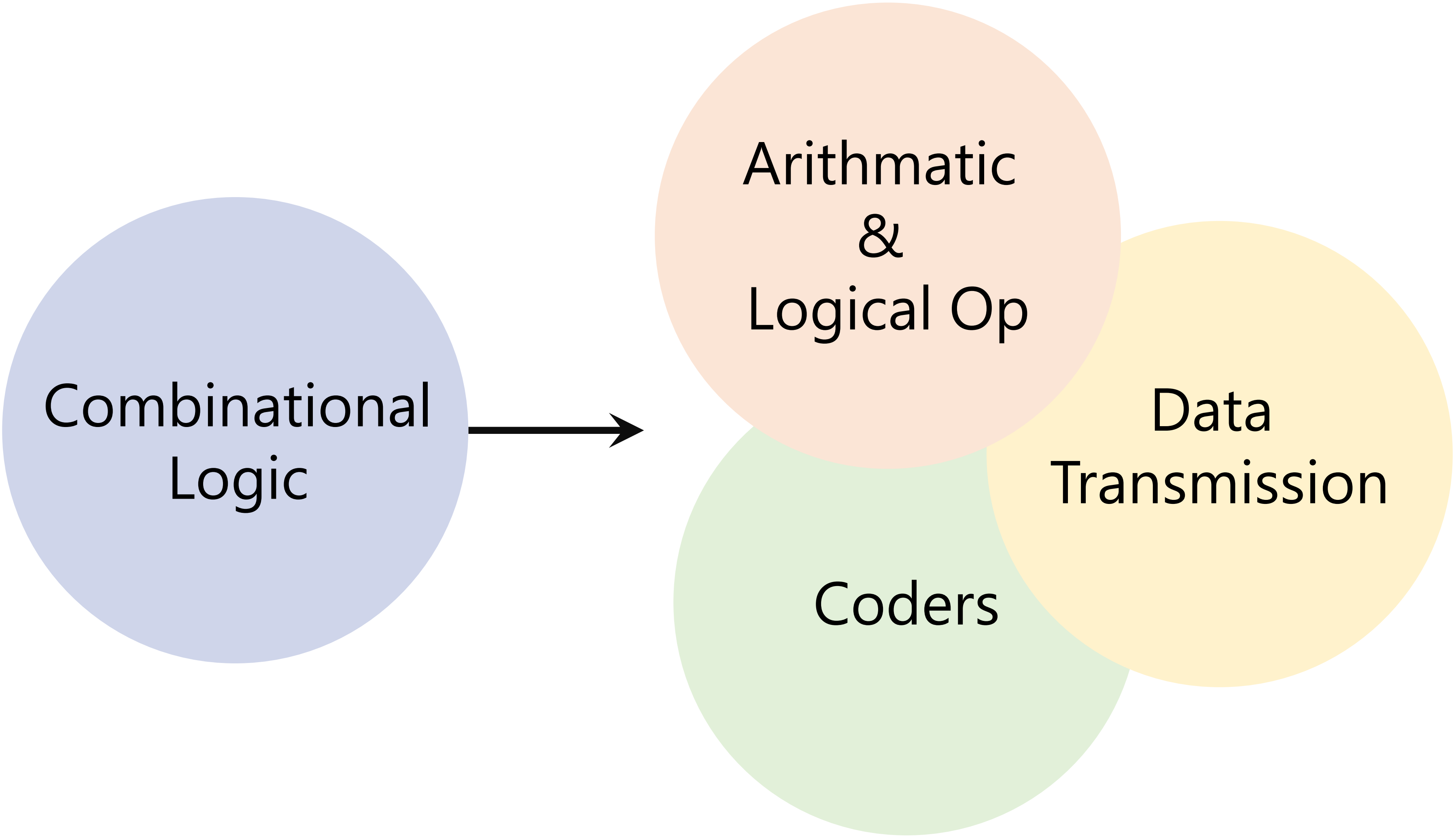
FIGURE 4.1
Block diagram of combinational circuit

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

Data Transmission

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

Binary Decoder

Binary Code Decoder
Display Decoder

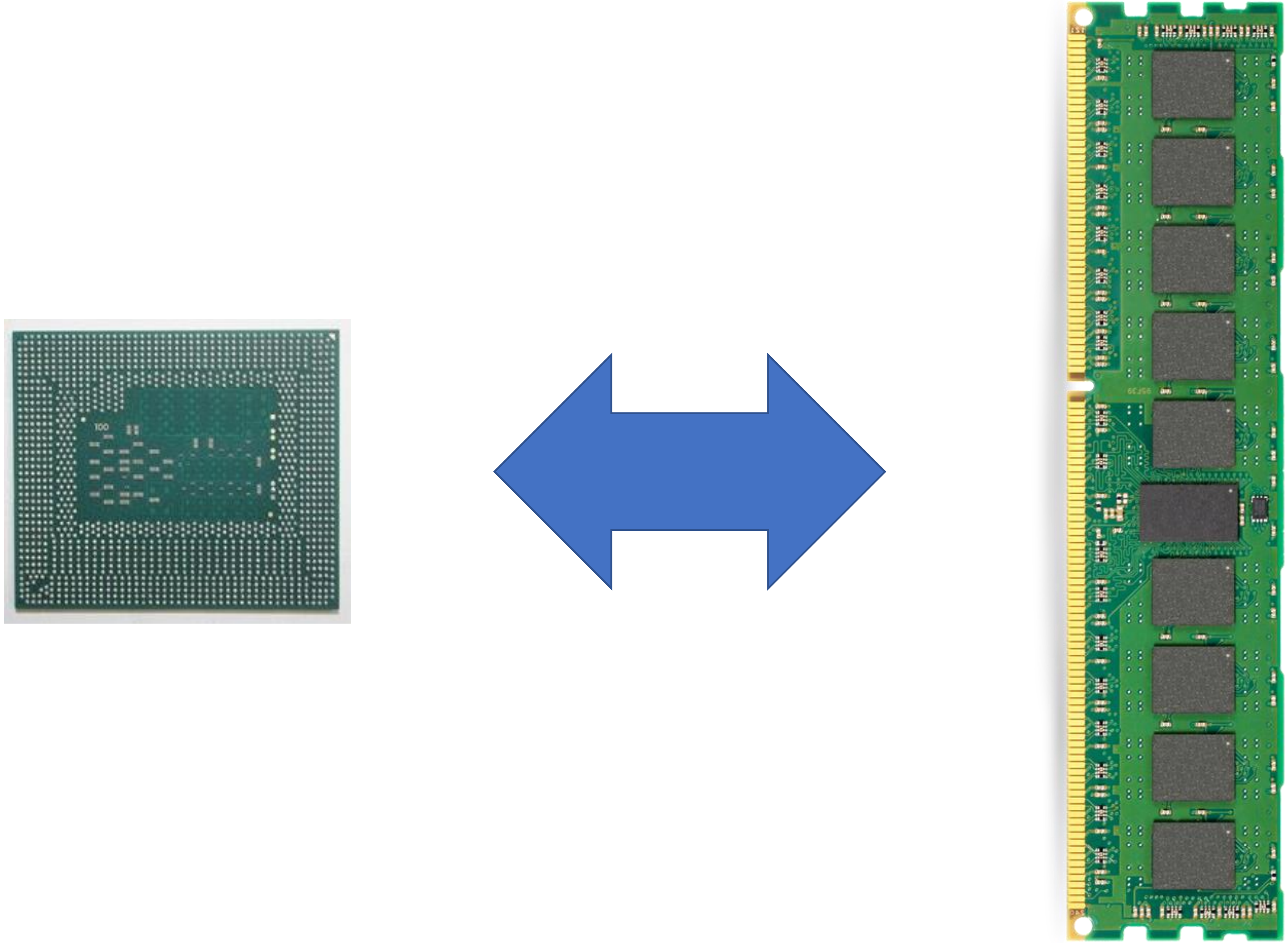
Decoder
Decode Binary to 1-hot

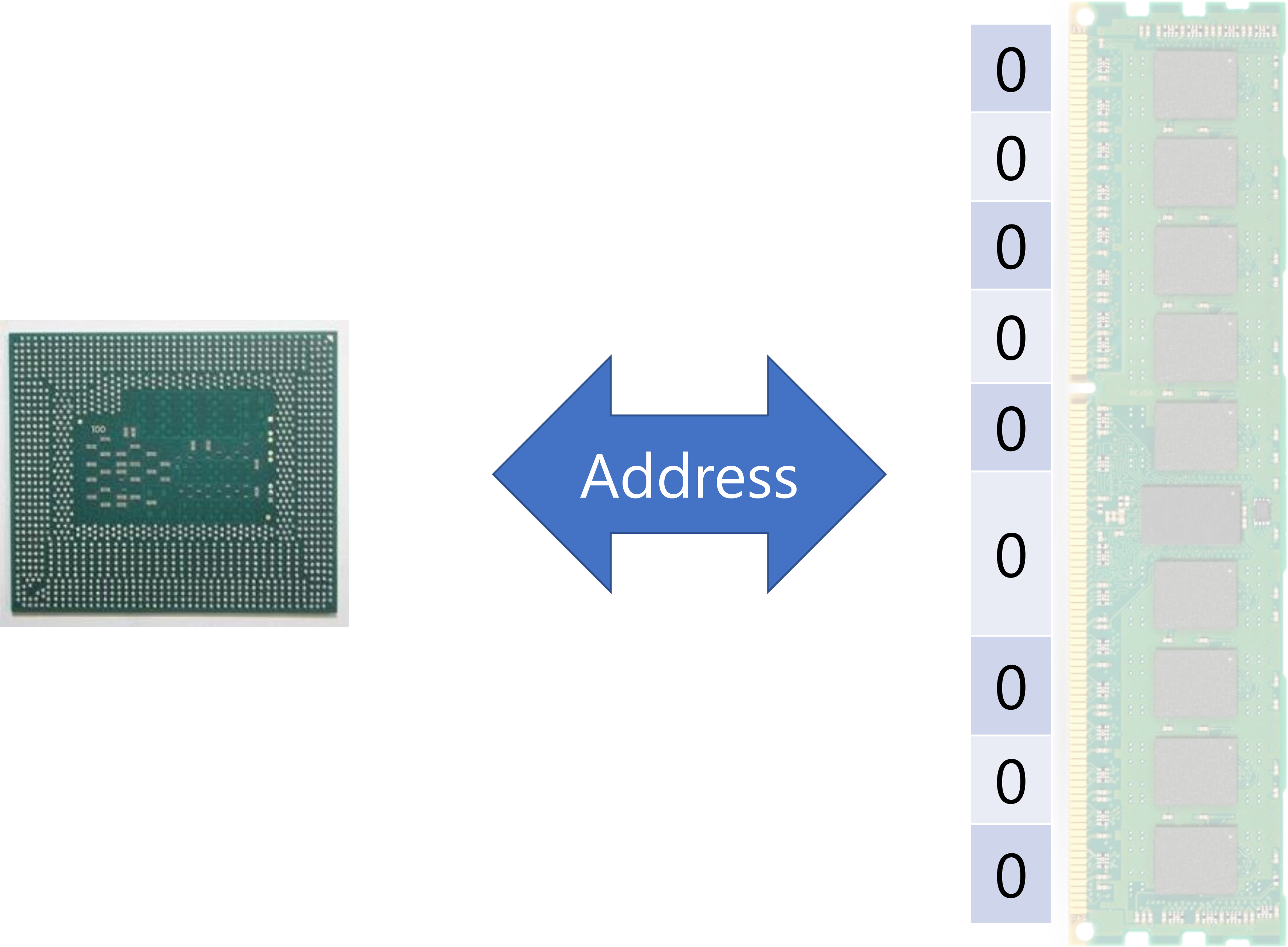
1-hot: a vector of bits with a single 1 and all the others 0

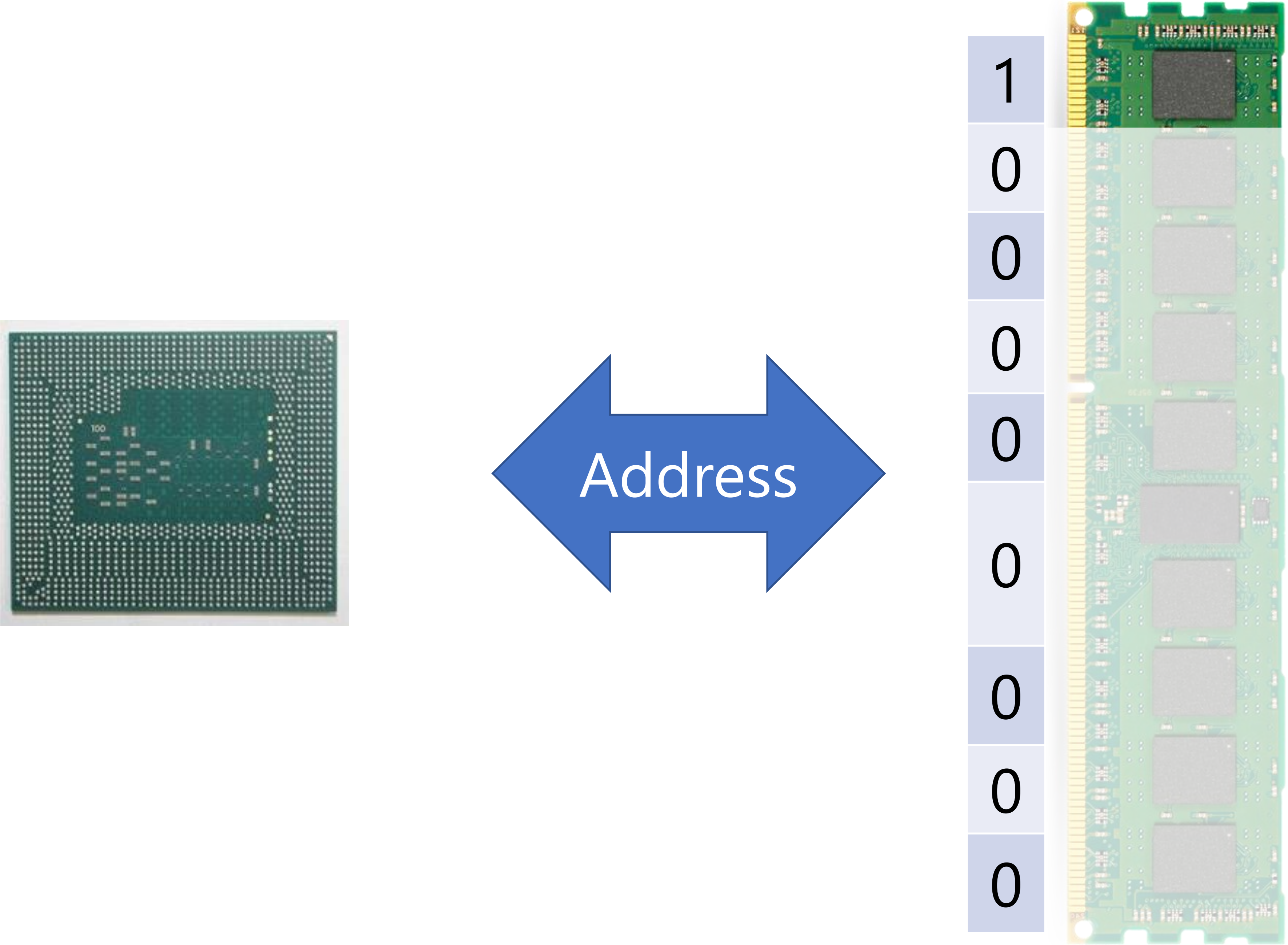
[0010000000]

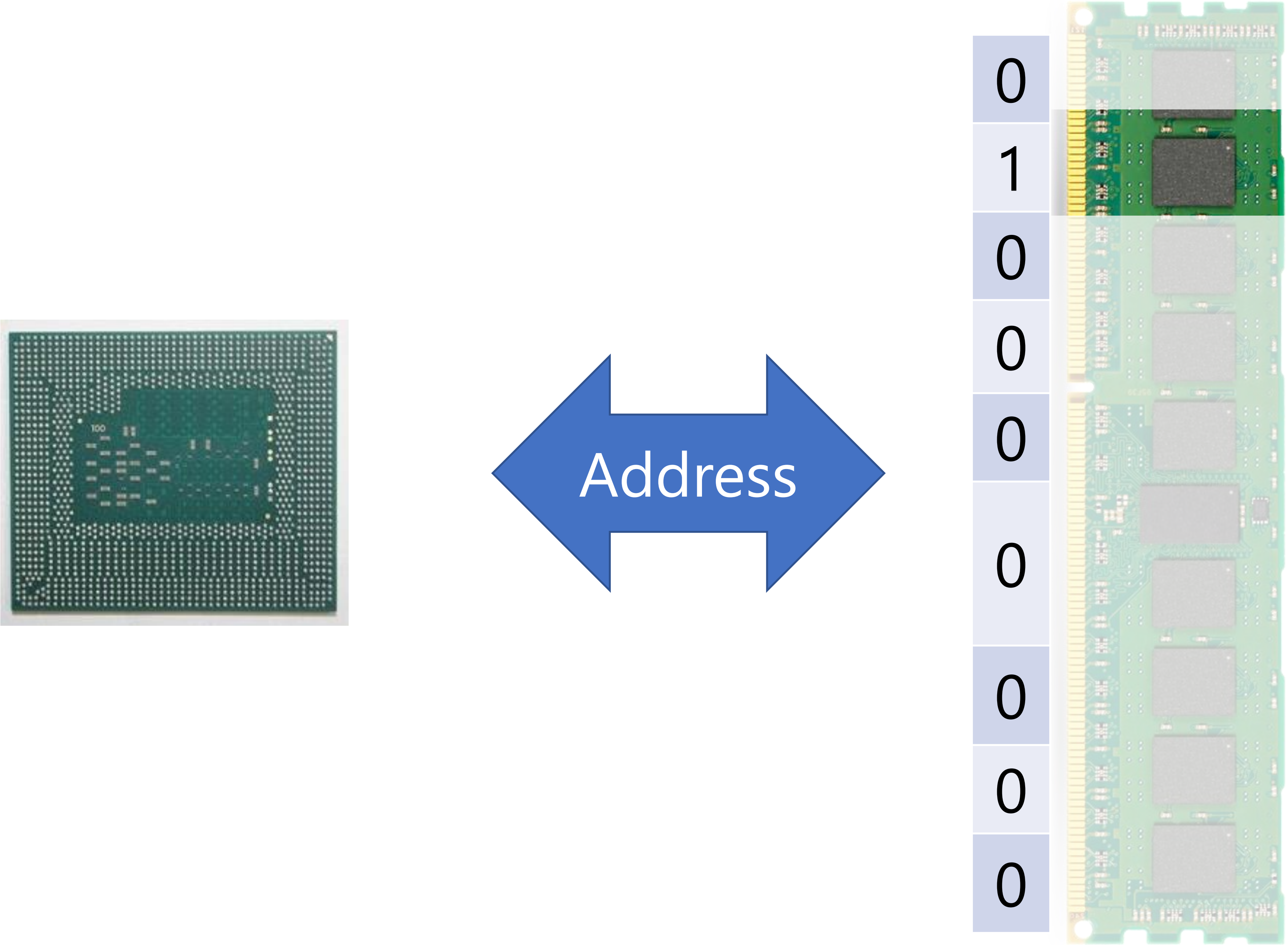
[0000000100]

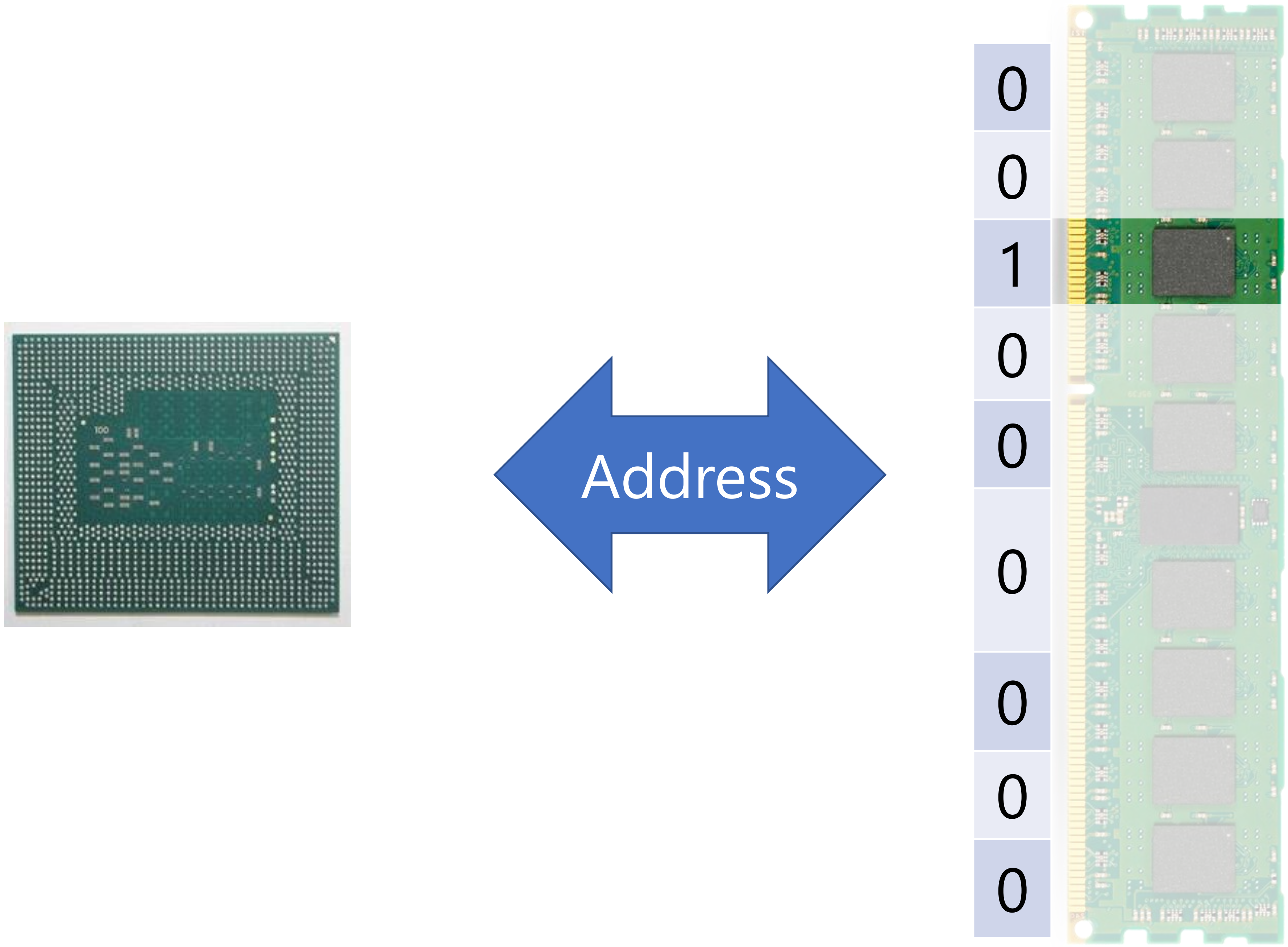
~~[0010010000]~~




Address
0
0
0
0
0
0
0
0
0


Address
1
0
0
0
0
0
0
0
0

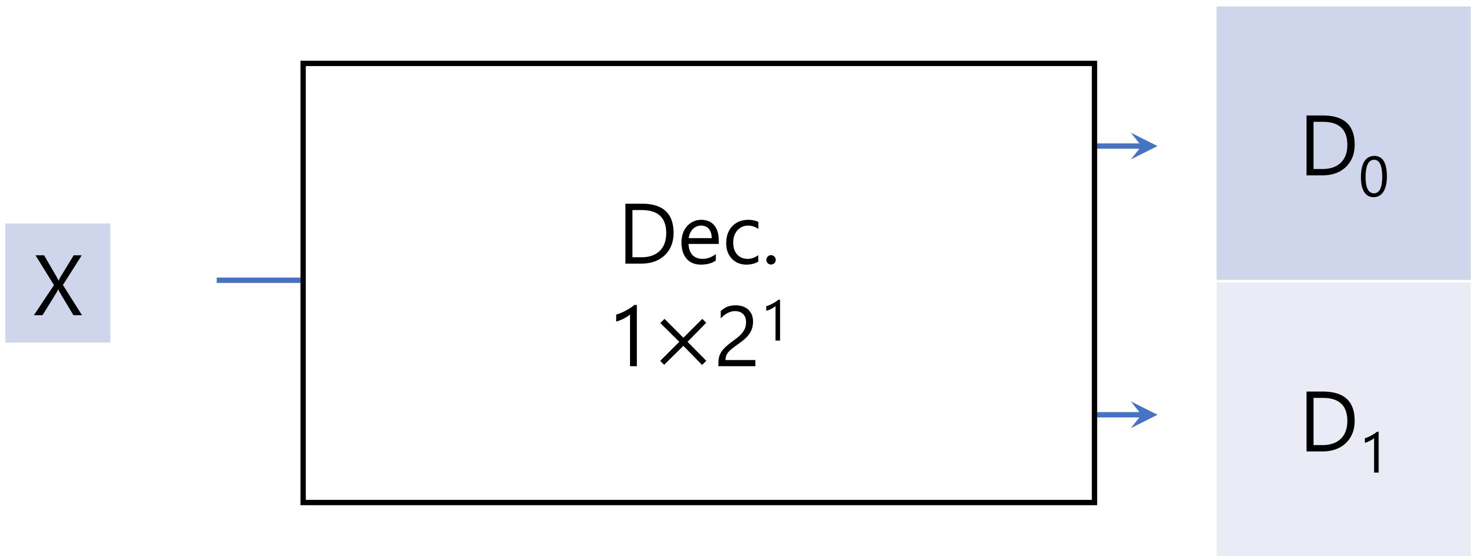

Address
0
1
0
0
0
0
0
0
0


Address
0
0
1
0
0
0
0
0
0


X
D_0
D_1

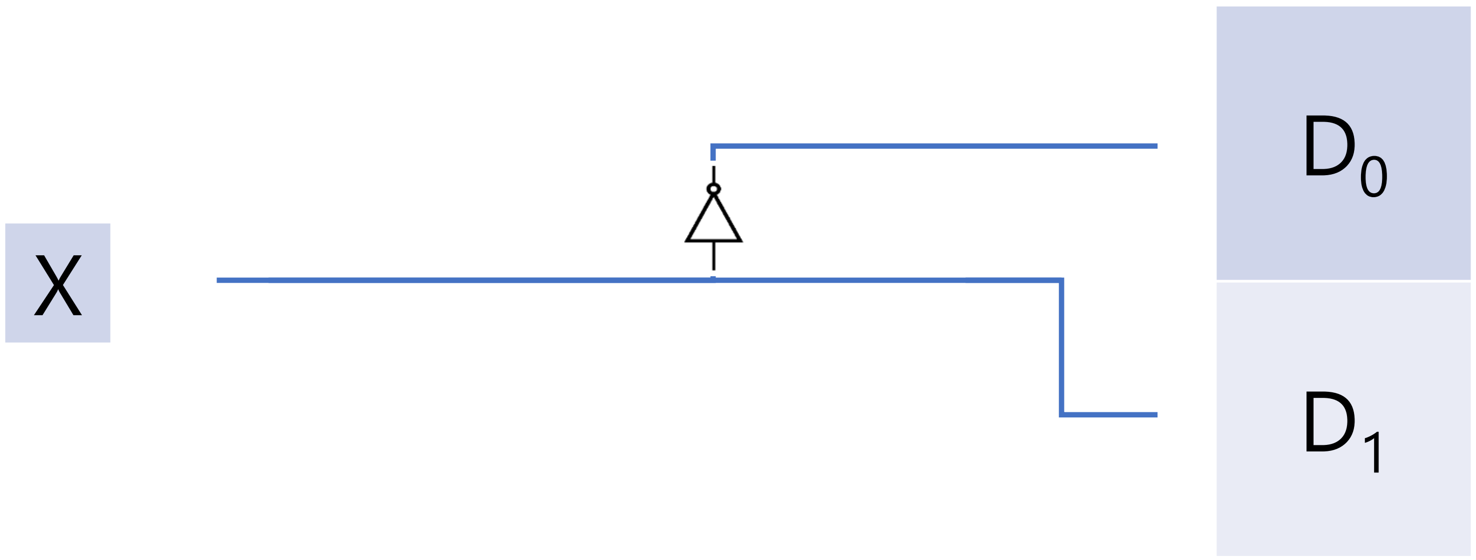

0
$D_0=1$
$D_1=0$

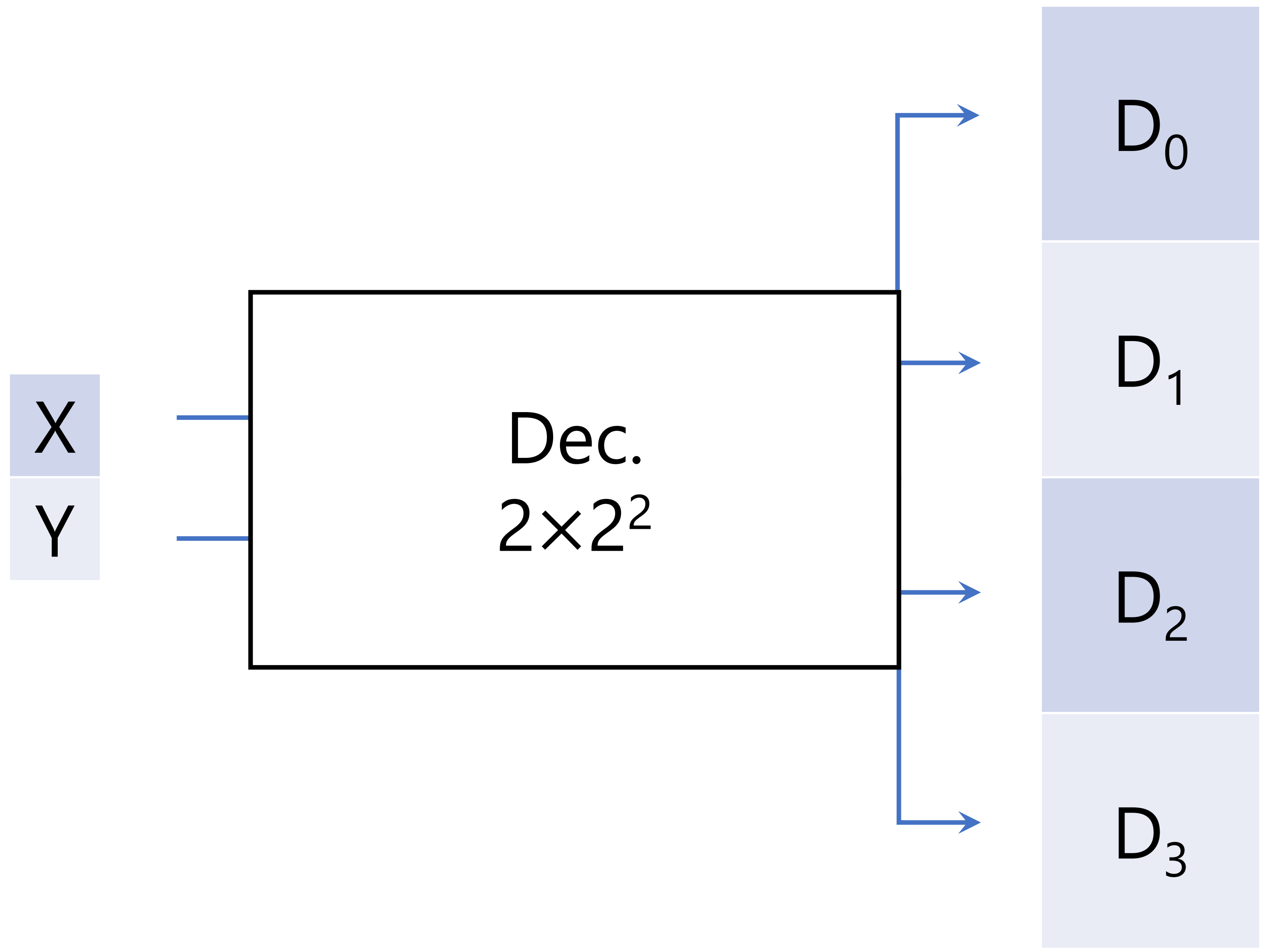

1
$D_0=0$
$D_1=1$

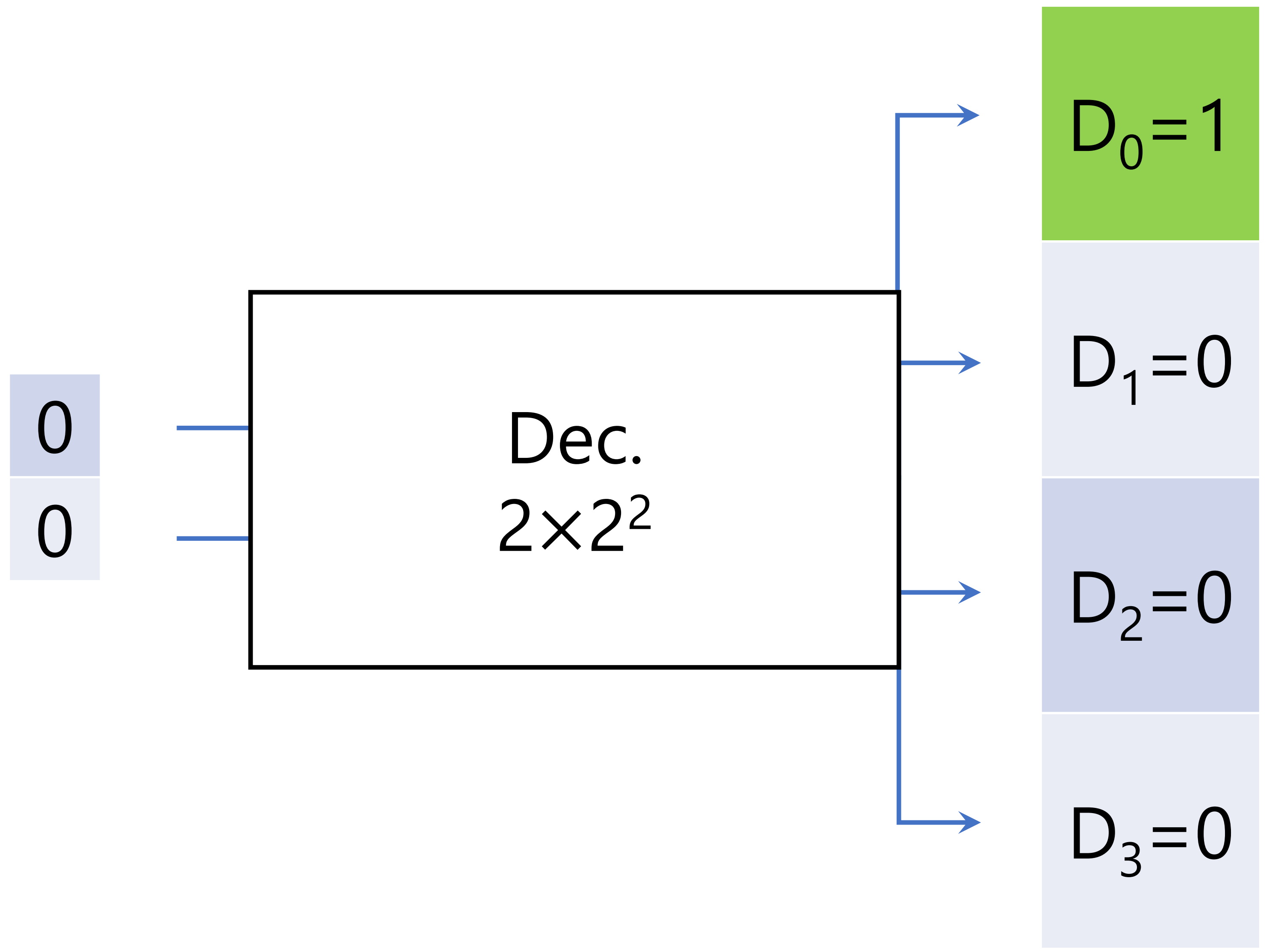

X
Dec.
1×2¹
D₀
D₁

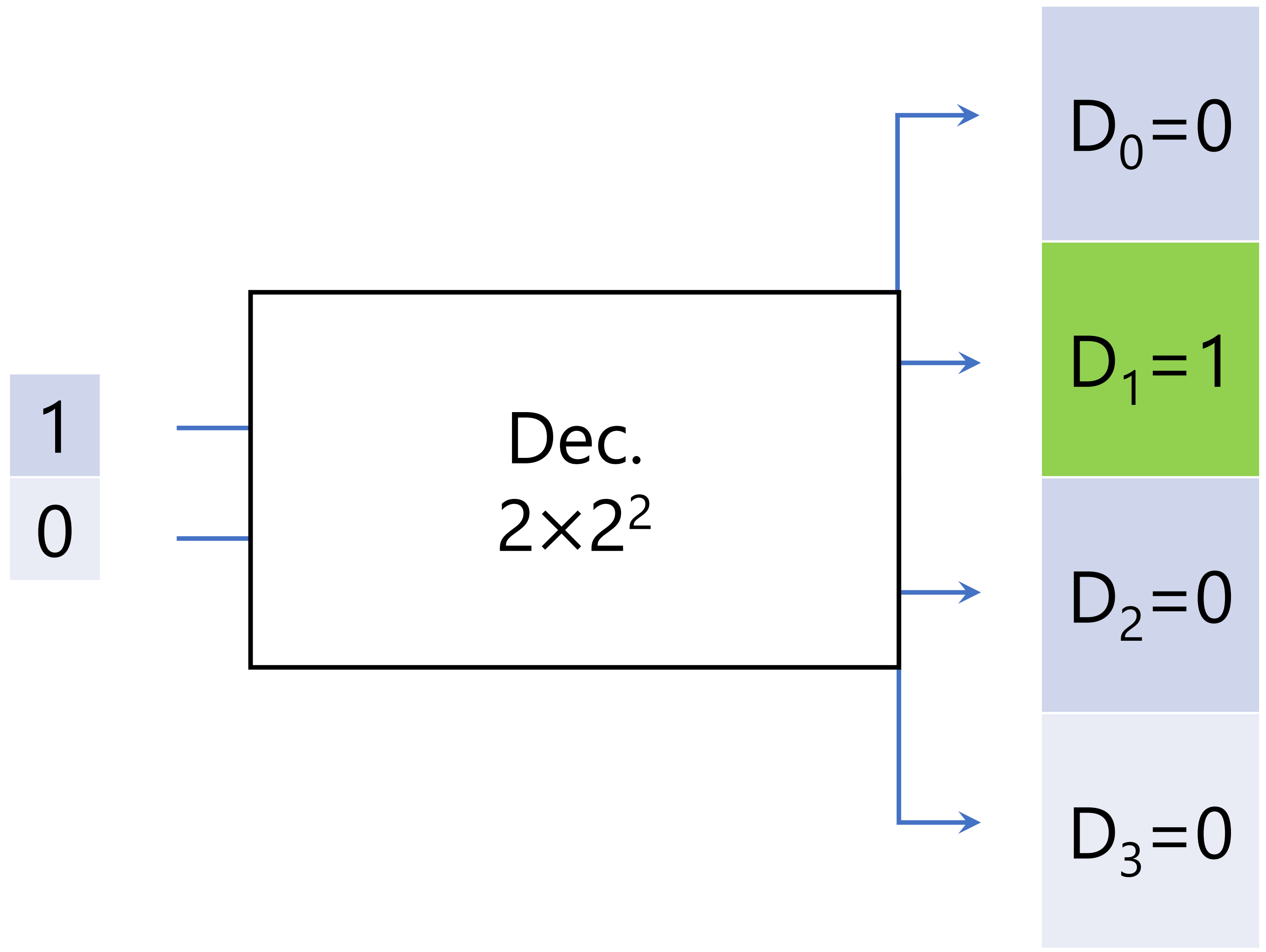
X	D_0	D_1
0	1	0
1	0	1

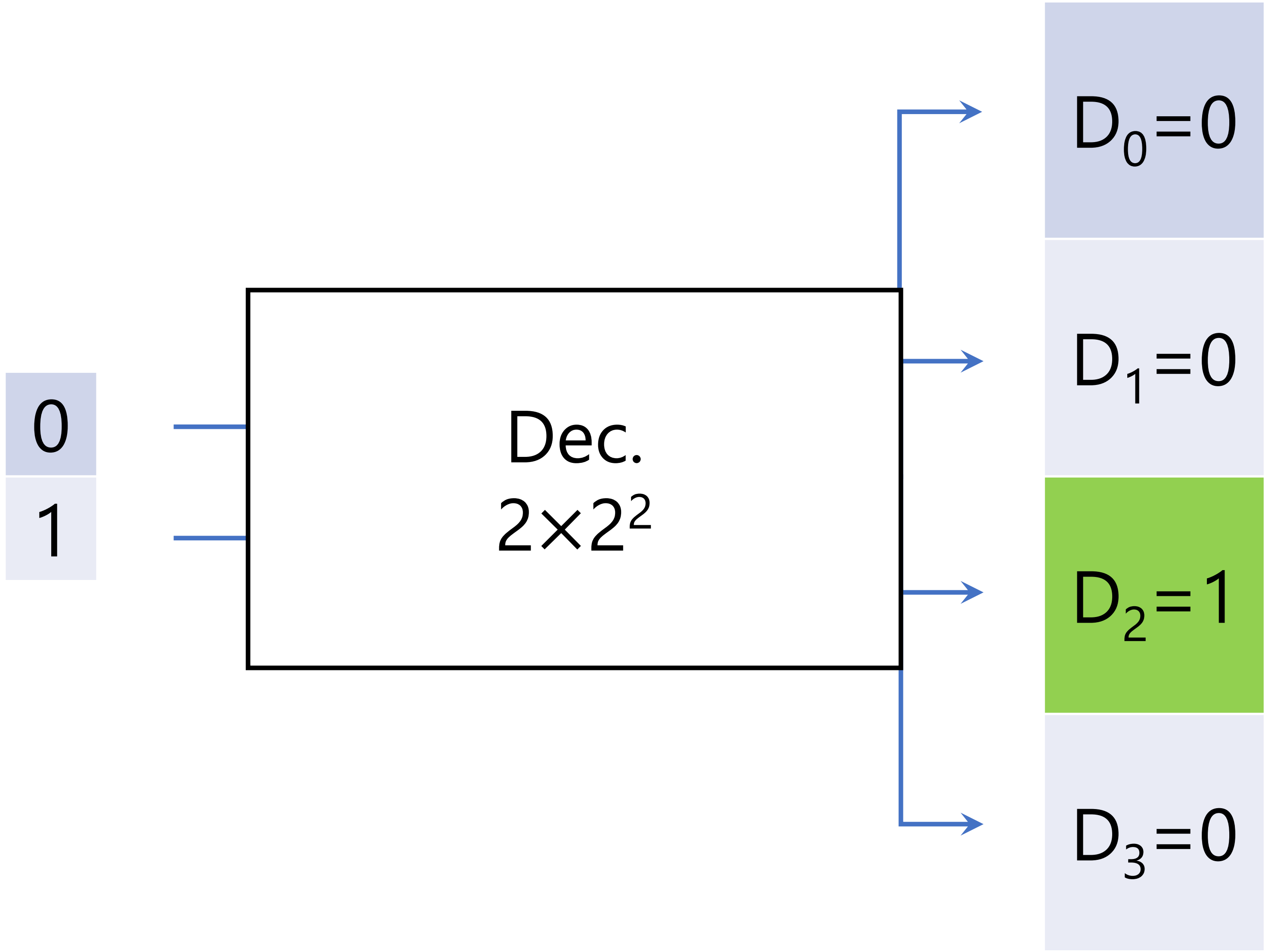
X	$D_0 = m_0$	$D_1 = m_1$
0	1	0
1	0	1

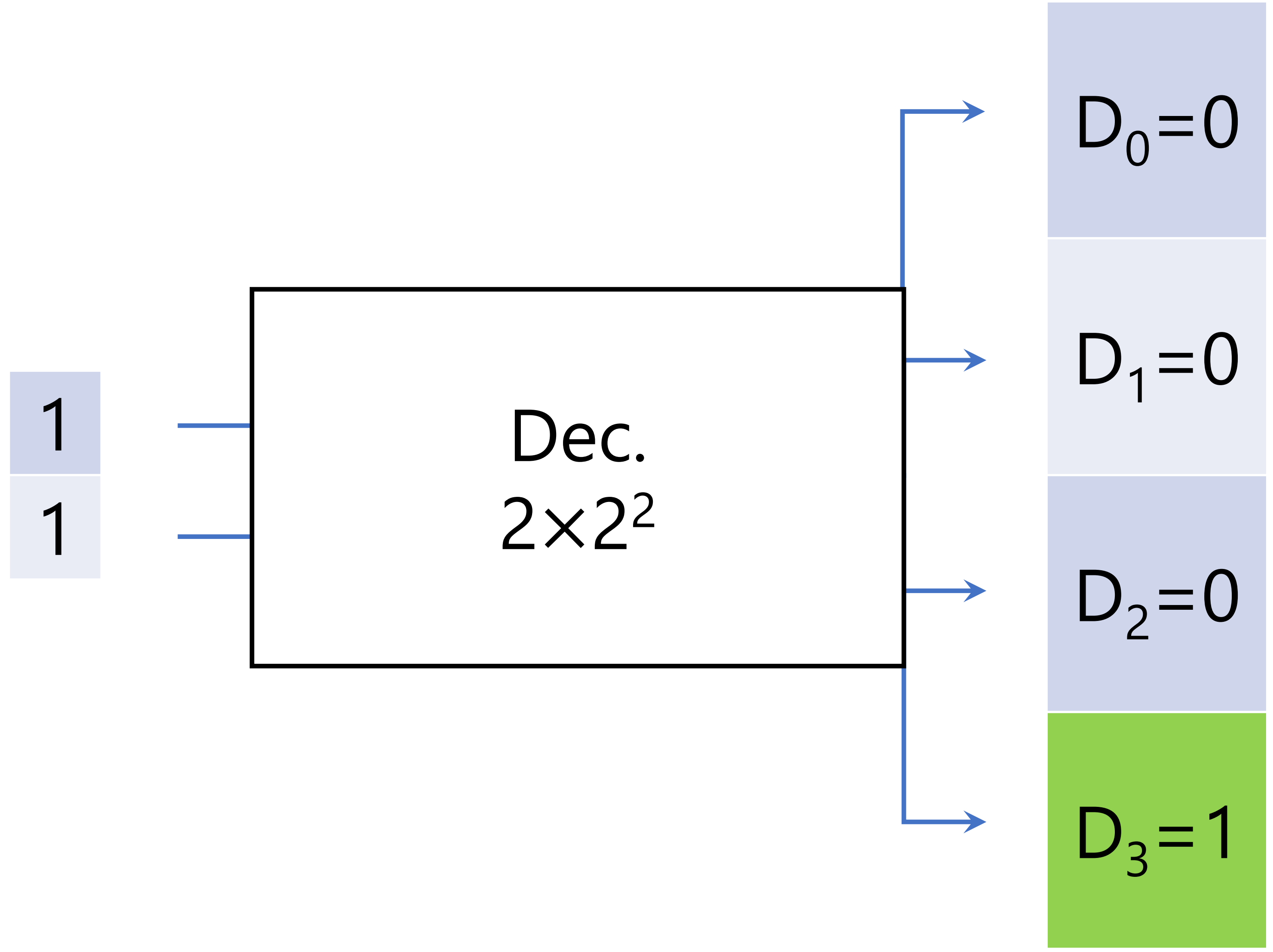

X
D0
D1


X
Y
Dec.
2×2^2
D_0
D_1
D_2
D_3


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

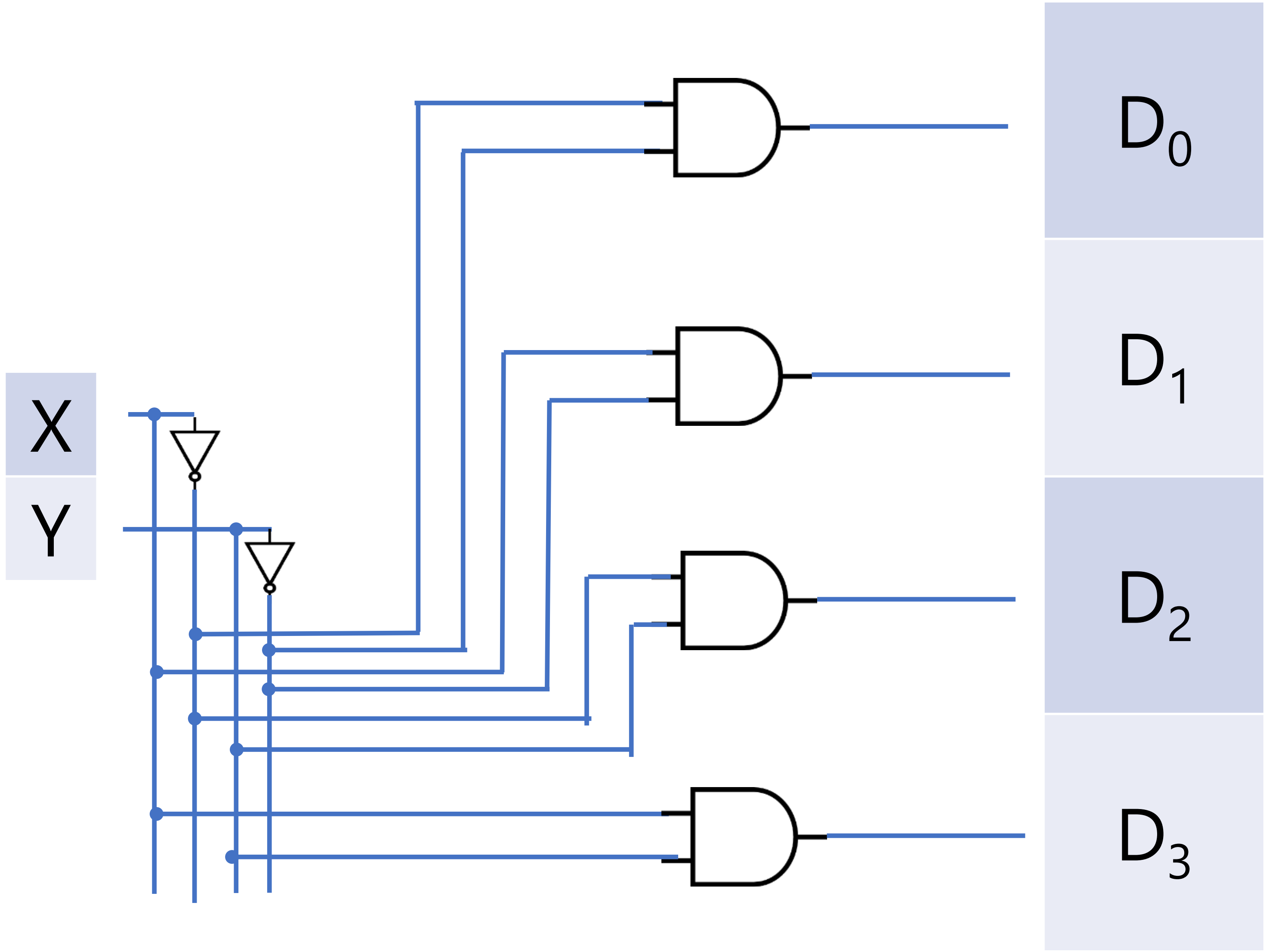

X
Y
D_0
D_1
D_2
D_3

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

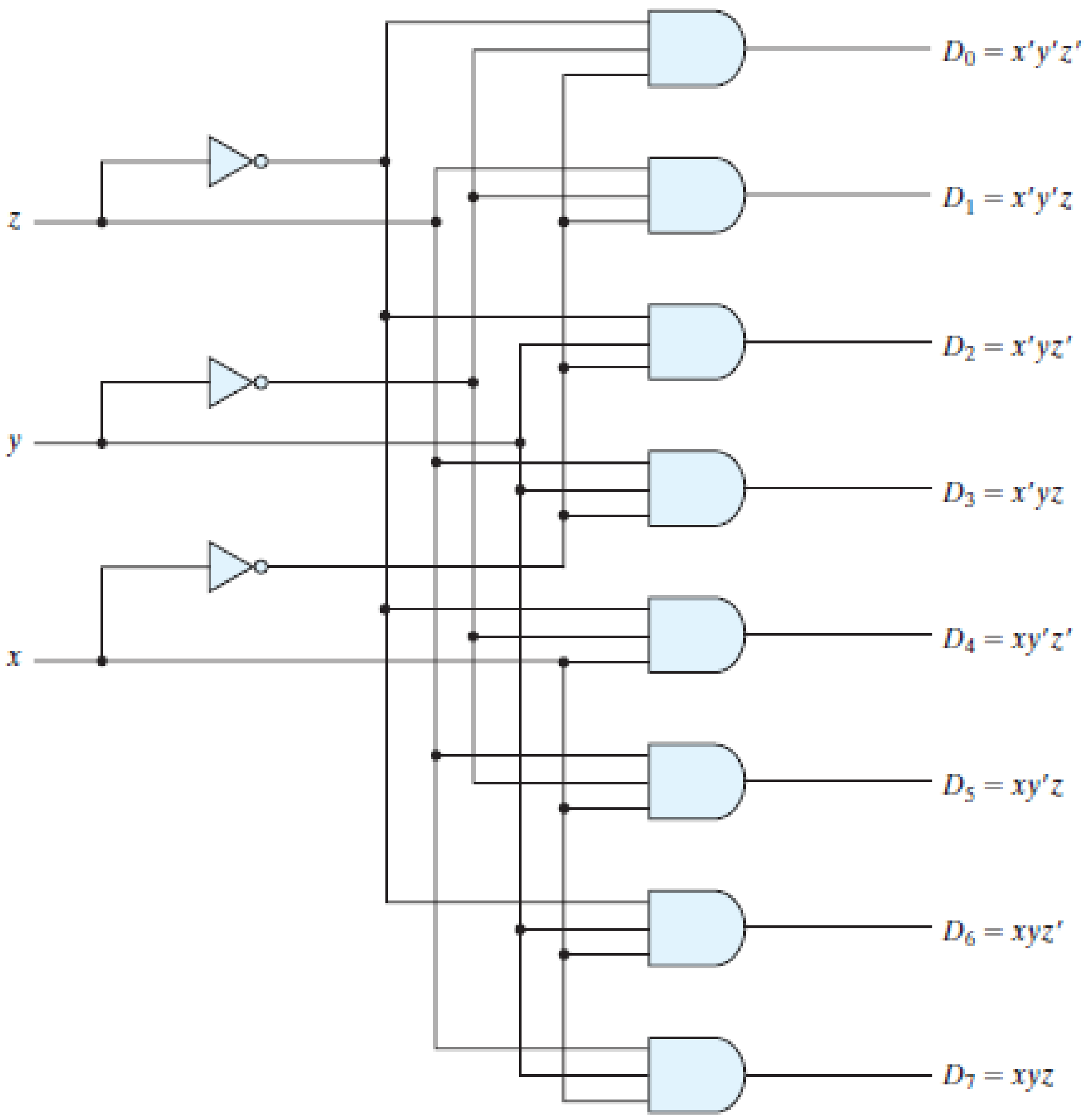


FIGURE 4.18
Three-to-eight-line decoder

Decoder

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

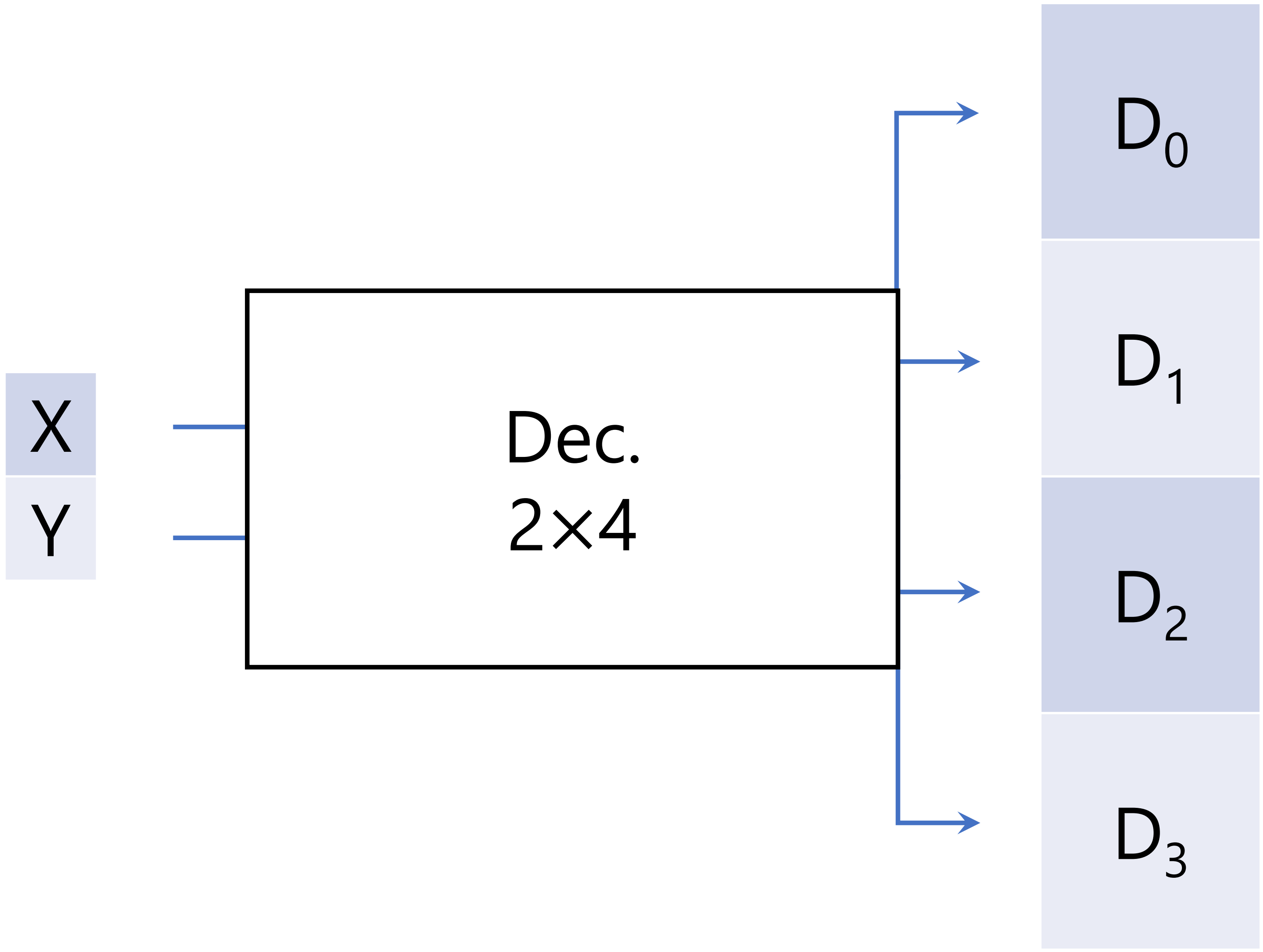
Decoder

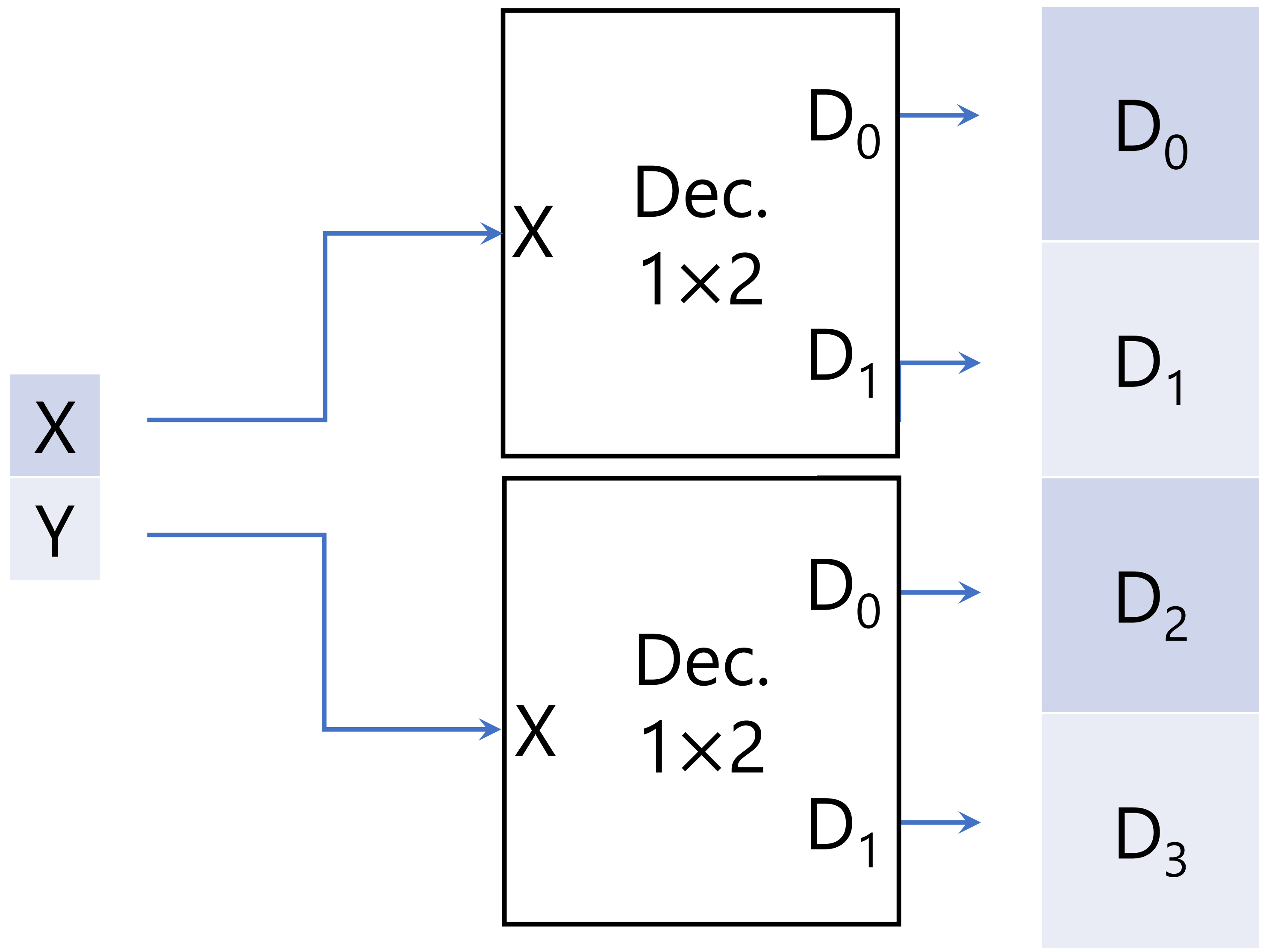
Decode n-Bit Binary to 2^n One-hot

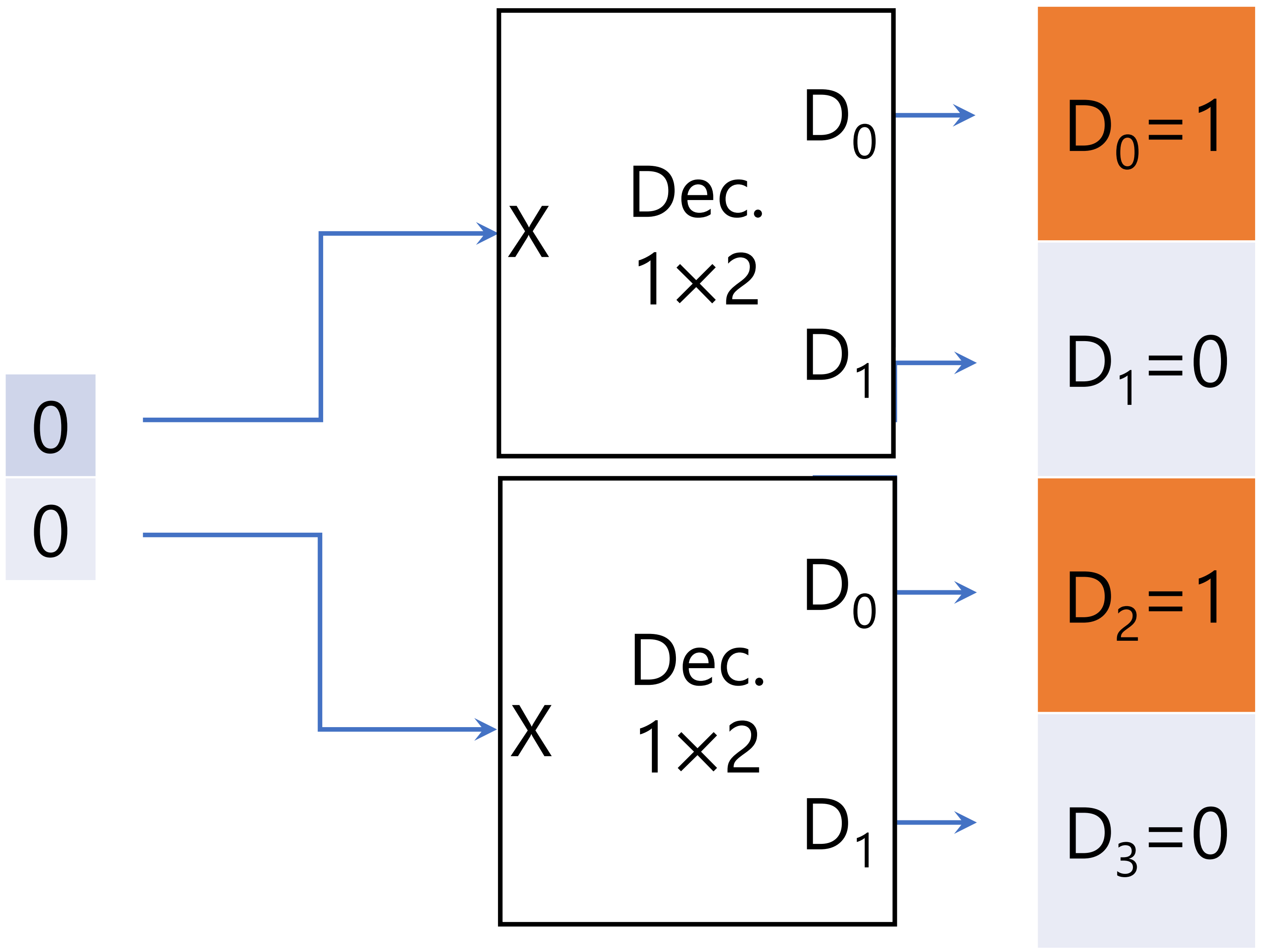
Decoder

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

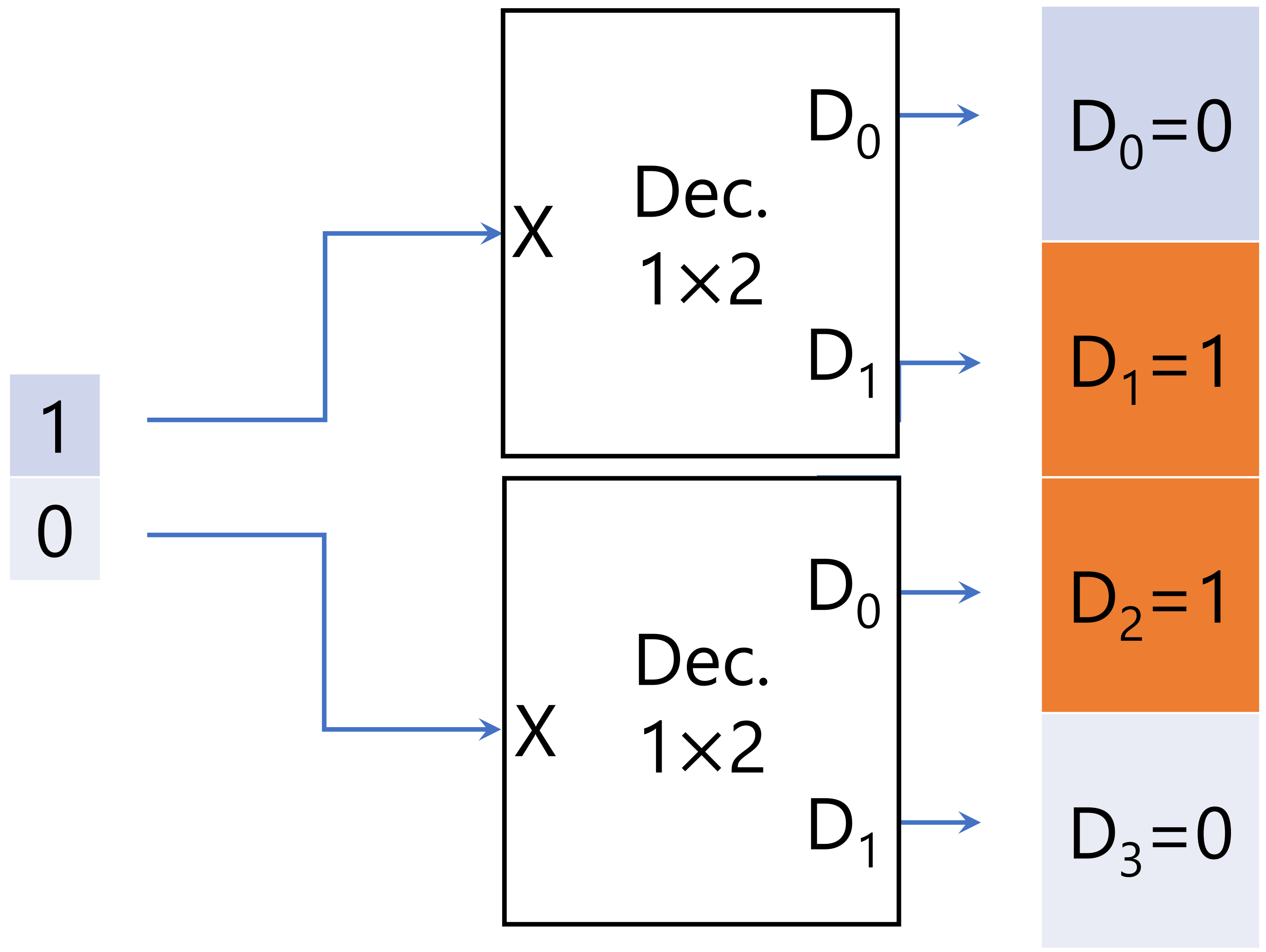
Re-Use 1×2^1 Decoder


X
Y
Dec.
2×4
D0
D1
D2
D3

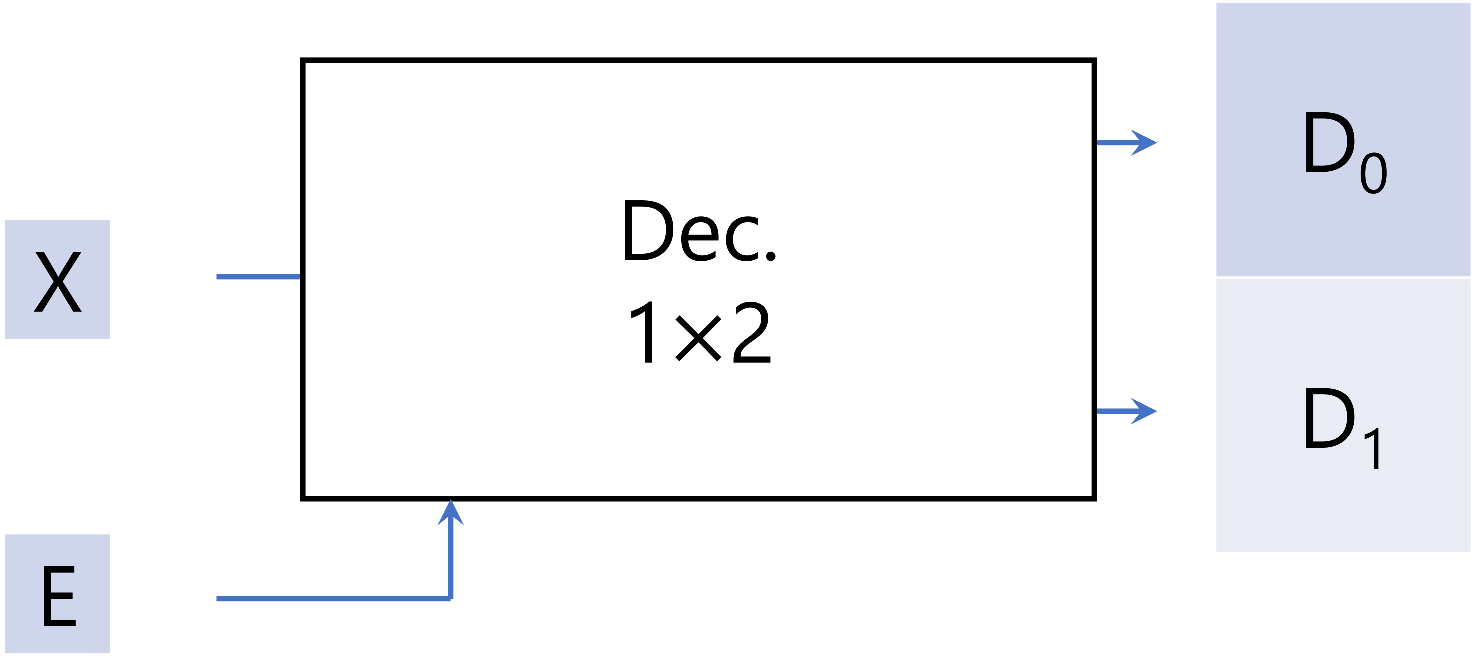

X
Y
X
Dec.
1×2
D0
D1
X
Dec.
1×2
D0
D1
D0
D1
D2
D3

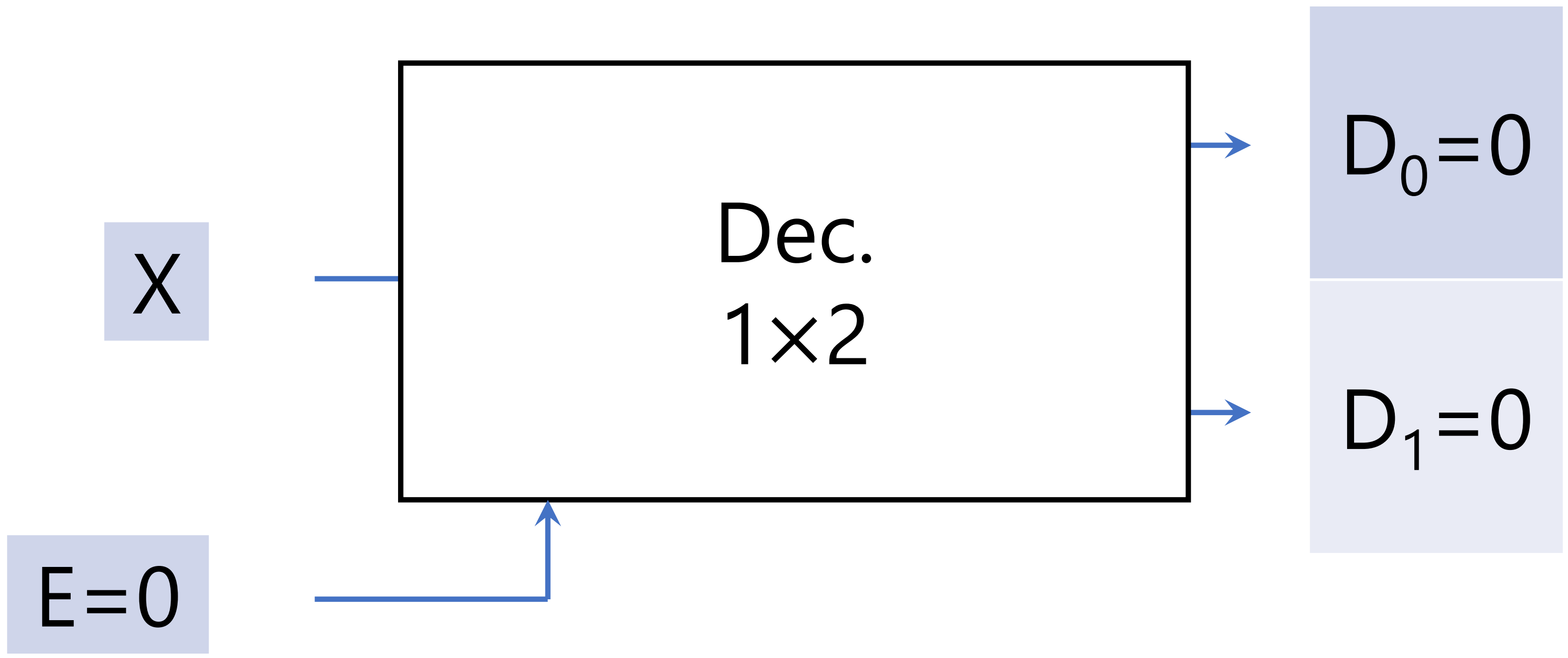


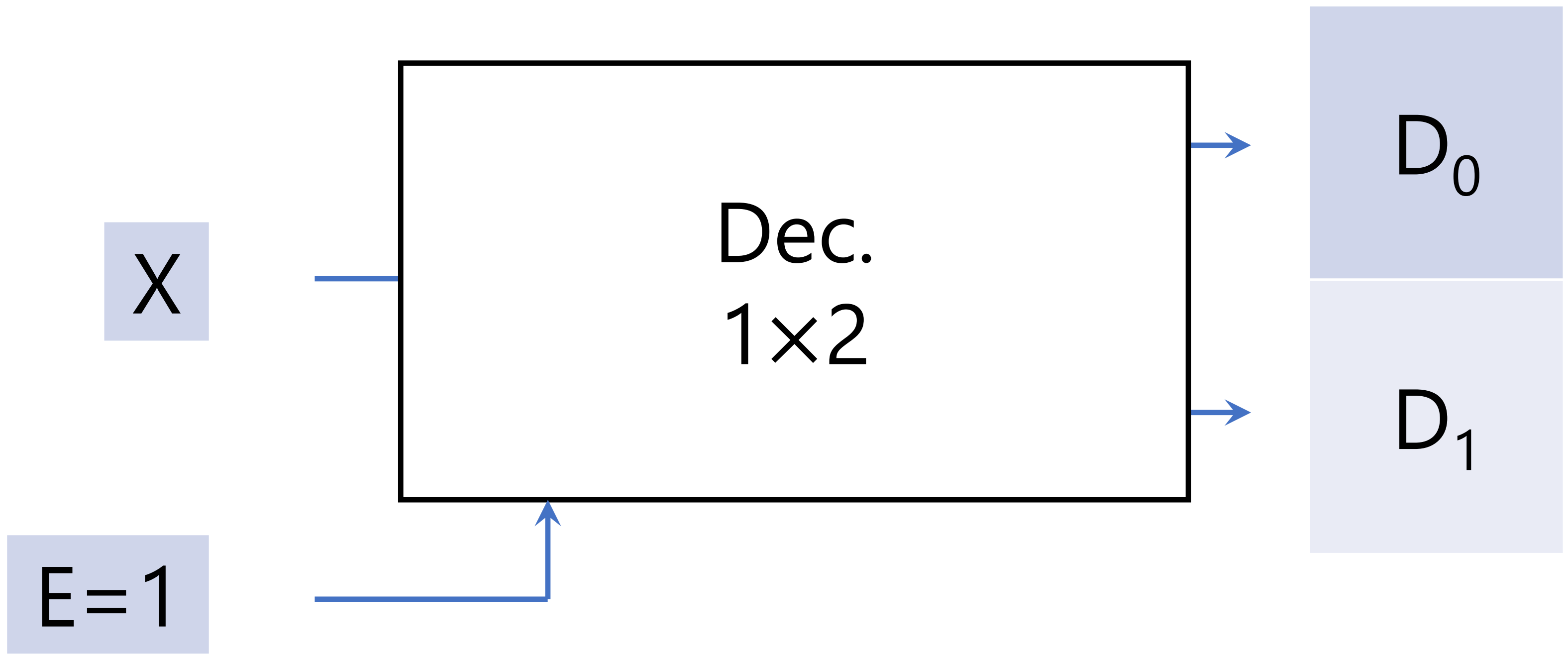
0
0
X
Dec.
1×2
D_0
D_1
X
Dec.
1×2
D_0
D_1
$D_0=1$
$D_1=0$
$D_2=1$
$D_3=0$


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

Decoder
Enable input

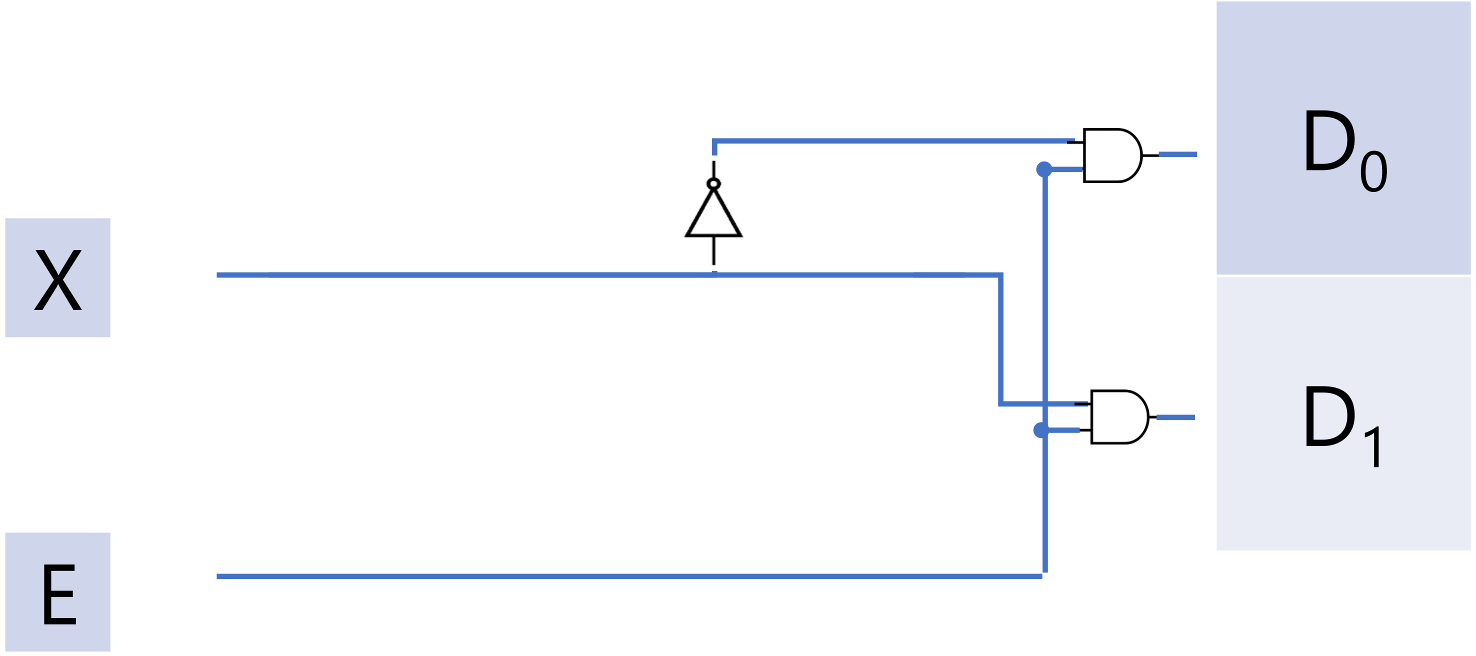

X
E
Dec.
1×2
D0
D1

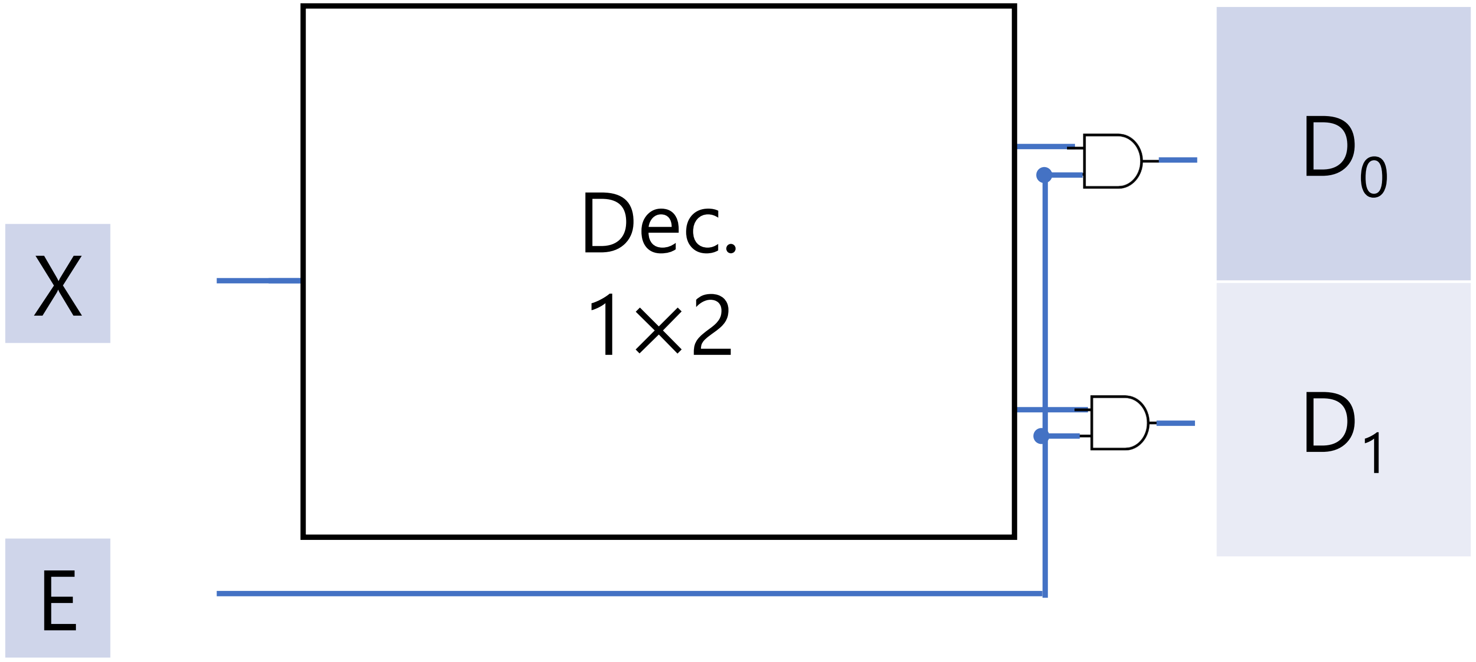

X
E=0
Dec.
1×2
D0=0
D1=0

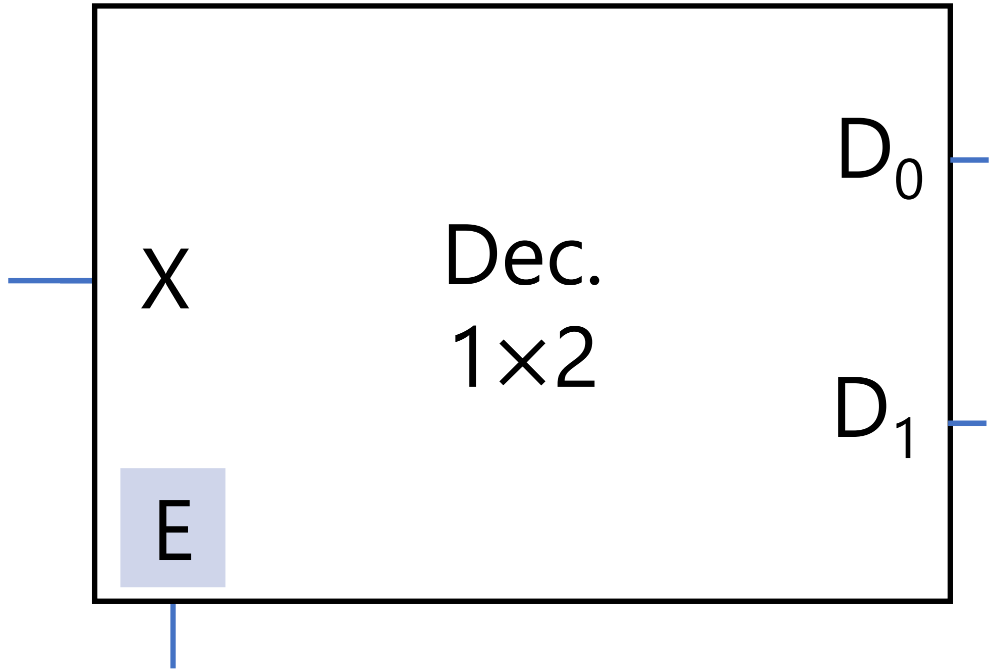

X
E=1
Dec.
1×2
D0
D1

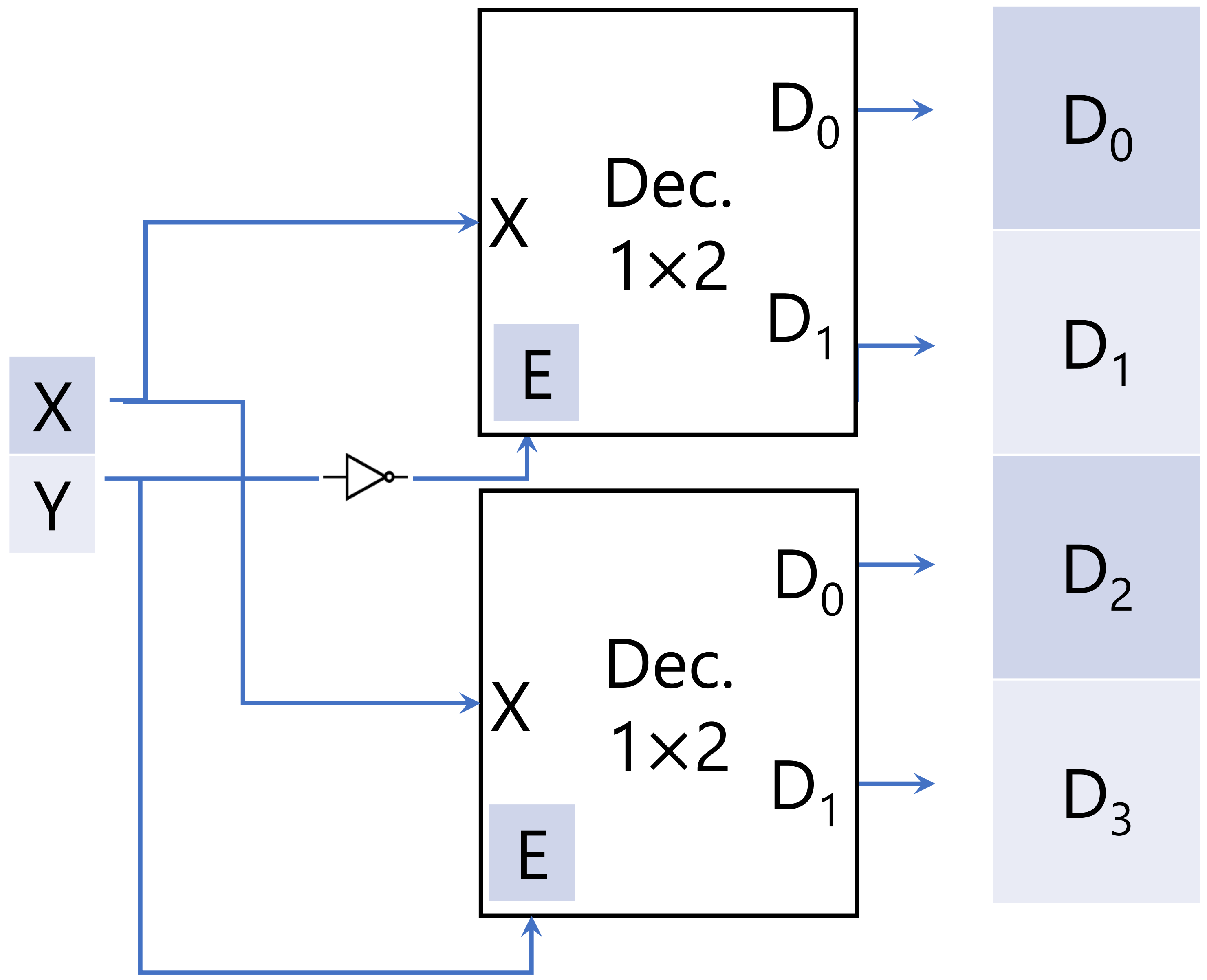
E	X	$D_0 = m_2$	$D_1 = m_3$
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

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

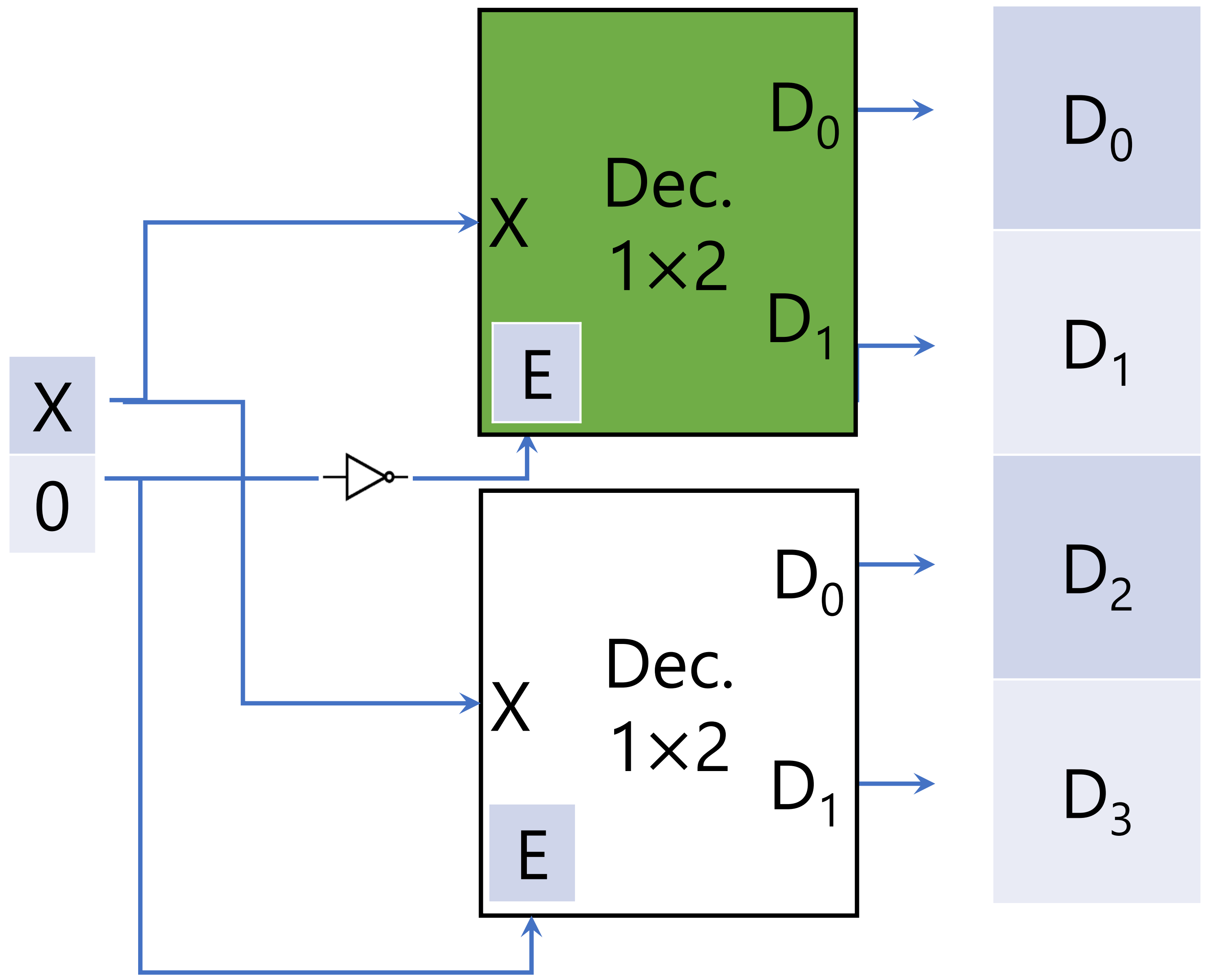

X
E
D0
D1

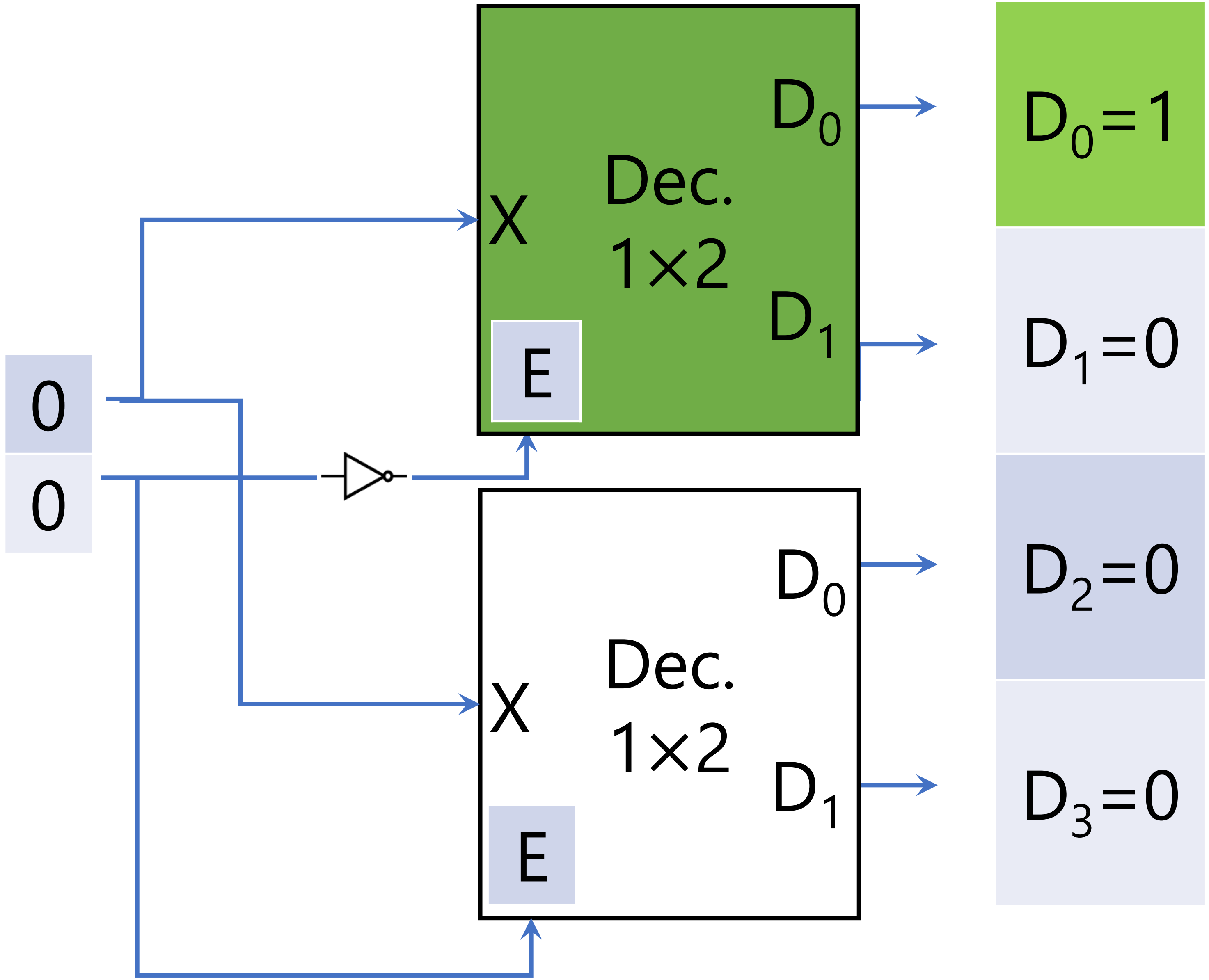

X
E
Dec.
1×2
D_0
D_1


X
Dec.
1×2
D0
D1
E

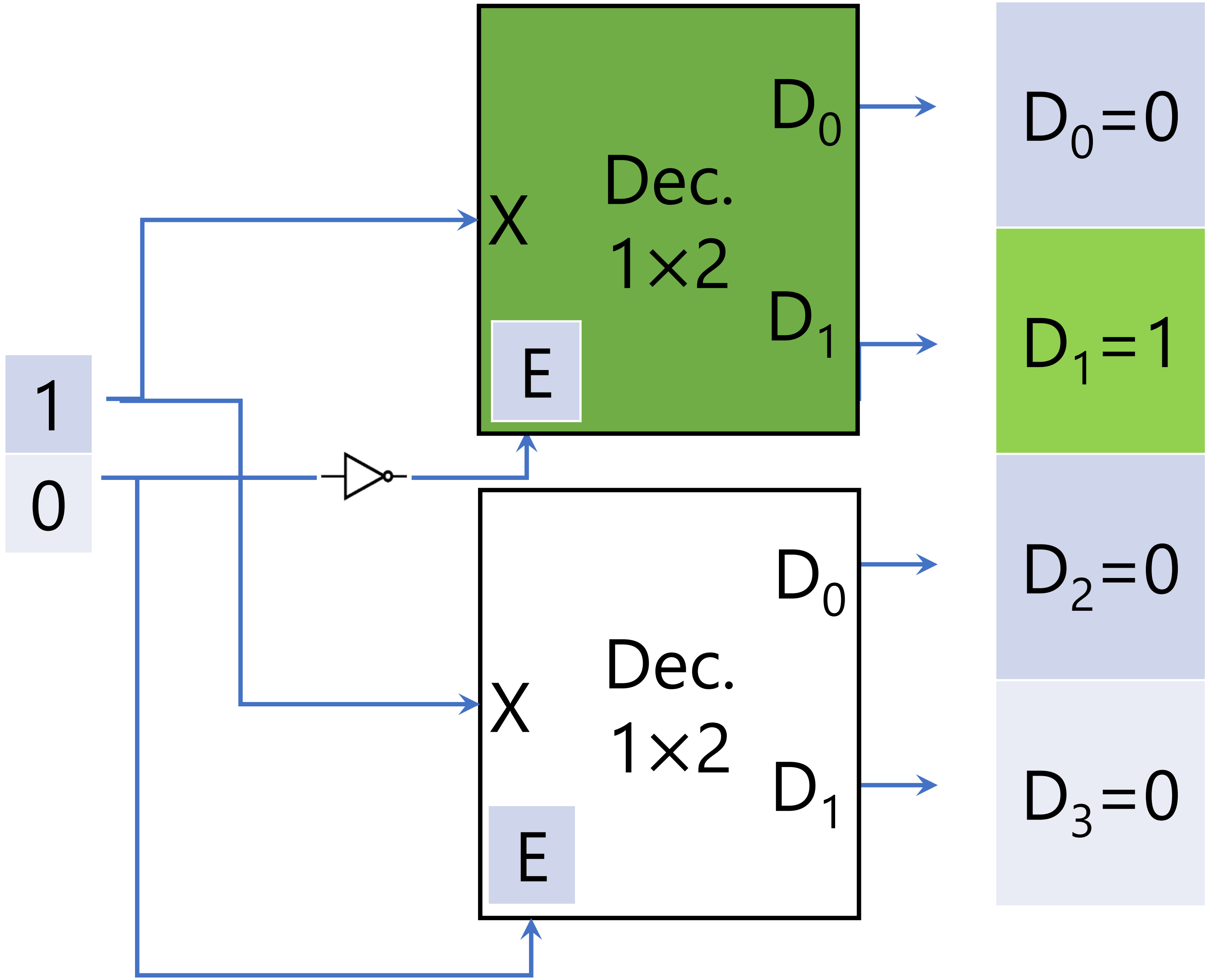


X
Y
X
Dec.
1×2
D_0
D_1
E
X
Dec.
1×2
D_0
D_1
E
D_0
D_1
D_2
D_3

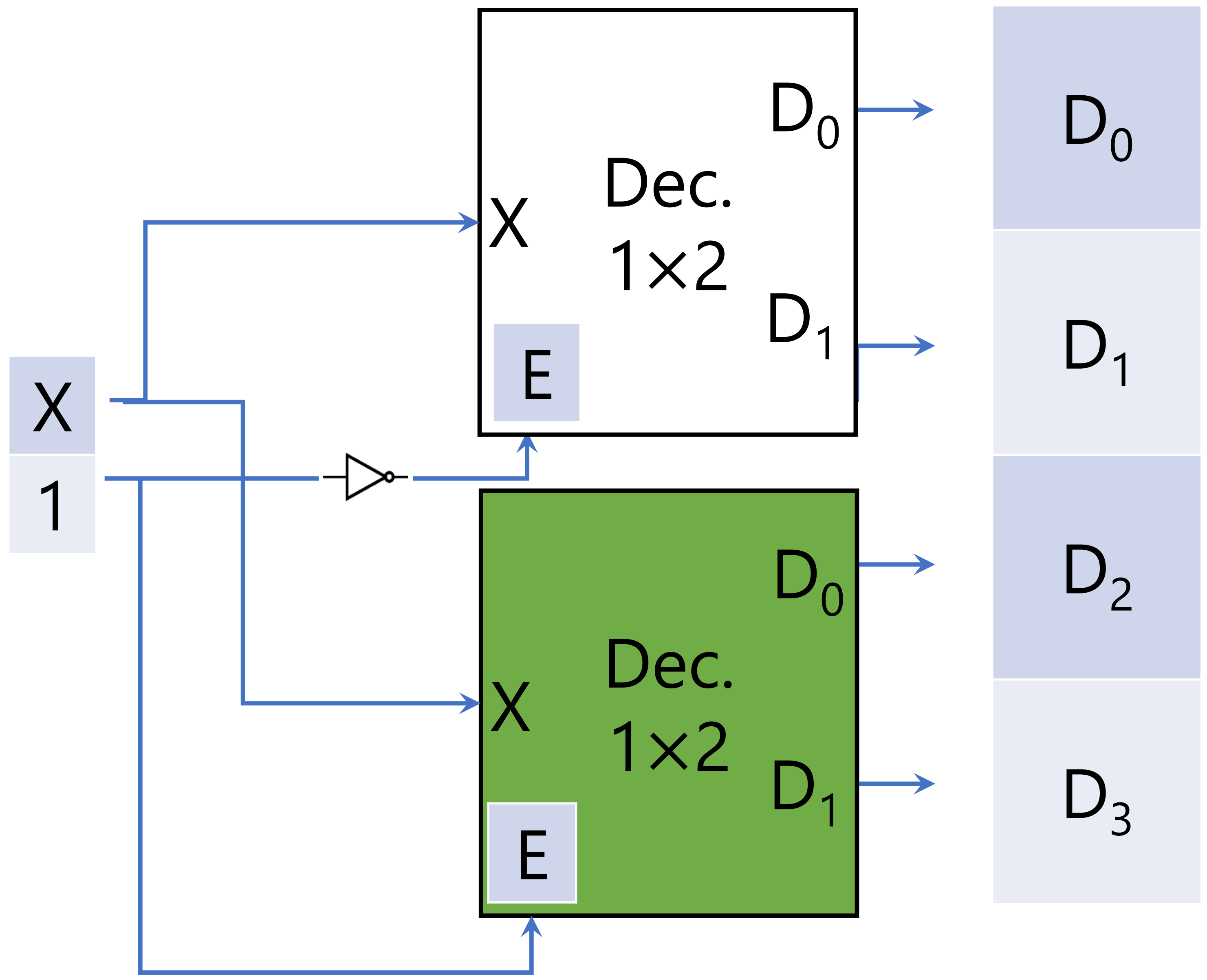

X
0
X
Dec.
1×2
E
D0
D1
X
Dec.
1×2
E
D0
D1
D0
D1
D2
D3

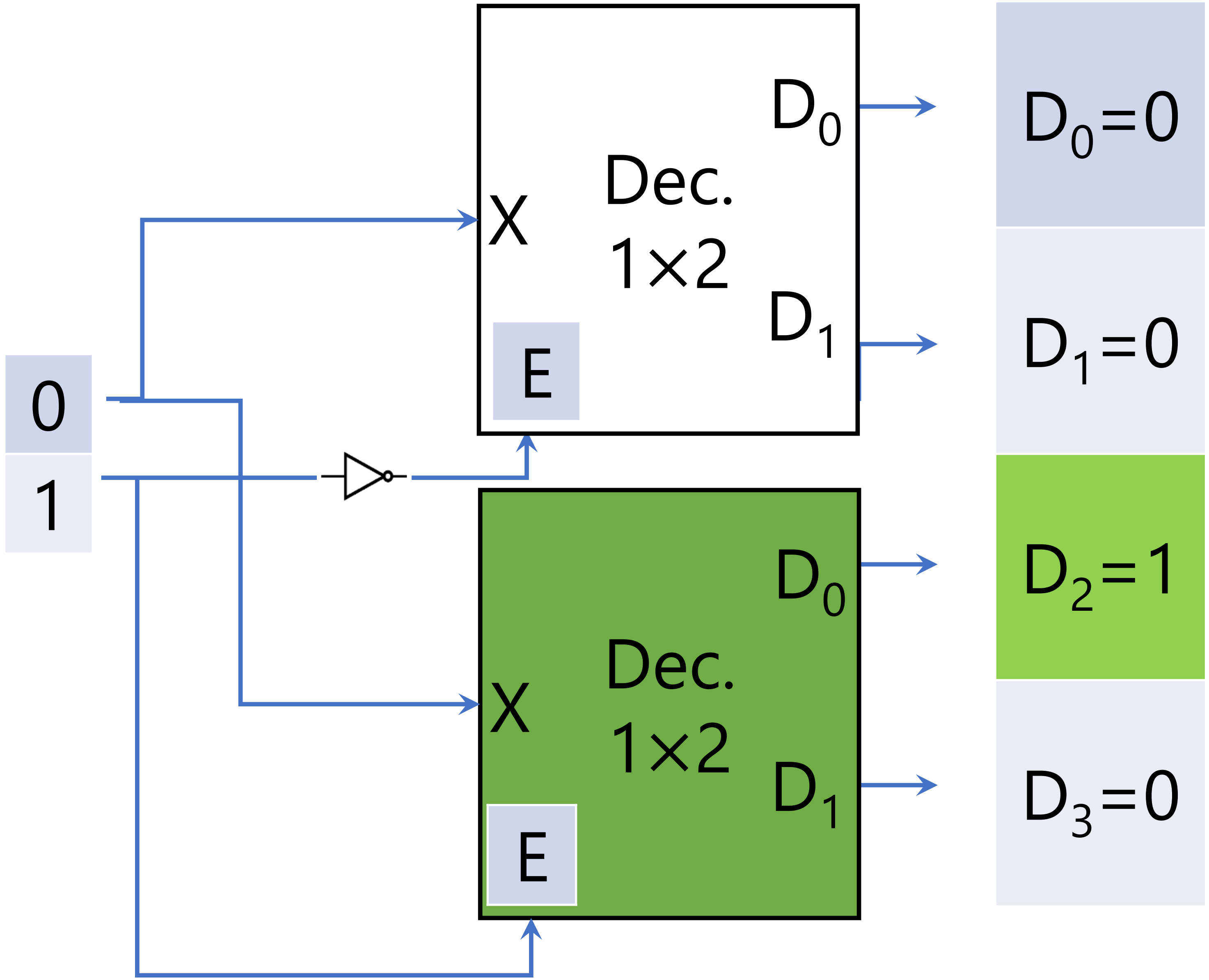


0
0
X
Dec.
1×2
E
D0
D1
X
Dec.
1×2
E
D0
D1
D0=1
D1=0
D2=0
D3=0

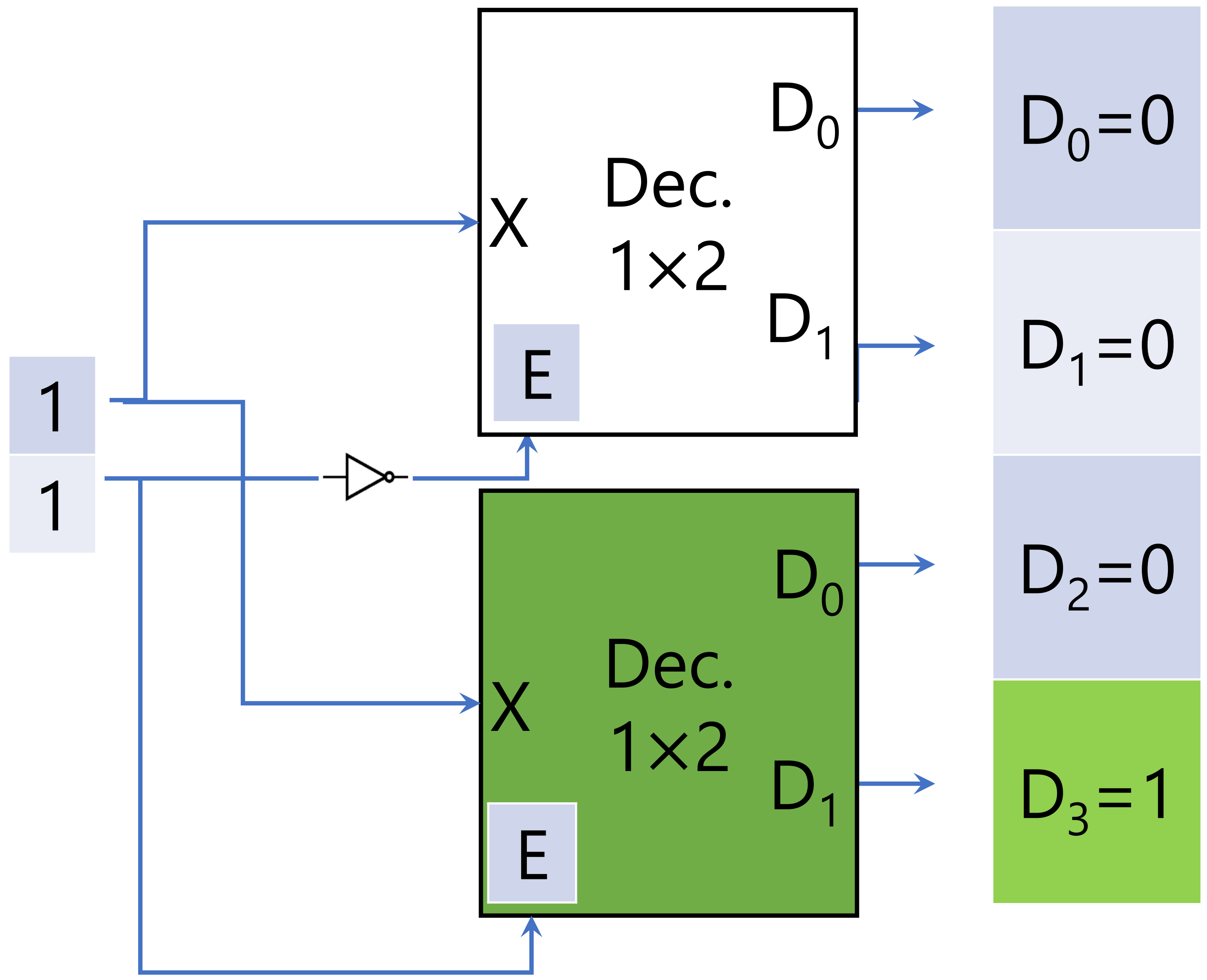


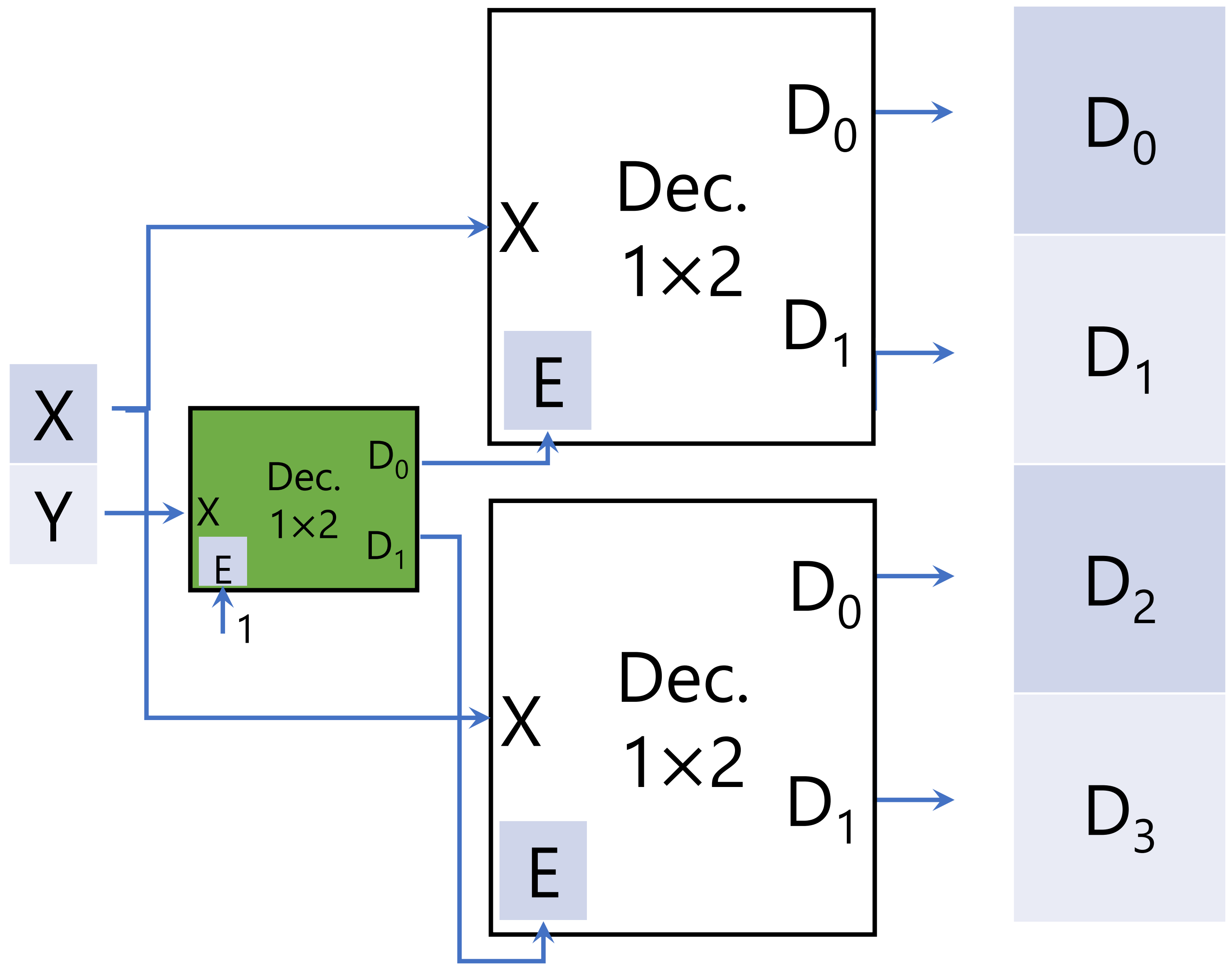
1
0
X
Dec.
1×2
E
D0
D1
X
Dec.
1×2
E
D0
D1
D0=0
D1=1
D2=0
D3=0


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

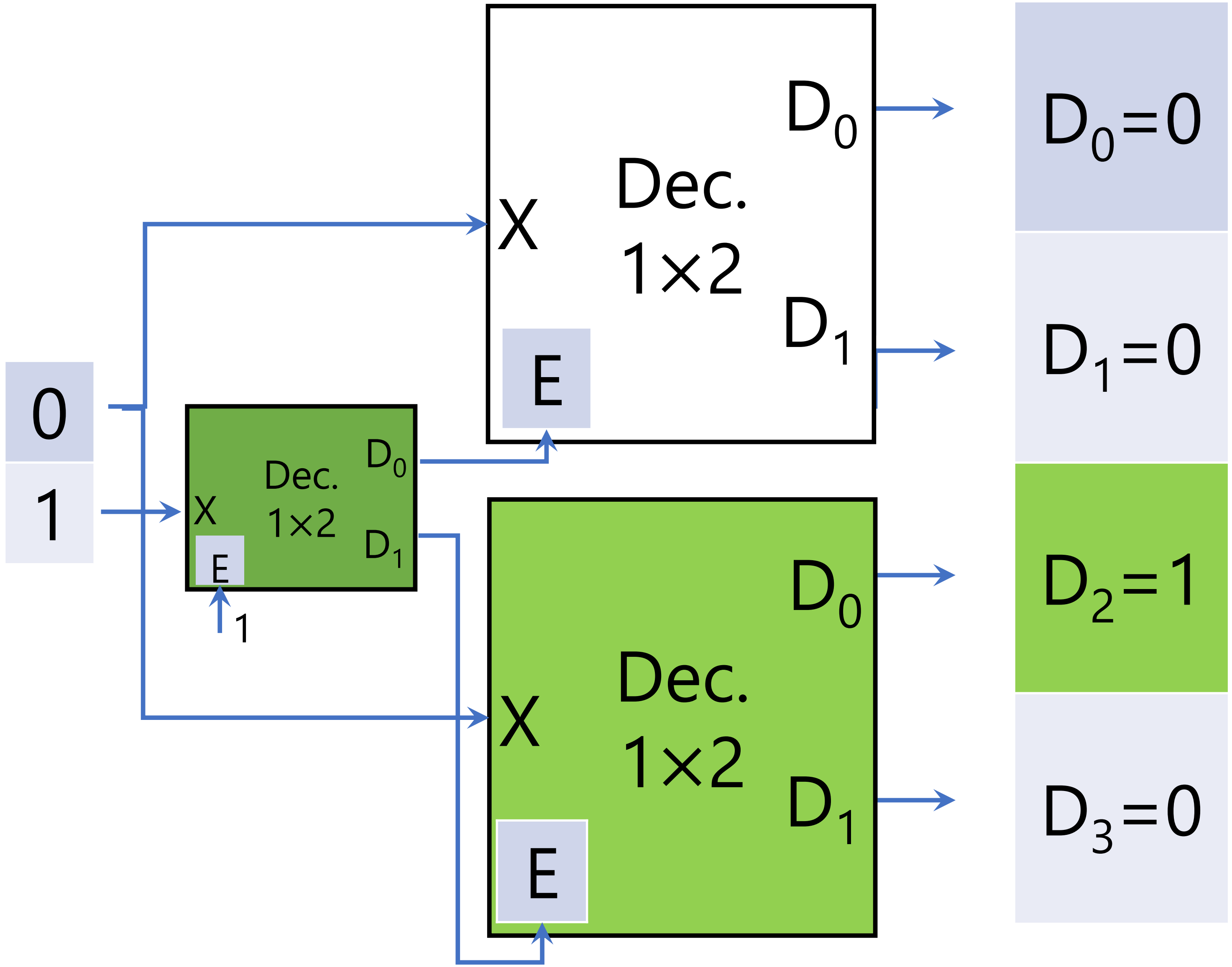


0
1
X
Dec.
1×2
E
D0
D1
X
Dec.
1×2
E
D0
D1
D0=0
D1=0
D2=1
D3=0


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



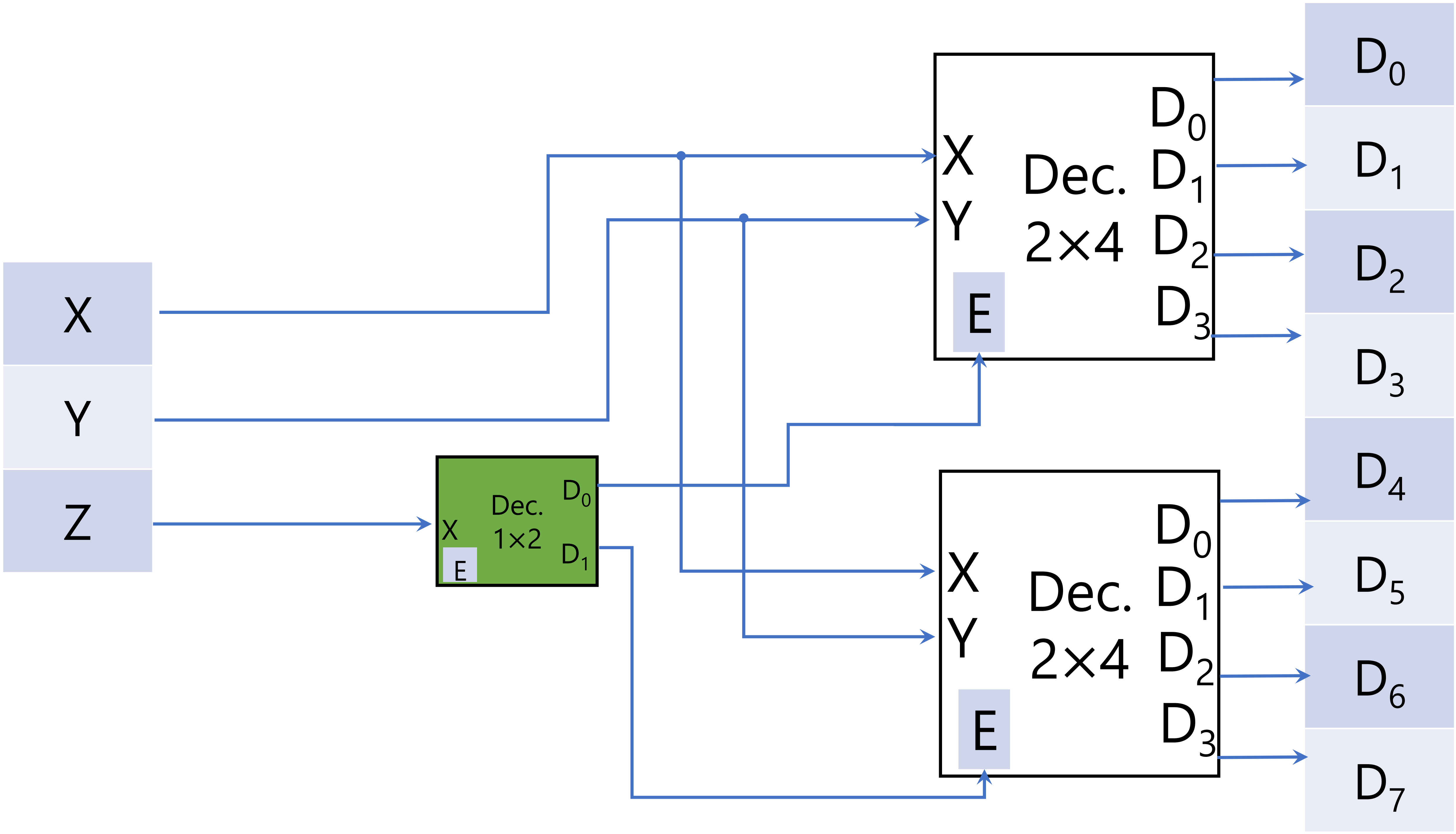
X
Y
X
Dec.
1×2
D0
D1
E
1
X
Dec.
1×2
D0
D1
E
X
Dec.
1×2
D0
D1
E
D0
D1
D2
D3


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

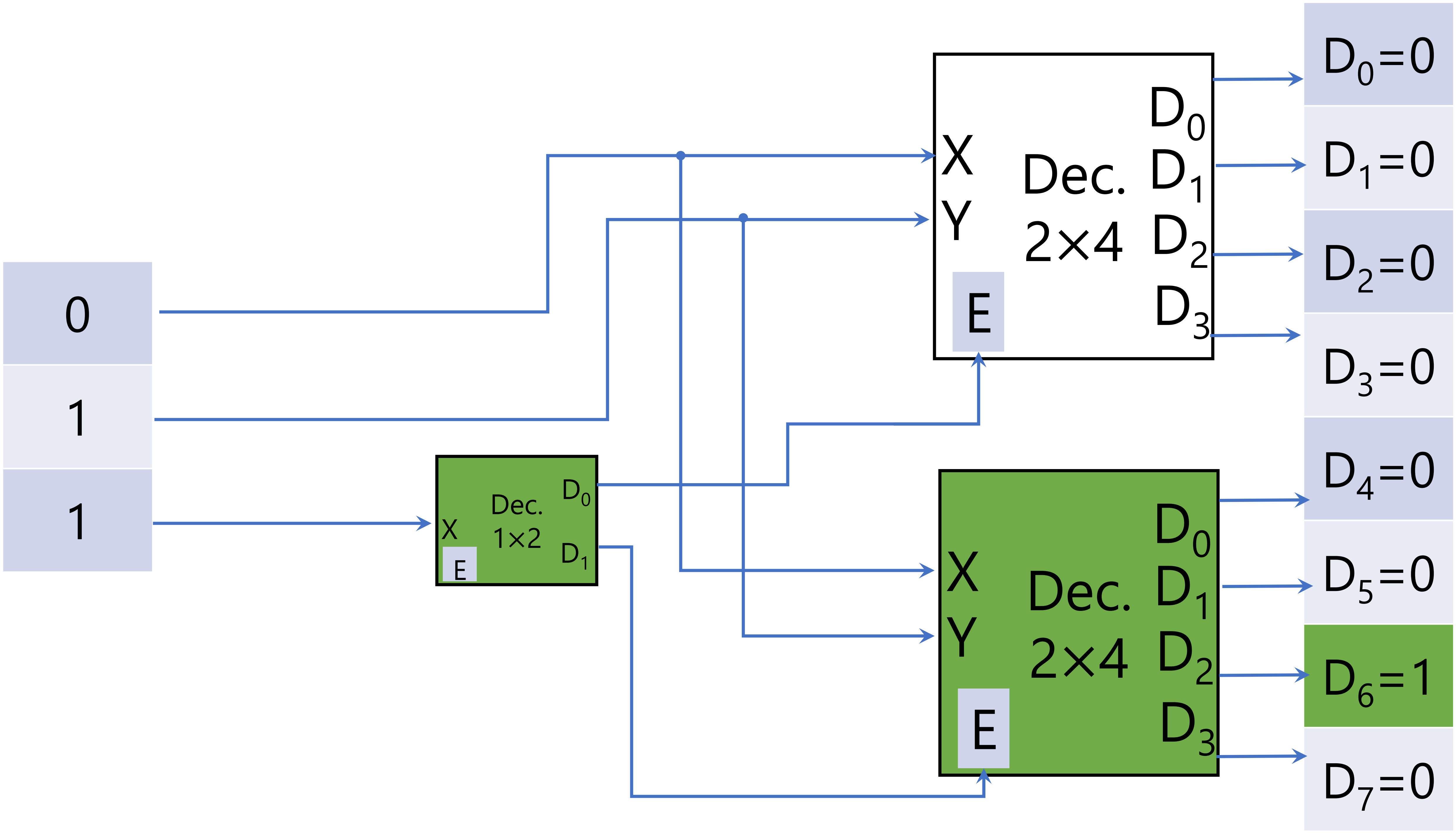
Decoder

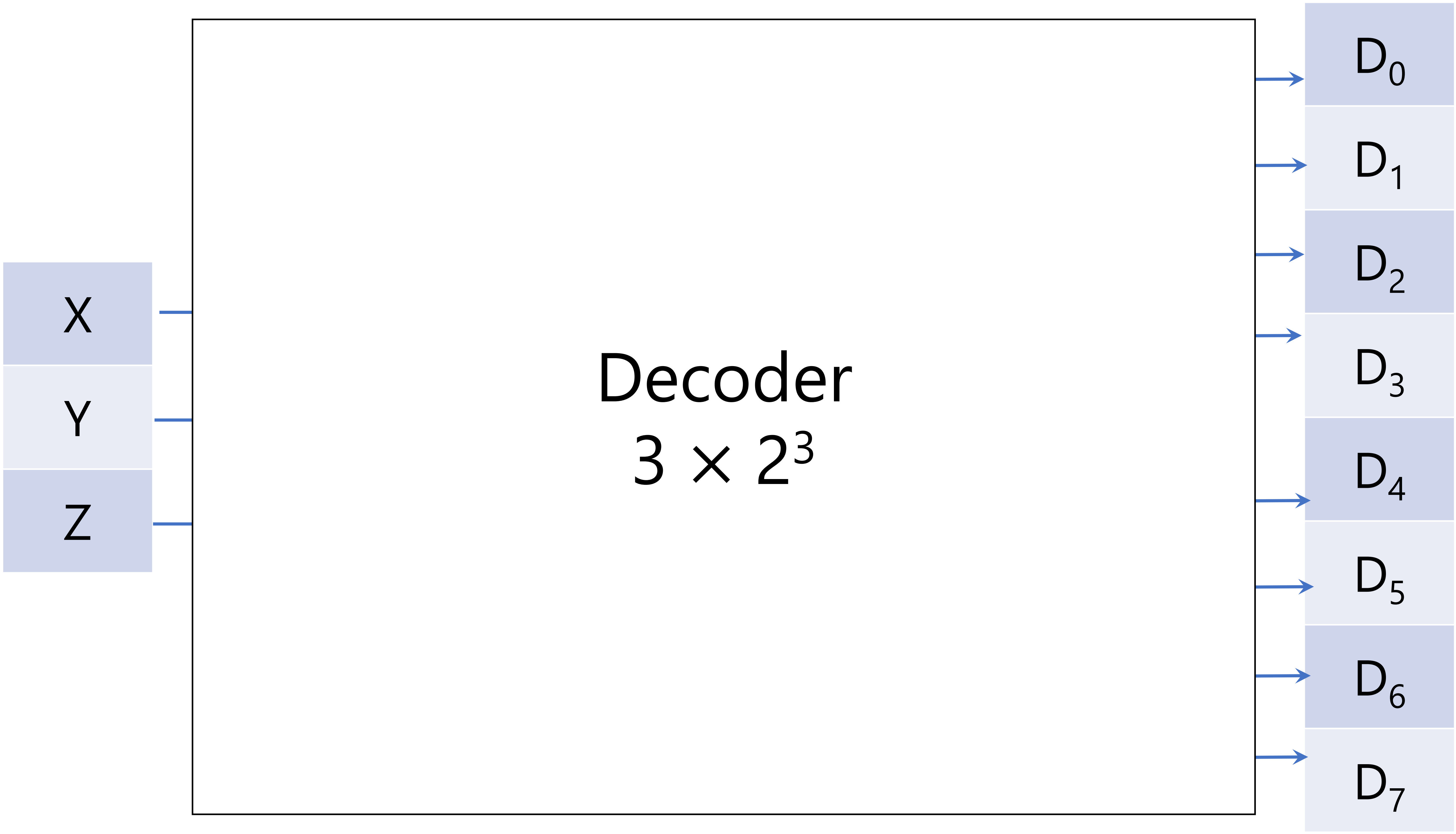
Decode 3-Bit Binary to 2^3 One-hot

Re-Use 2×2^2 Decoder



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


0
1
1
X
Dec.
1×2
E
D0
D1
X
Y
Dec.
2×4
E
D0
D1
D2
D3
X
Y
Dec.
2×4
E
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 × 2^3
D0
D1
D2
D3
D4
D5
D6
D7

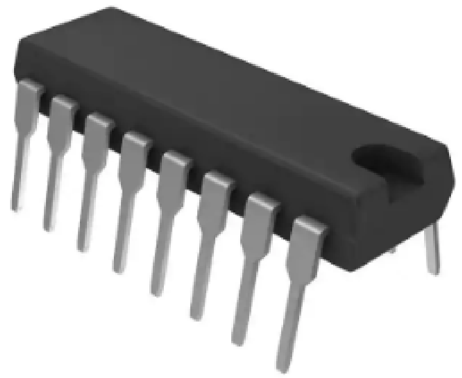
Decoder

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
Detailed Description	Decoder/Demultiplexer 1 x 3:8 16-PDIP

Customer Reference

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

Media & Downloads

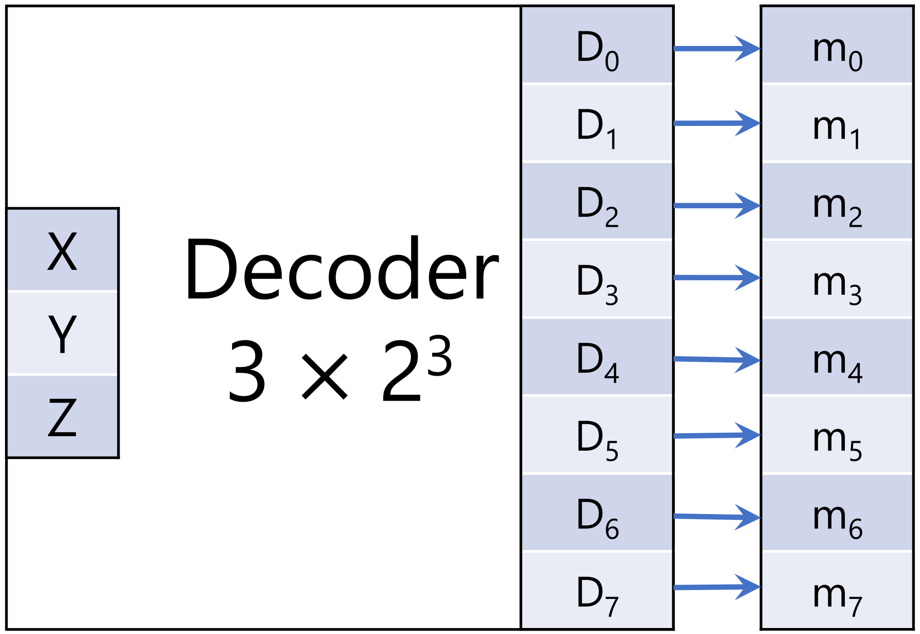
RESOURCE TYPE	LINK
Datasheets	SN54LS138, SN54S138, SN74LS138, SN74S138A
Featured Product	Logic Solutions Analog Solutions
PCN Design/Specification	Material Set 30/Mar/2017
EDA / CAD Models	SN74LS138N by SnapEDA SN74LS138N by Ultra Librarian

Decoder

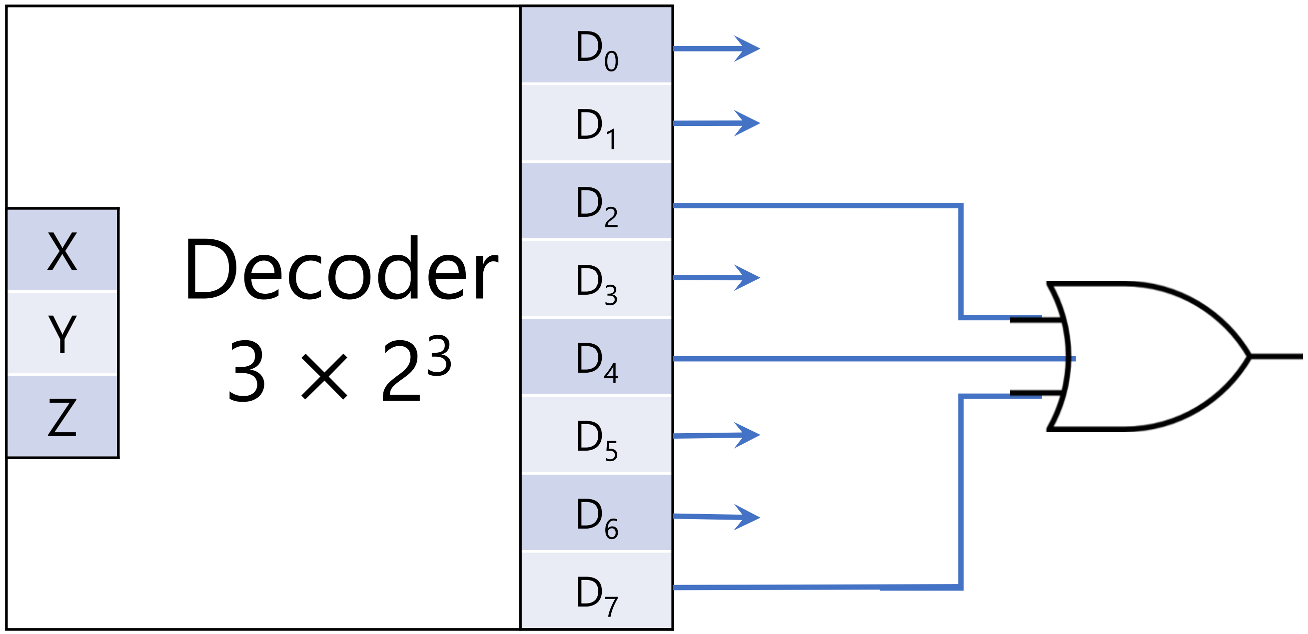
Boolean Function

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

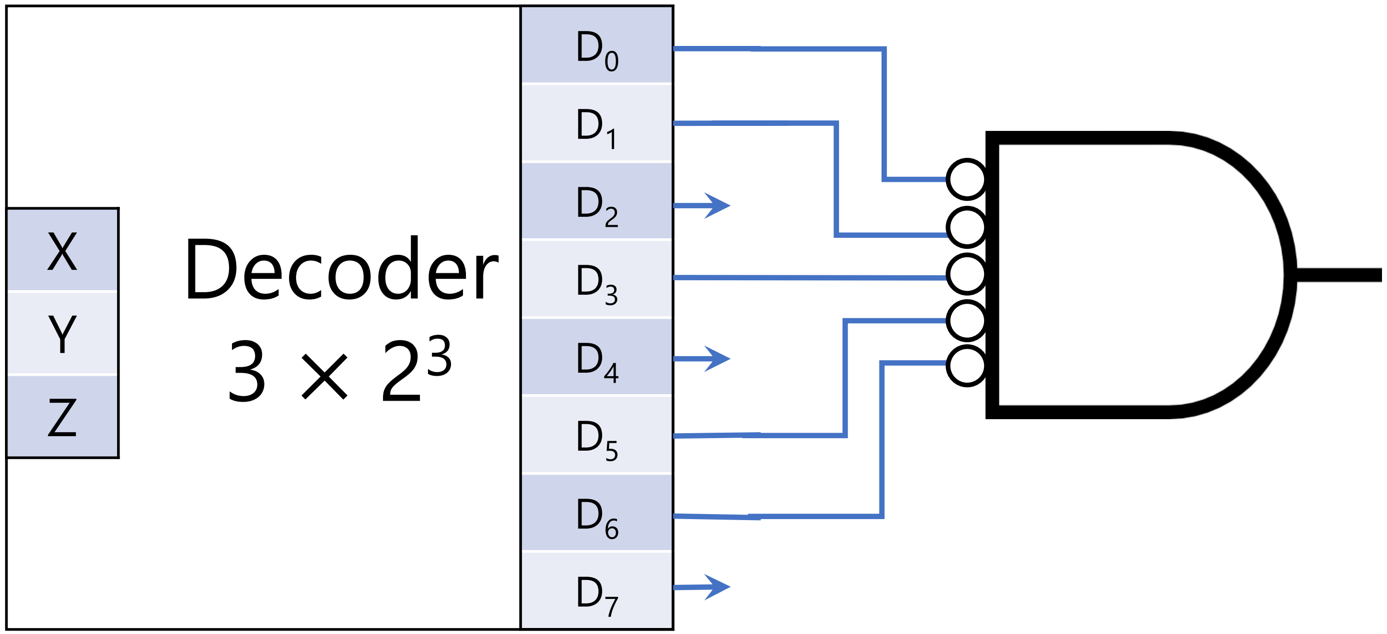
$$F_{PoS} = \prod M(\ldots)$$



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

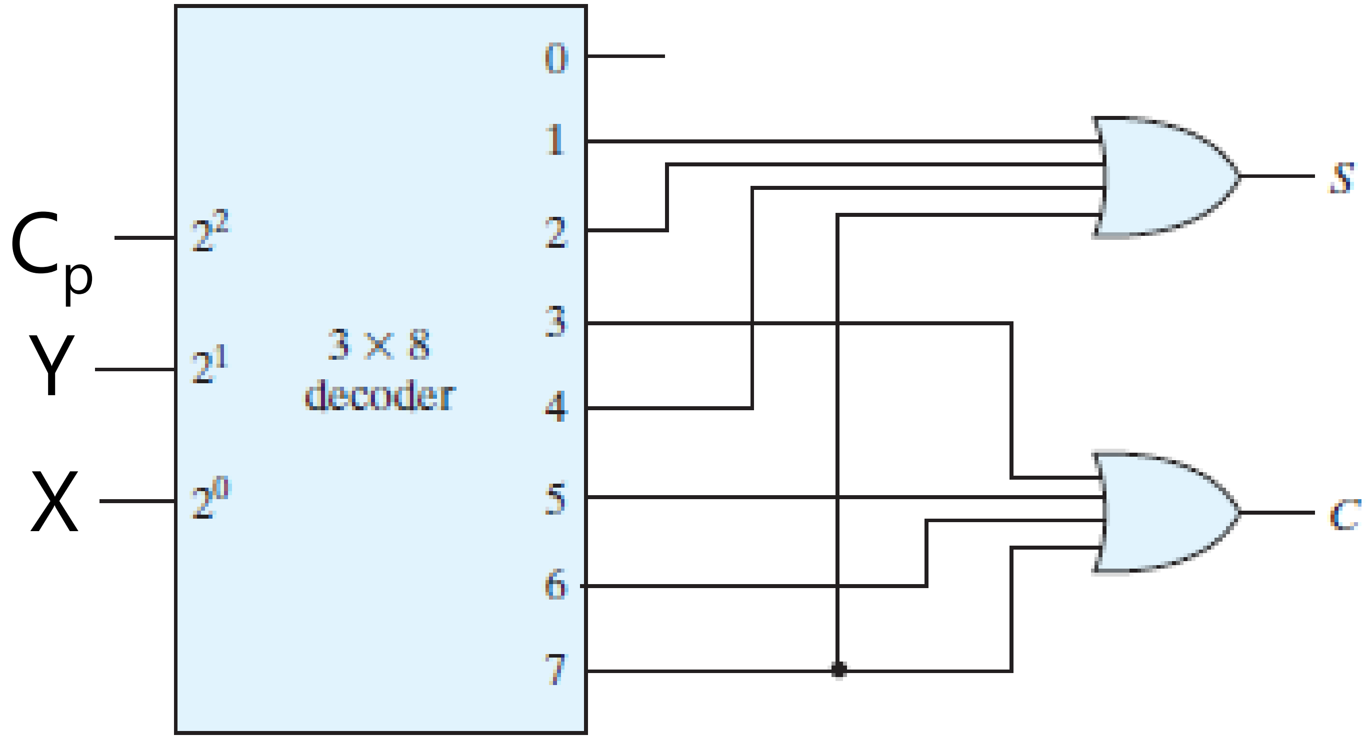
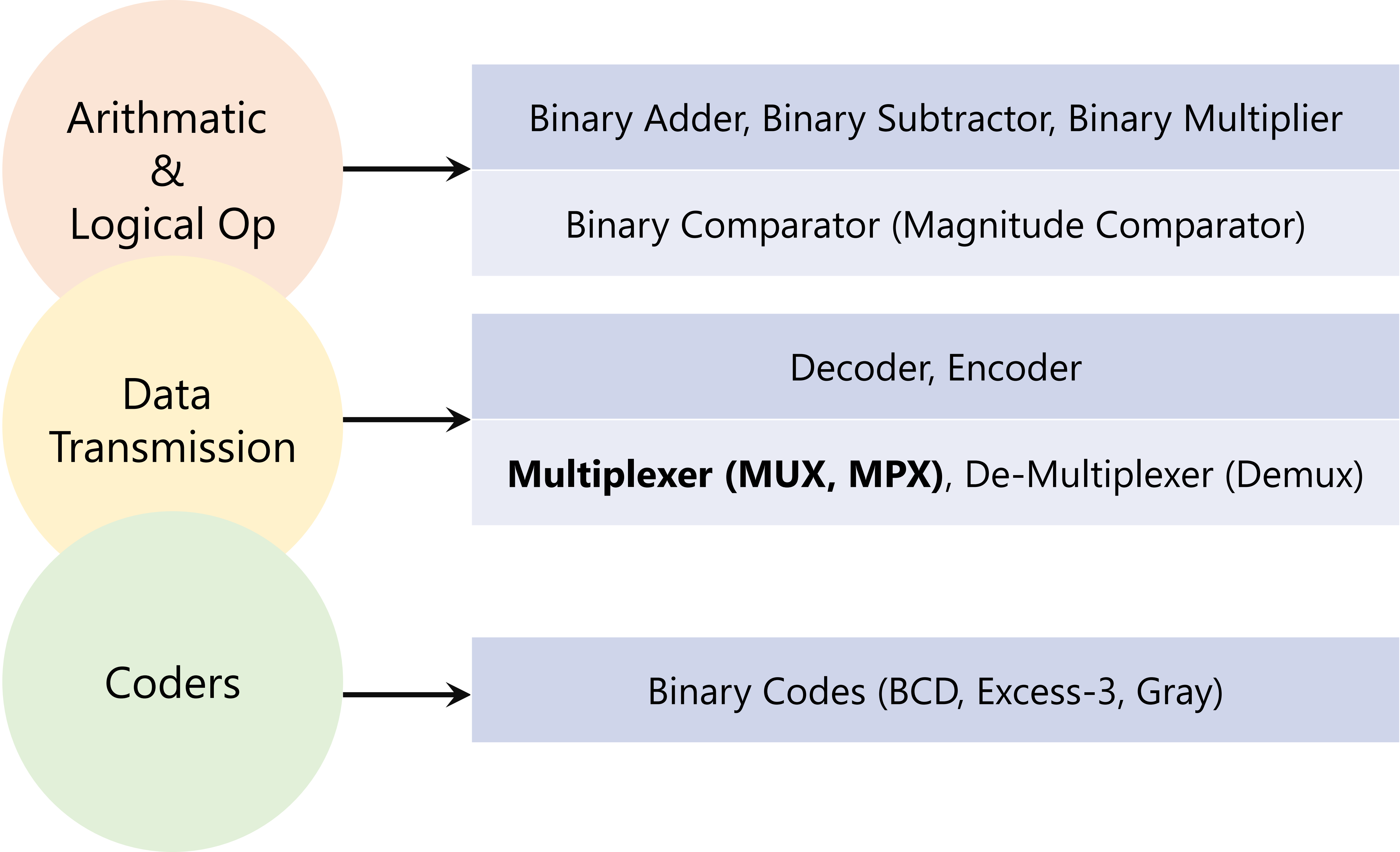


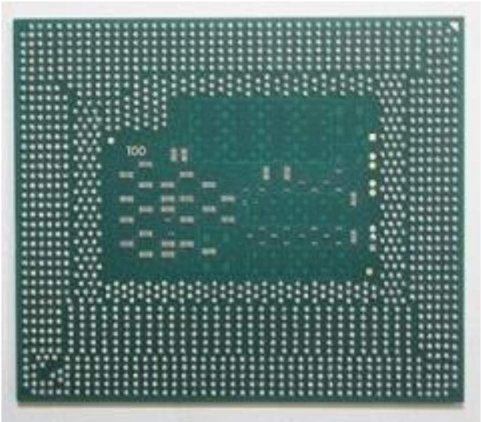
FIGURE 4.21
Implementation of a full adder with a decoder

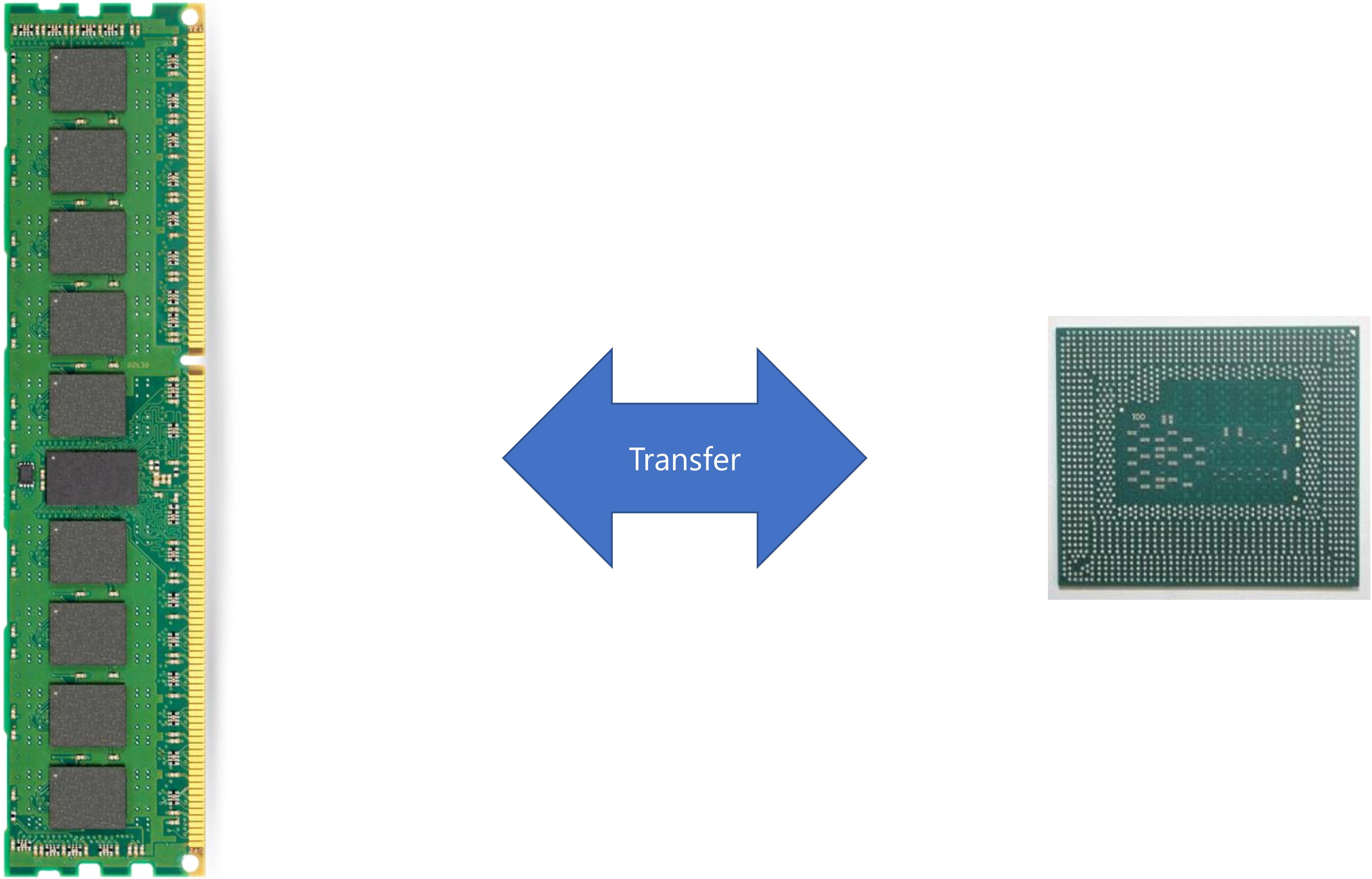

Arithmatic
&
Logical Op
Binary Adder, Binary Subtractor, Binary Multiplier
Binary Comparator (Magnitude Comparator)
Data
Transmission
Decoder, Encoder
Multiplexer (MUX, MPX), De-Multiplexer (Demux)
Coders
Binary Codes (BCD, Excess-3, Gray)

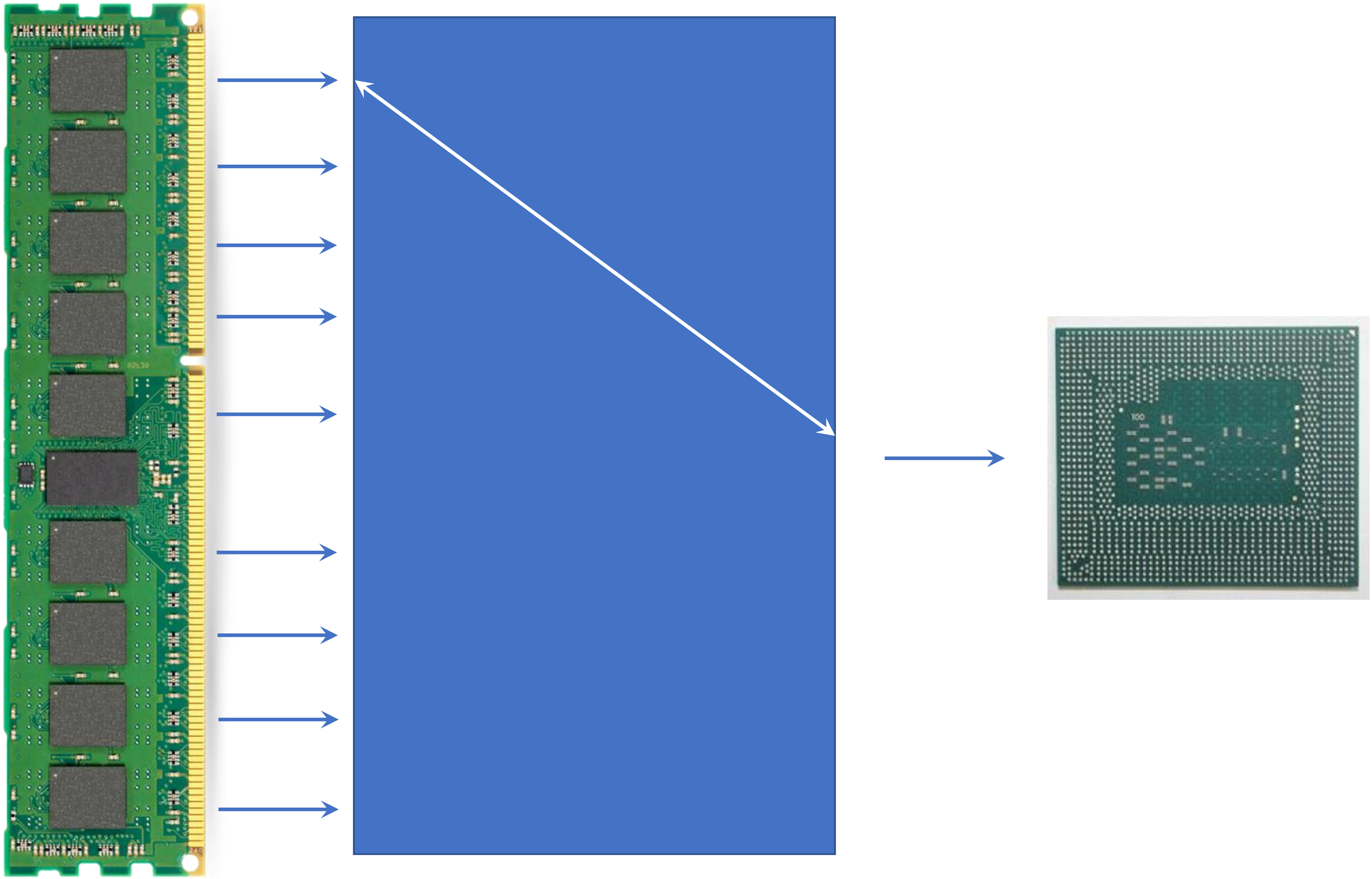
Multiplexer

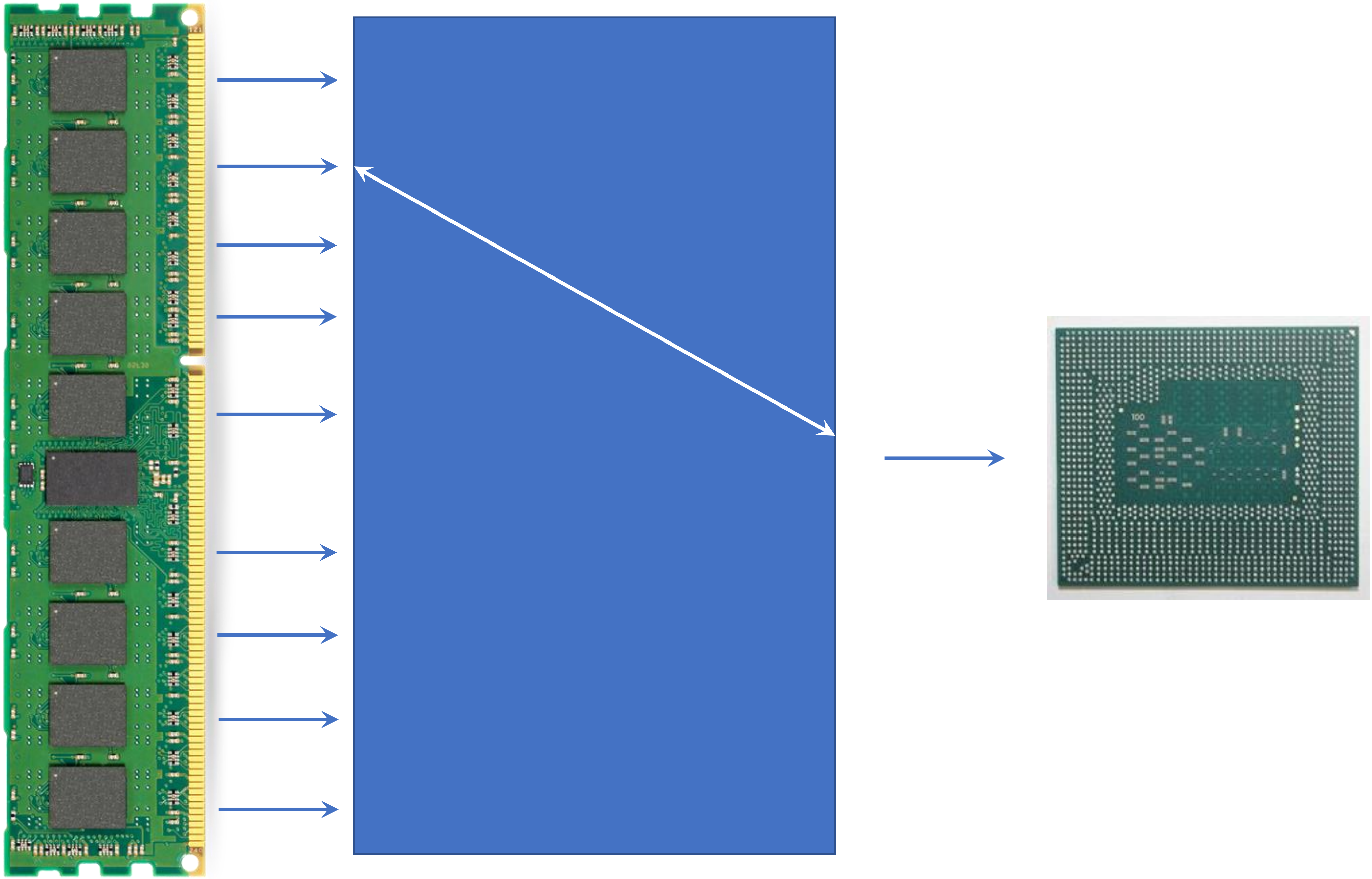
Shortened to MUX or MPX

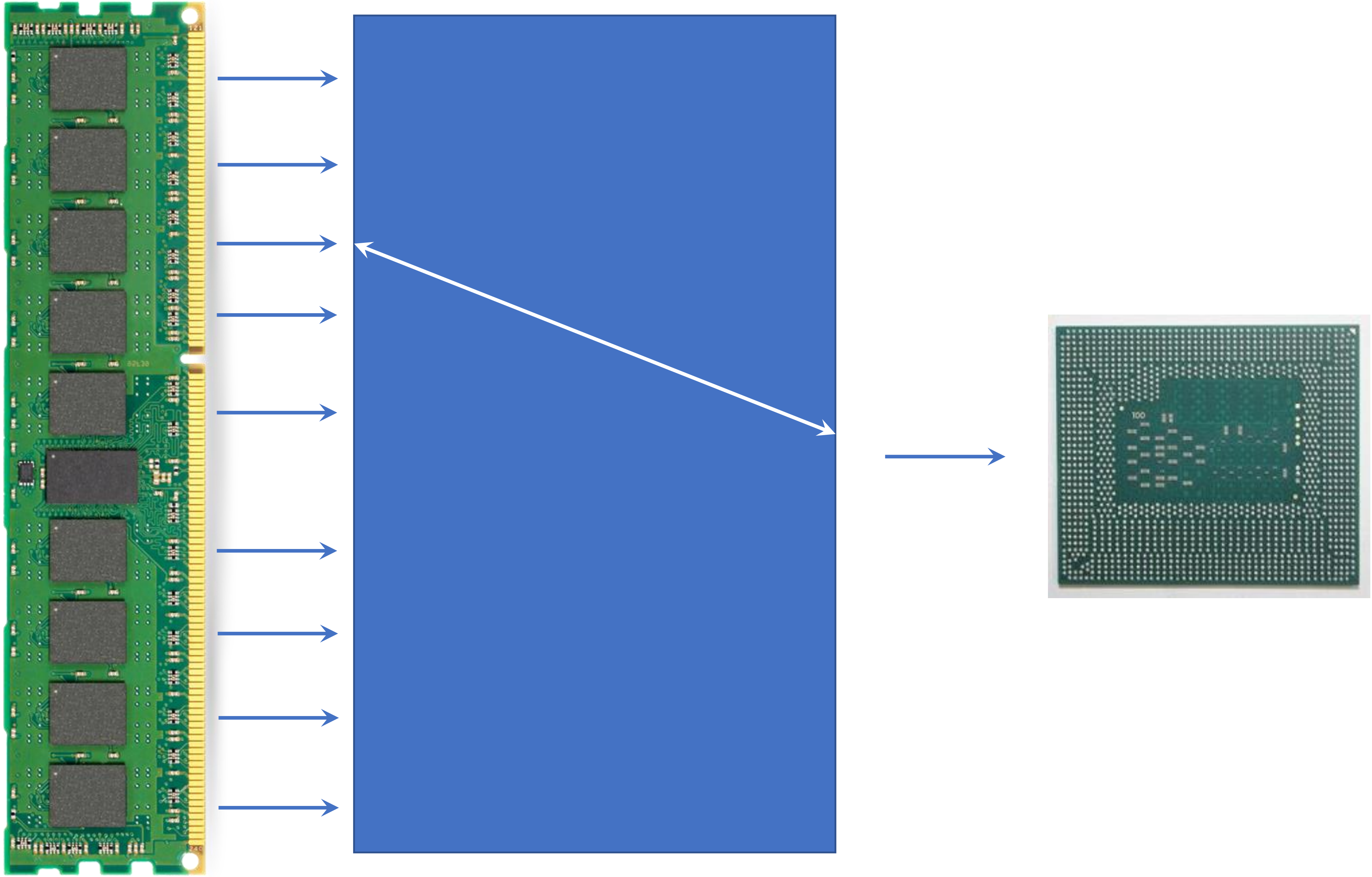
Address to Select

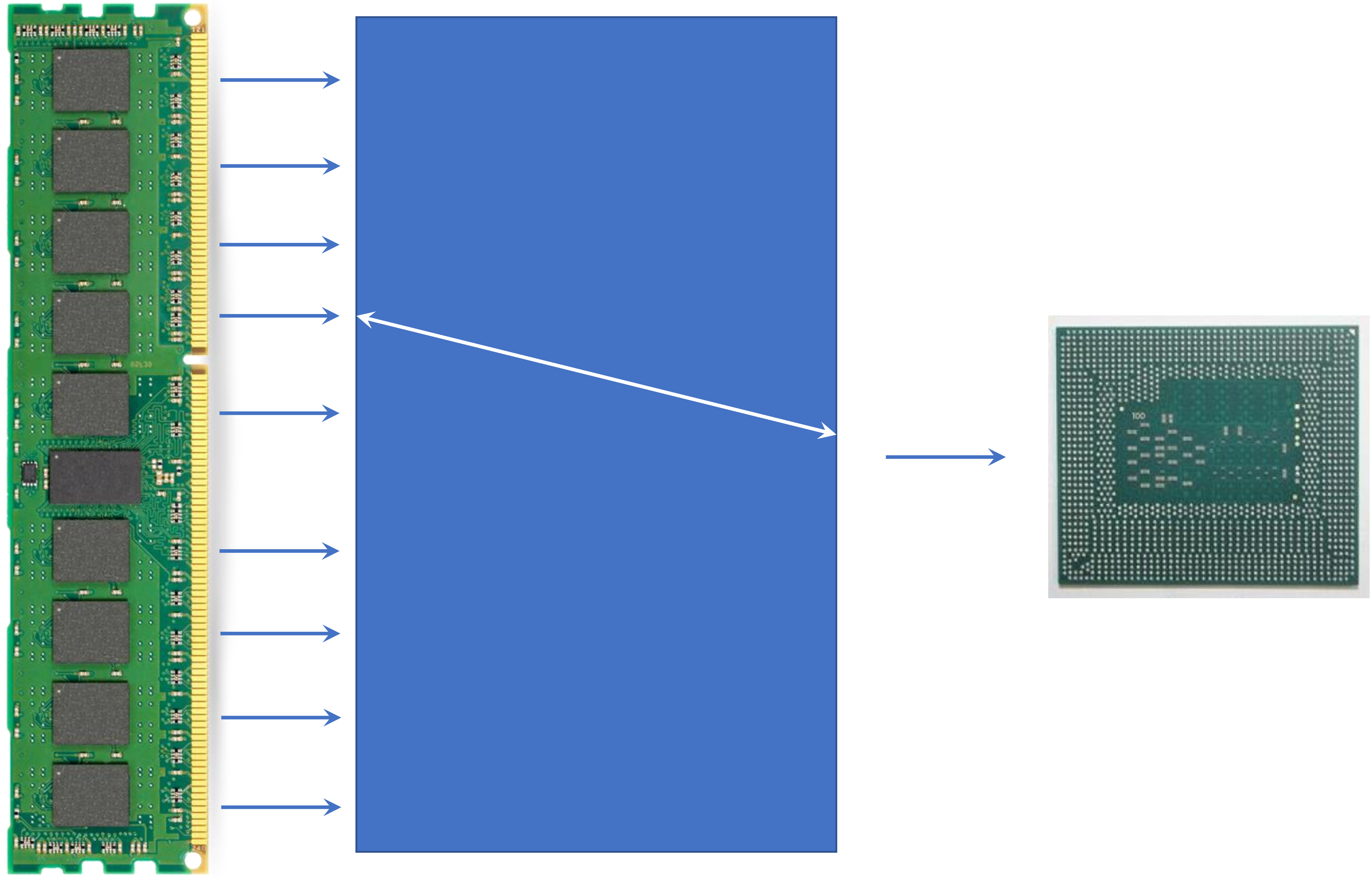


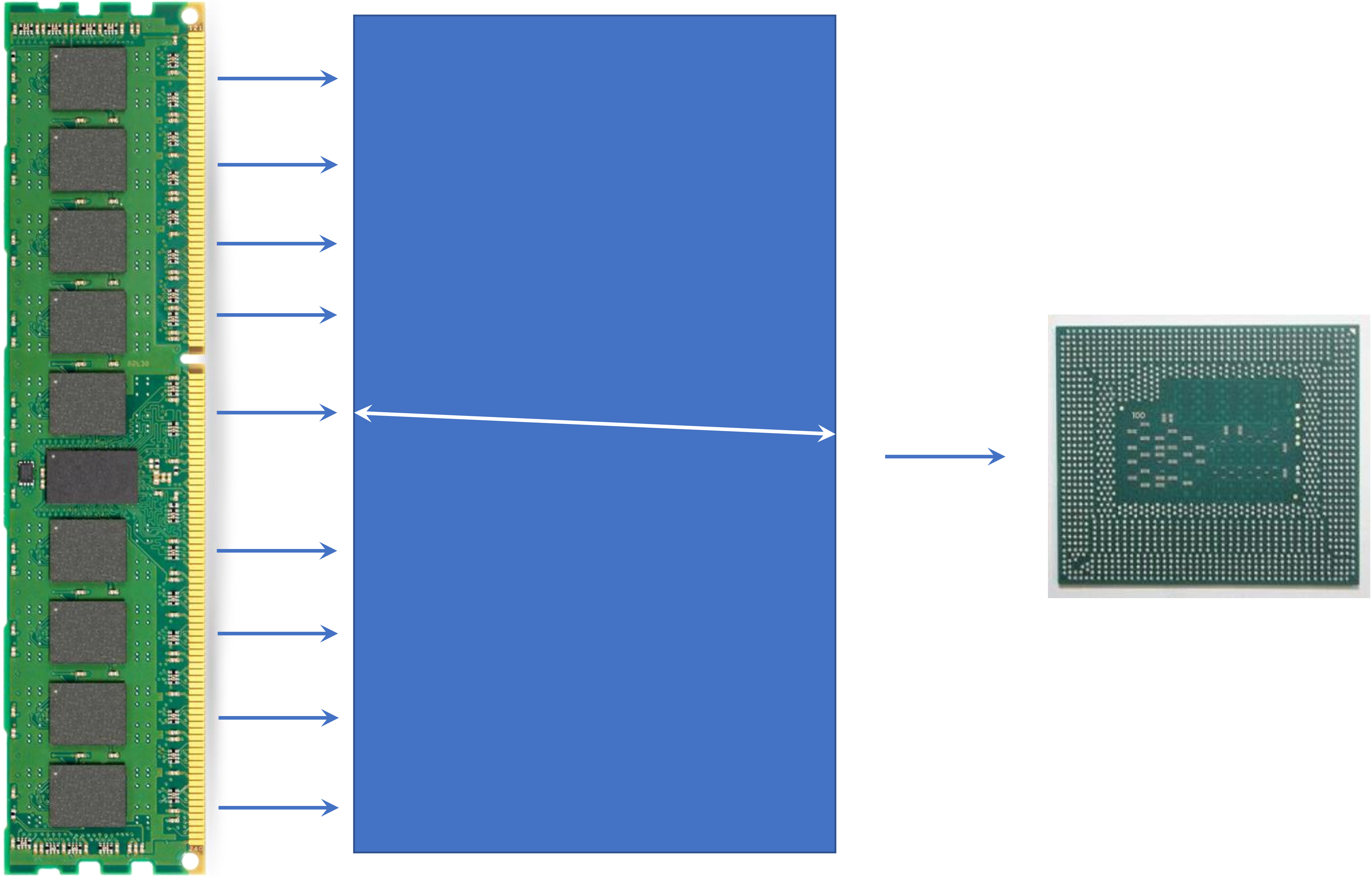

Transfer

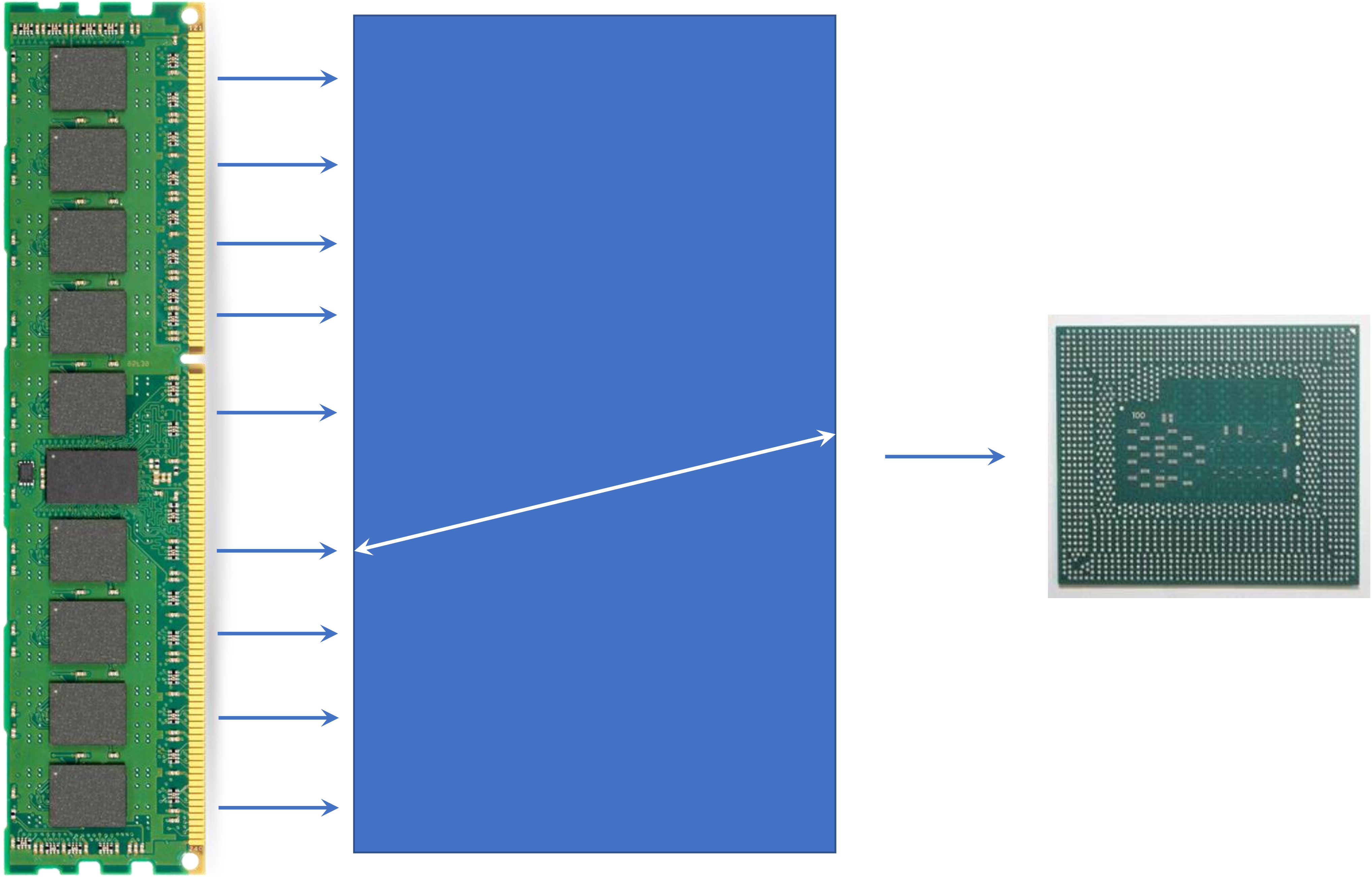


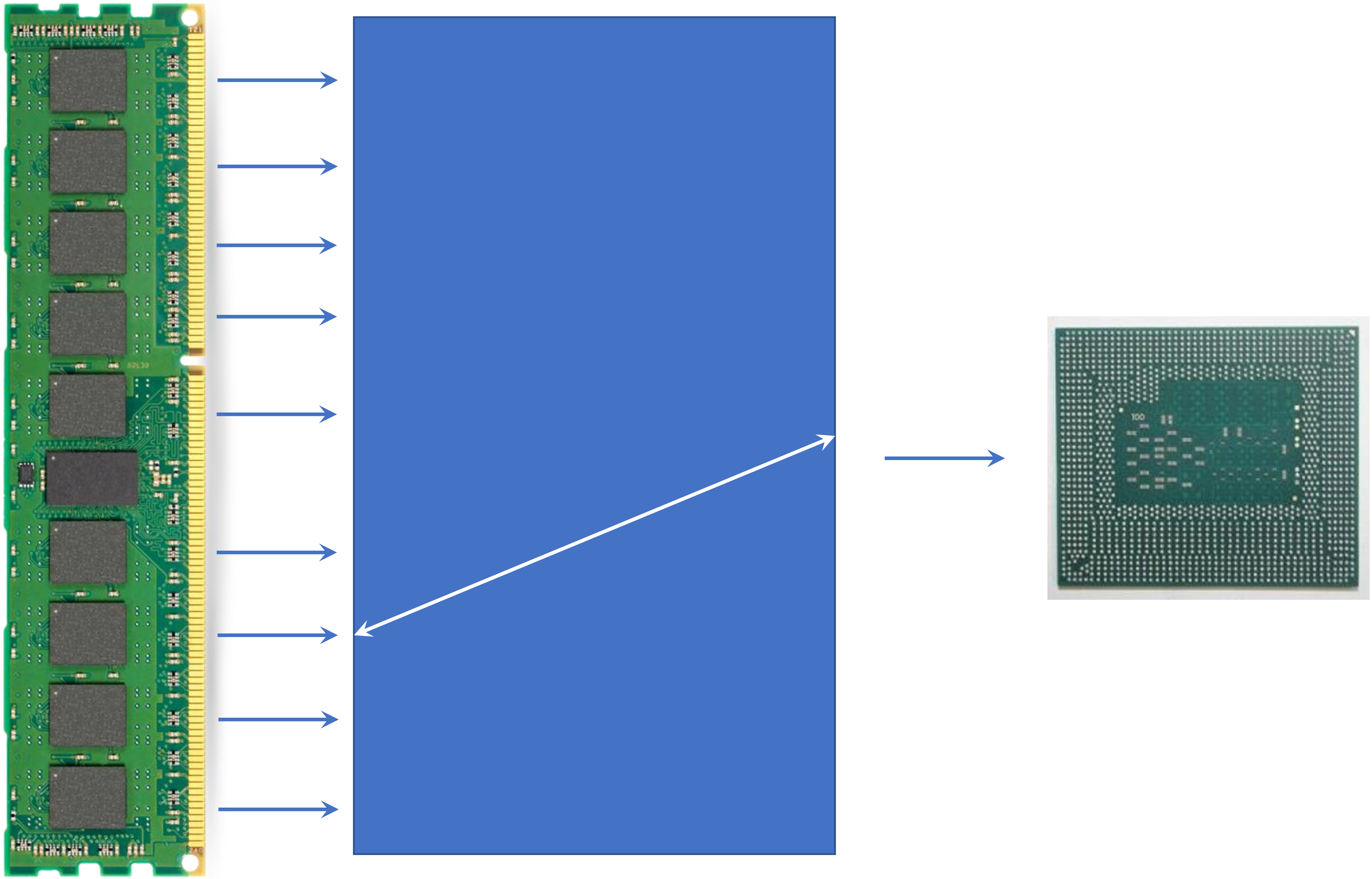


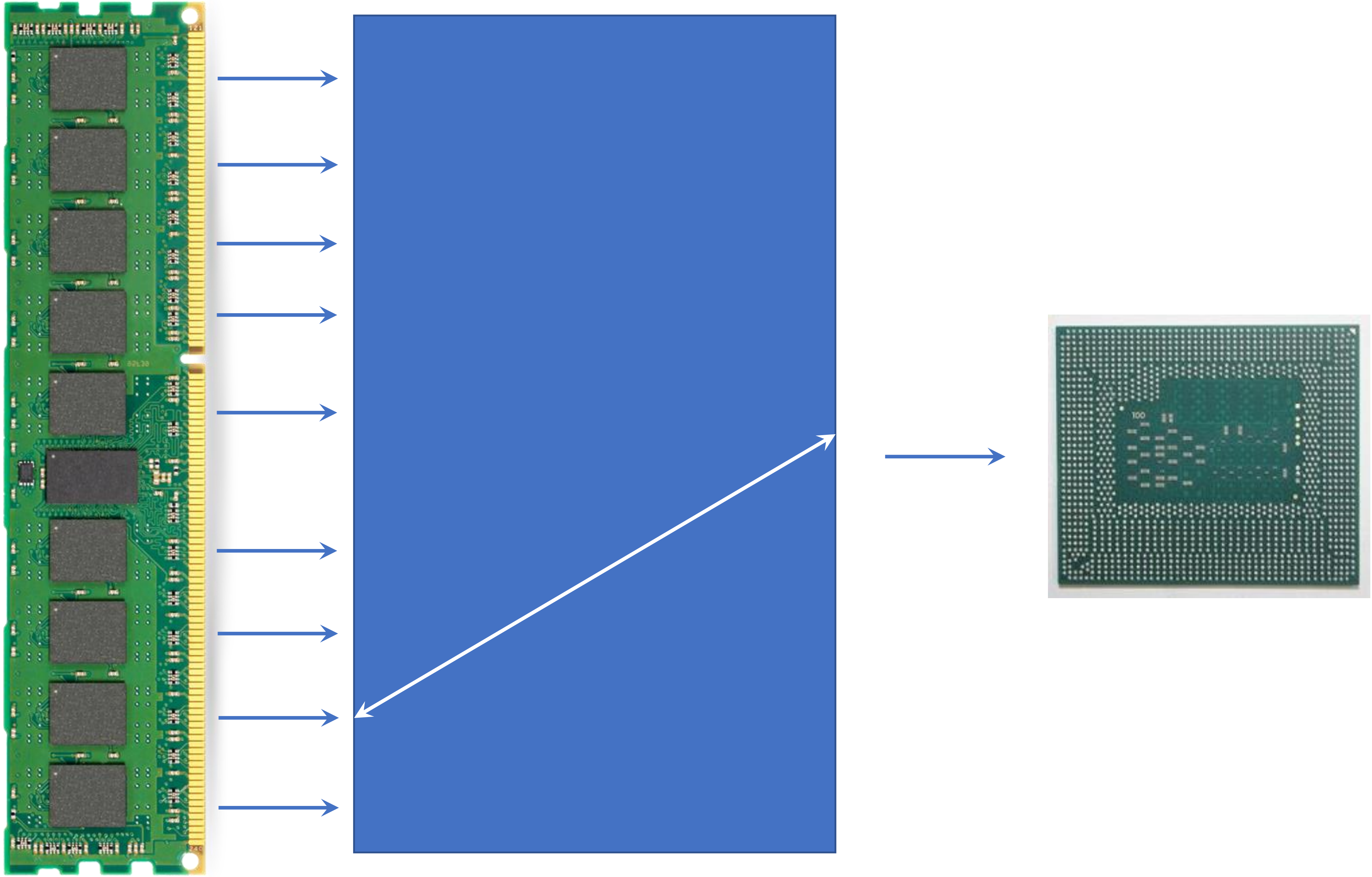


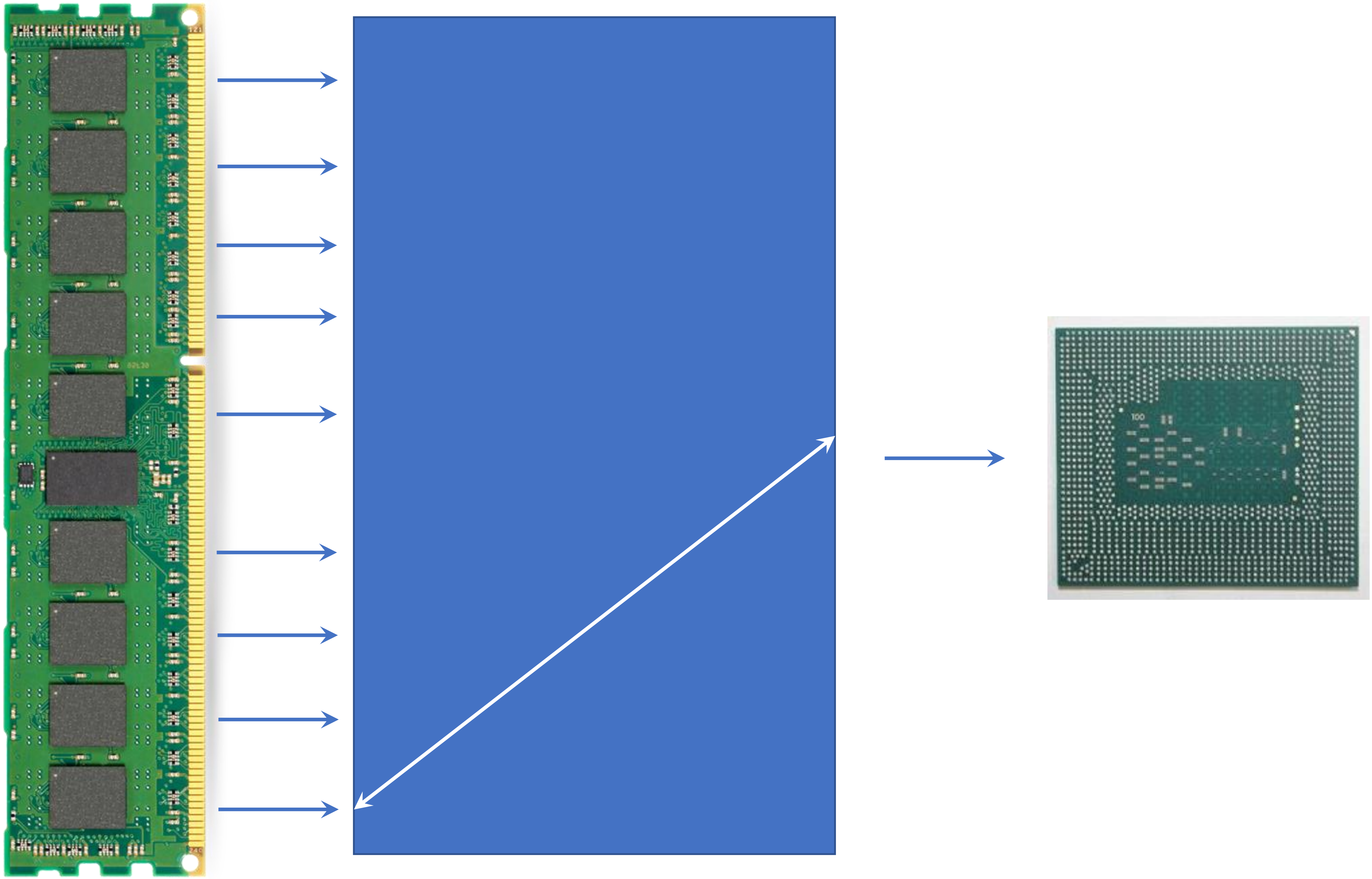




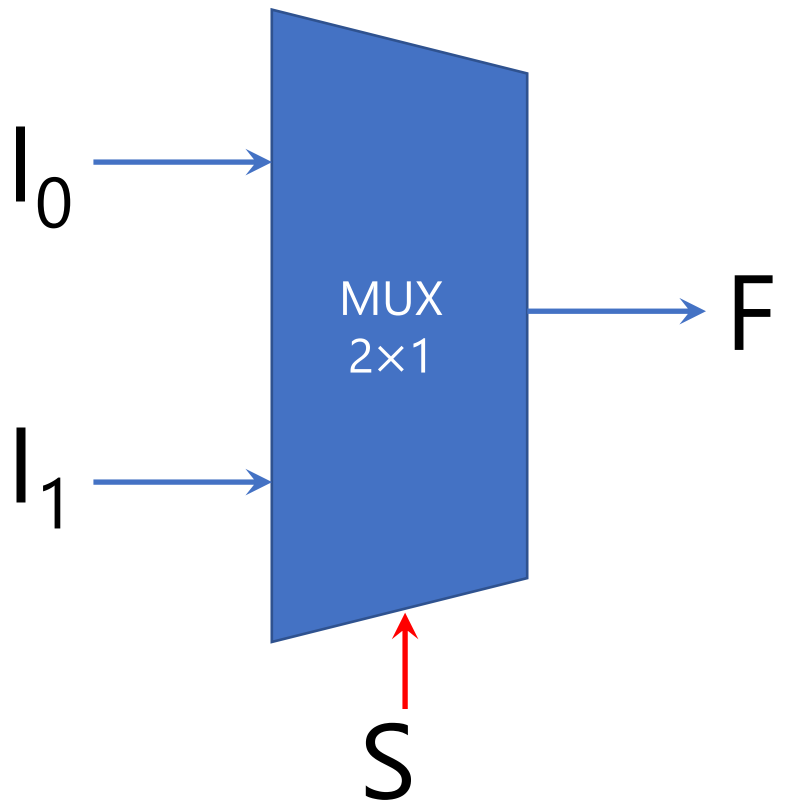


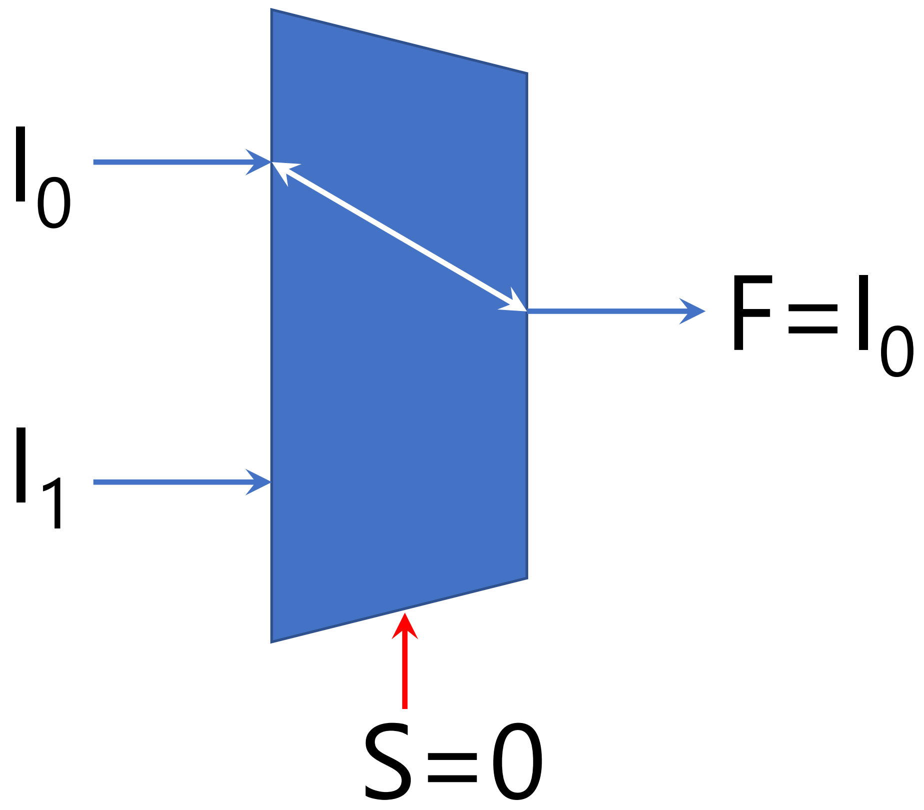




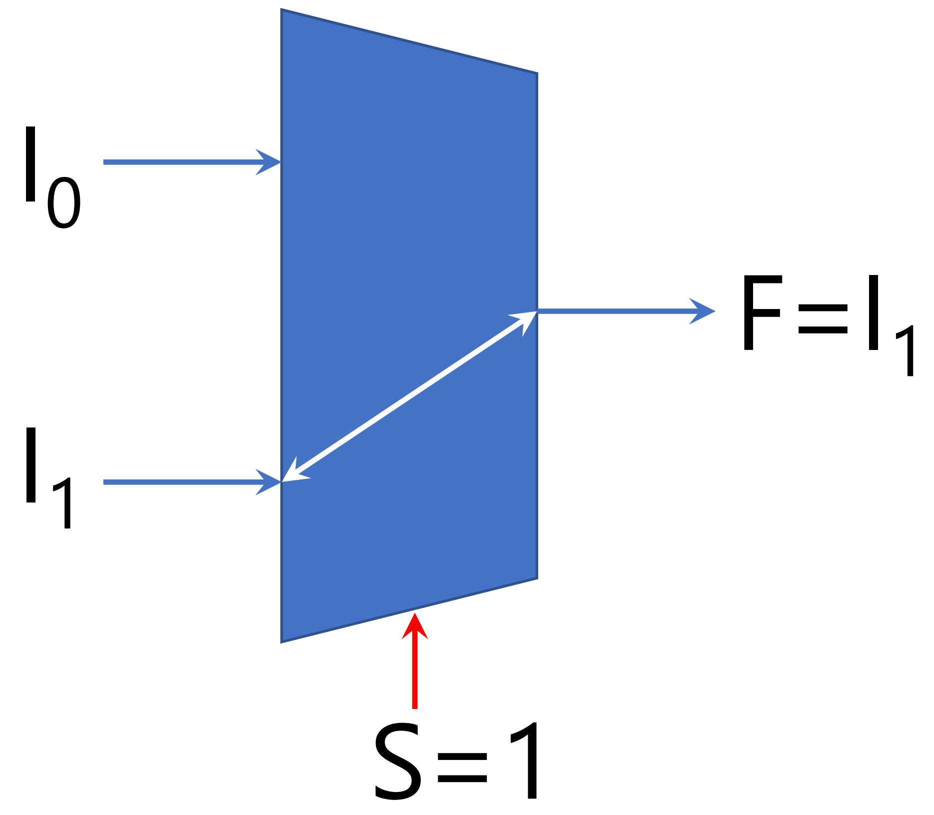


Multiplexer
$2^1 \times 1$


I_0
MUX
2×1
F
I_1
S



I_0
I_1
$F=I_0$
$S=0$



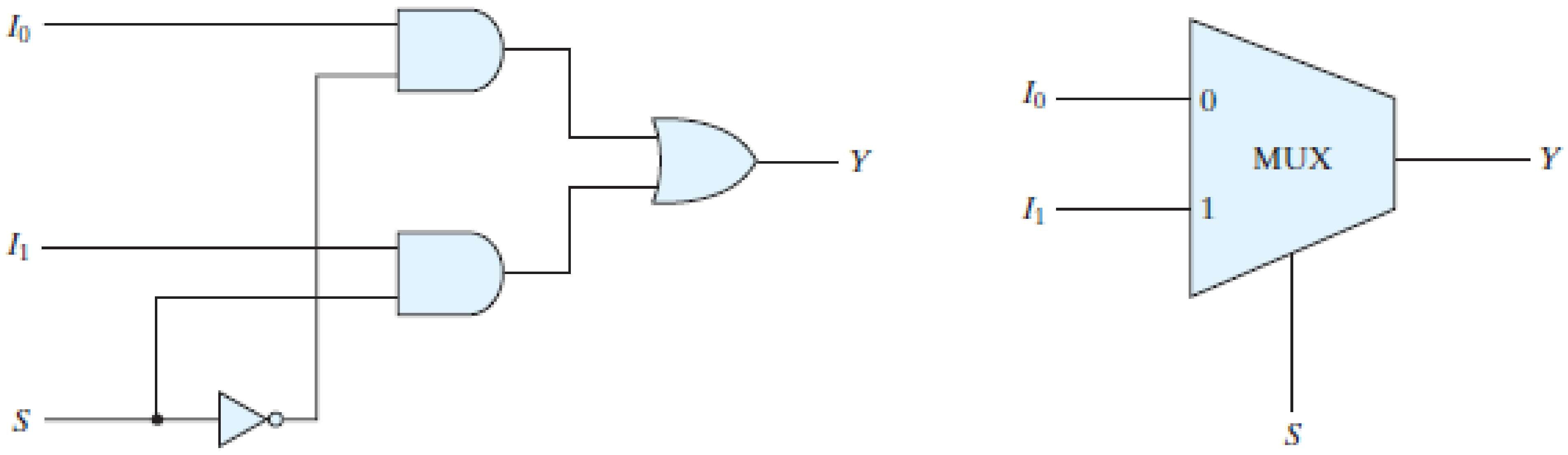
I_0
I_1
$F=I_1$
$S=1$

S	I_1	I_0	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

S	I_1	I_0	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

S \ I_1I_0	00	01	11	10
0	0 m_0	1 m_1	1 m_3	0 m_2
1	0 m_4	0 m_5	1 m_7	1 m_6

$$F=S'I_0+SI_1$$



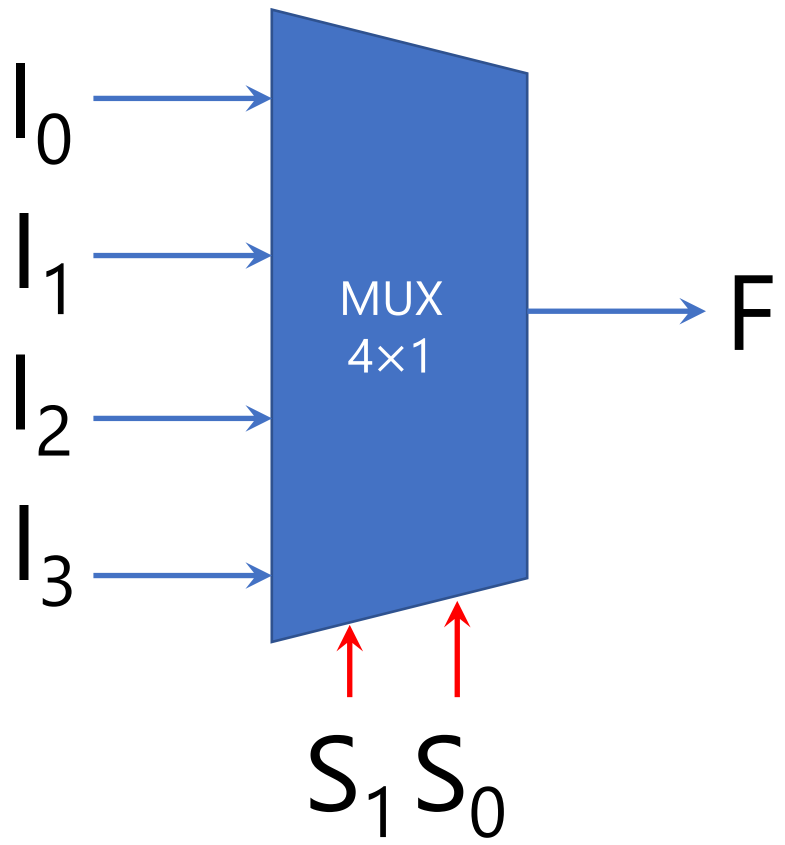
(a) Logic diagram

(b) Block diagram

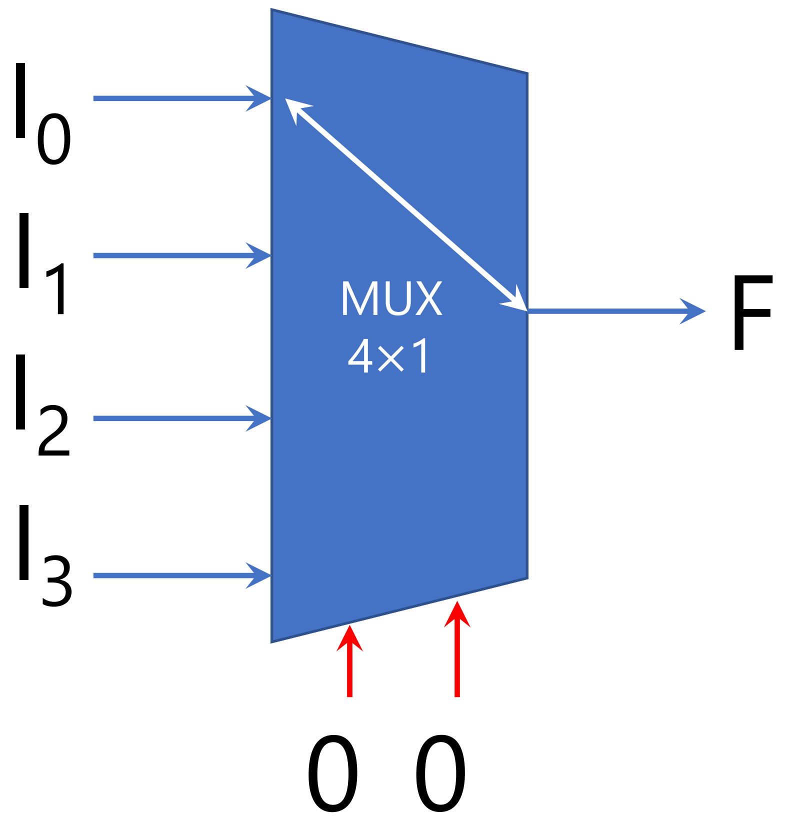
FIGURE 4.24
Two-to-one-line multiplexer

S	$F=S'I_0+SI_1$
0	I_0
1	I_1

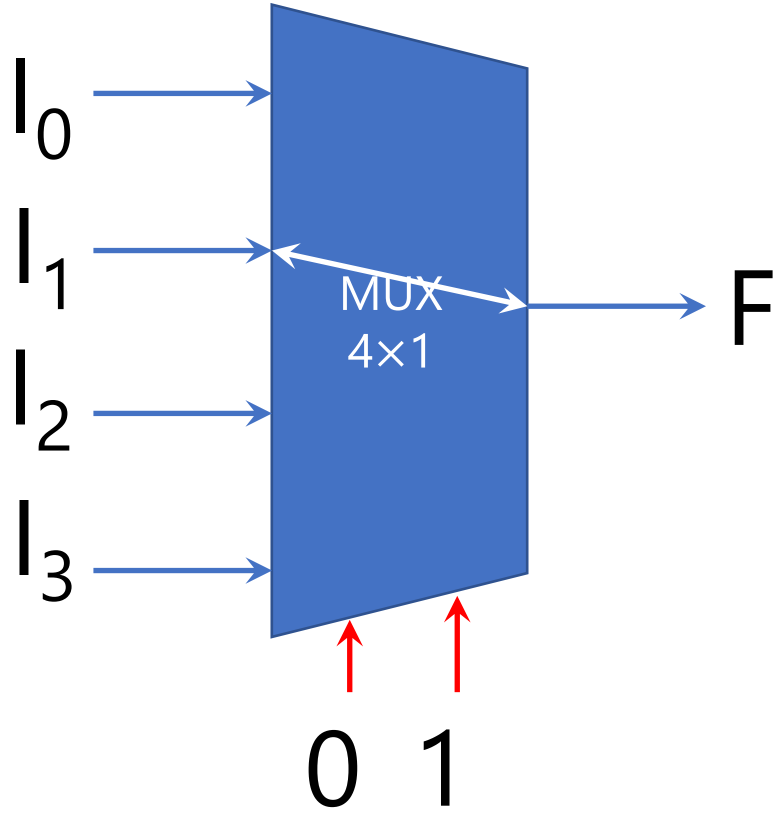
Multiplexer
$2^2 \times 1$

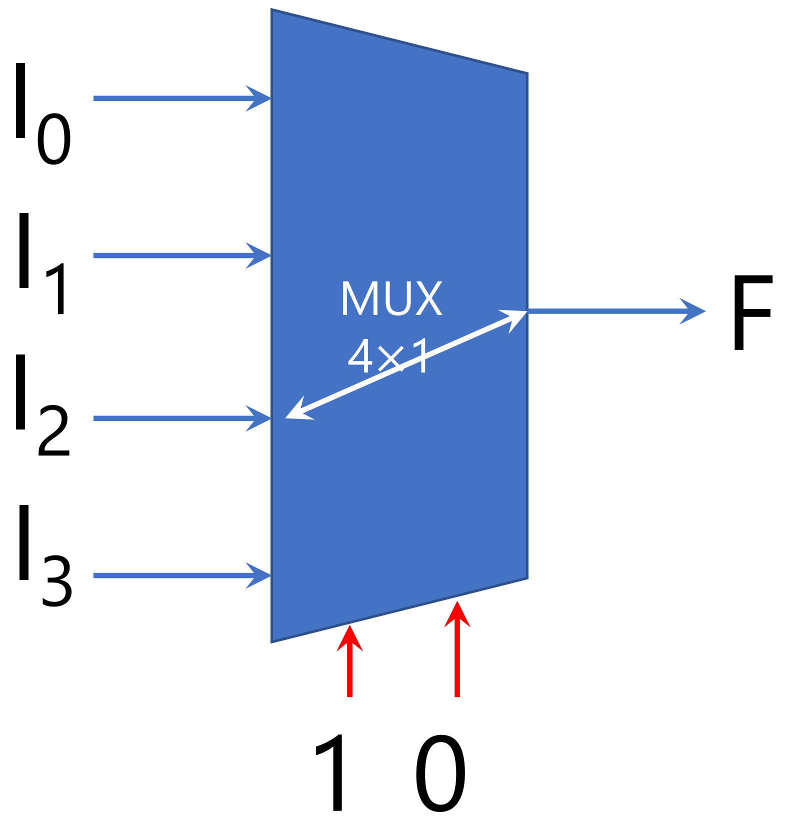


I_0
I_1
I_2
I_3
MUX
4×1
F
S_1 S_0

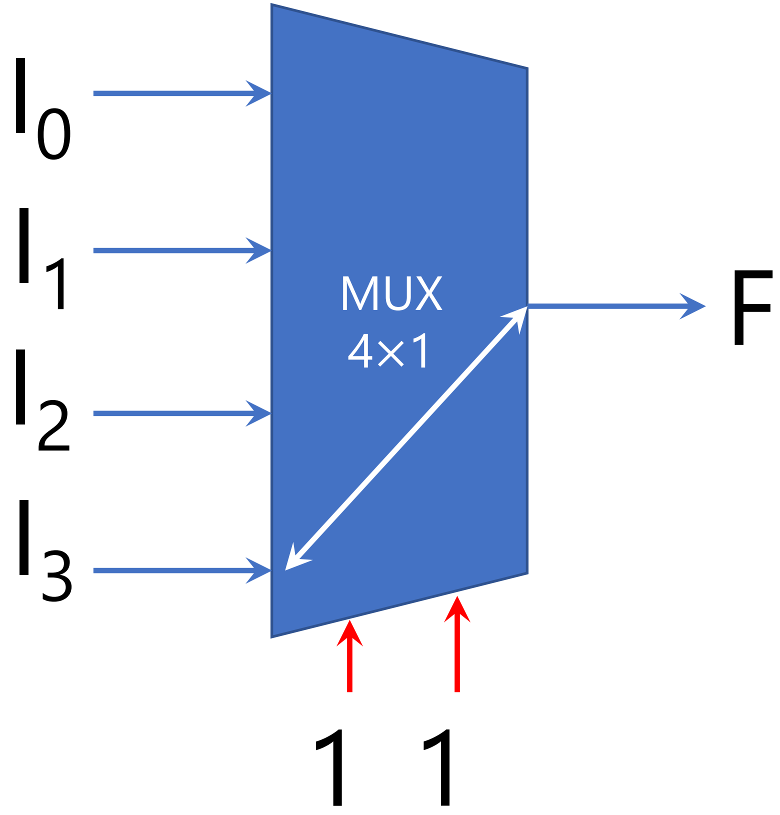


I_0
I_1
I_2
I_3
MUX
4×1
F
0 0


I0
I1
I2
I3
MUX
4×1
F
0 1

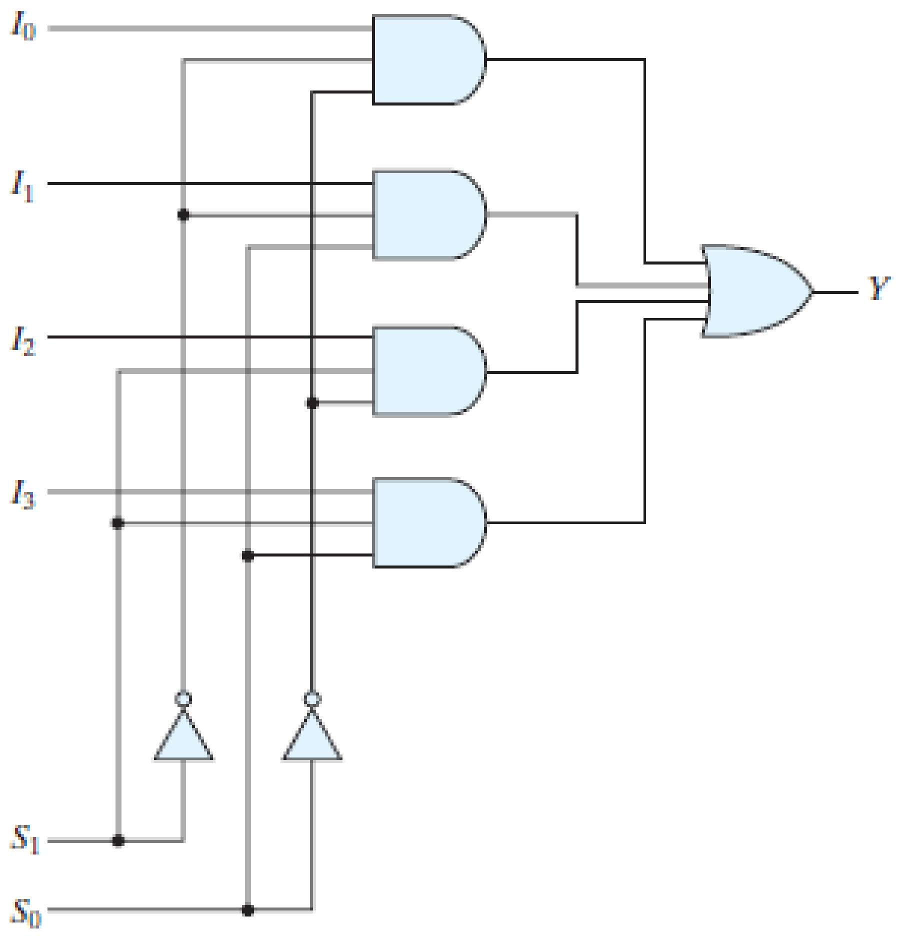


I_0
I_1
I_2
I_3
MUX
4×1
F
1 0


I_0
I_1
I_2
I_3
MUX
4×1
F
1 1

S_1	S_0	I_3	I_2	I_1	I_0	F
0	0	x	x	x	0	0
0	0	x	x	x	1	1
0	1	x	x	0	x	0
0	1	x	x	1	x	1
1	0	x	0	x	x	0
1	0	x	1	x	x	1
1	1	0	x	x	x	0
1	1	1	x	x	x	1

S_1	S_0	$F=S'_1S'_0I_0+S'_1S_0I_1+S_1S'_0I_2+S_1S_0I_3$
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



(a) Logic diagram

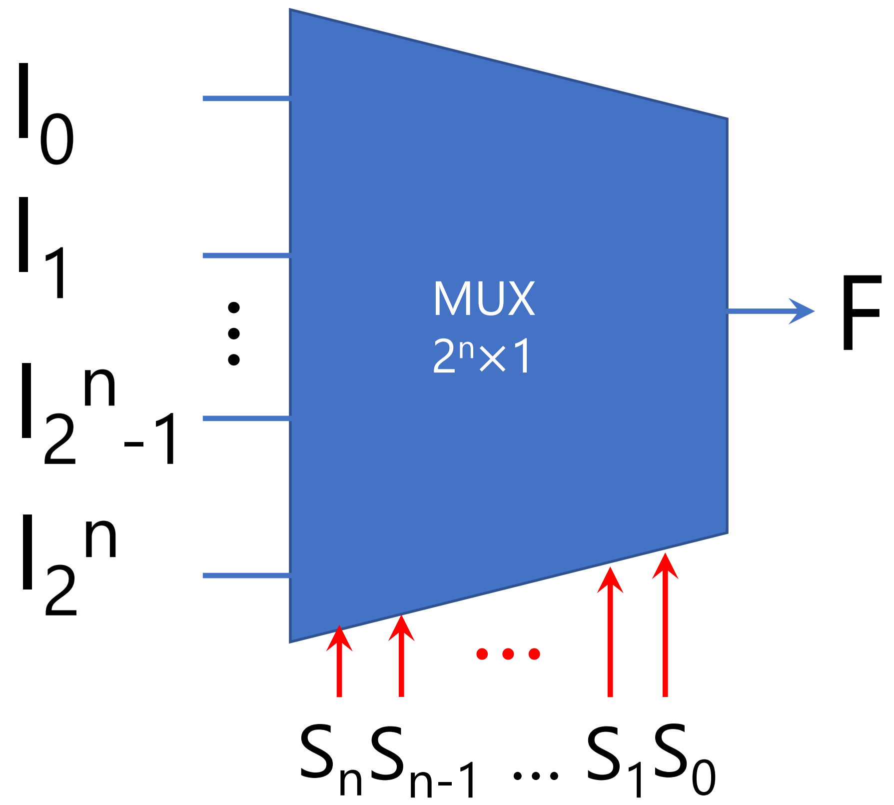
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

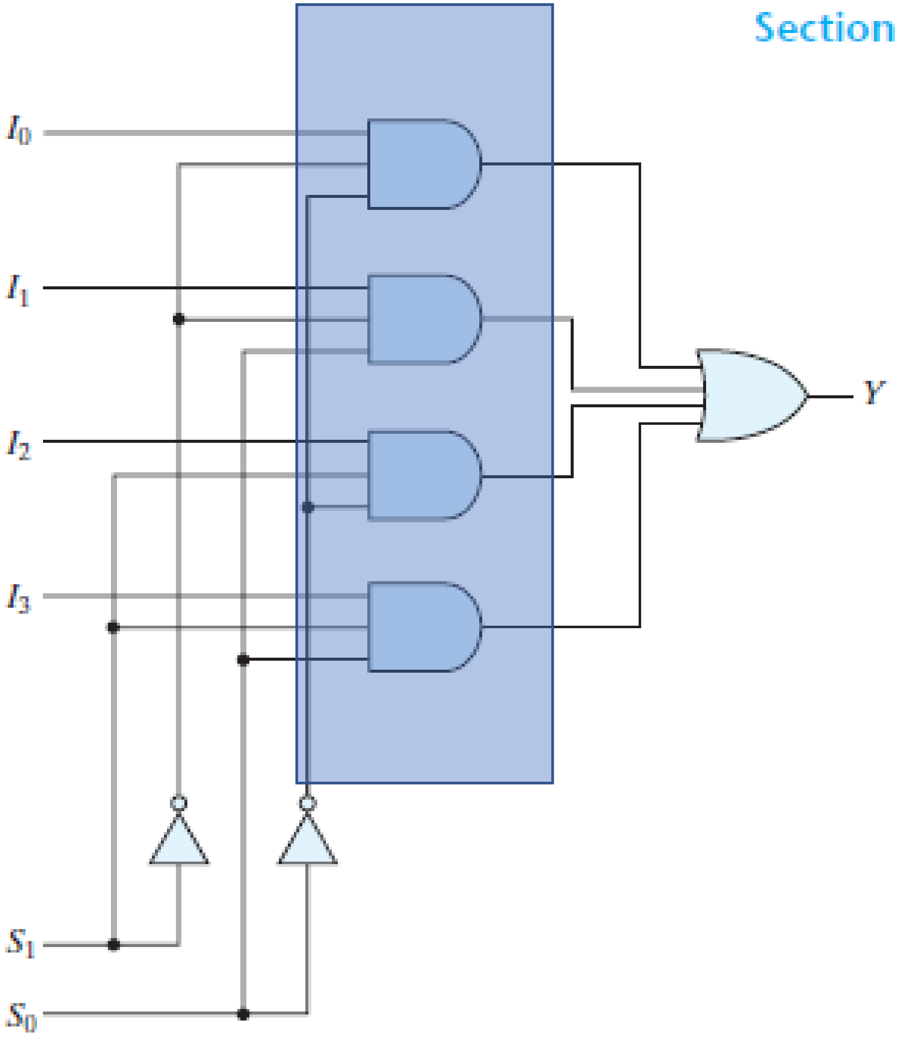
(b) Function table

FIGURE 4.25
Four-to-one-line multiplexer

Multiplexer

$2^n \times 1$


I_0
I_1
I_{2^n-1}
I_{2^n}
MUX
$2^n \times 1$
F
$S_n S_{n-1} \dots S_1 S_0$



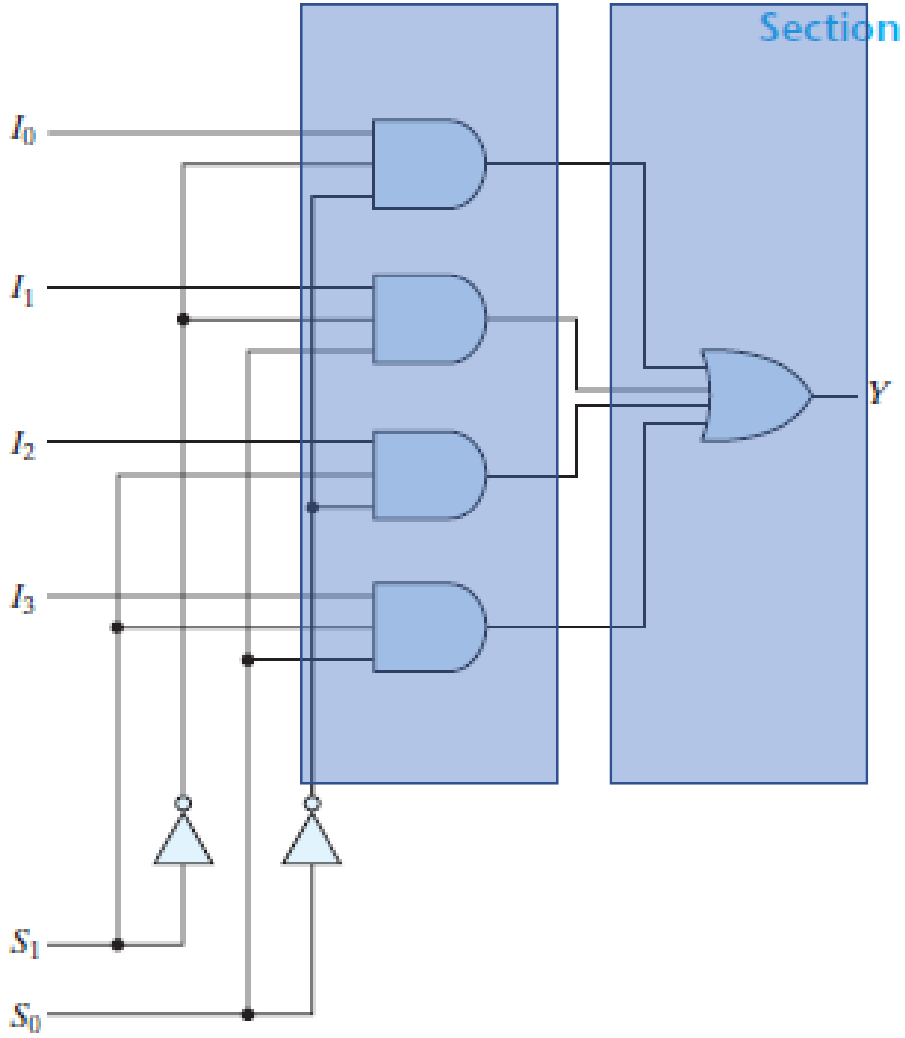
(a) Logic diagram

≈ Decoder + OR

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(b) Function table

FIGURE 4.25
Four-to-one-line multiplexer

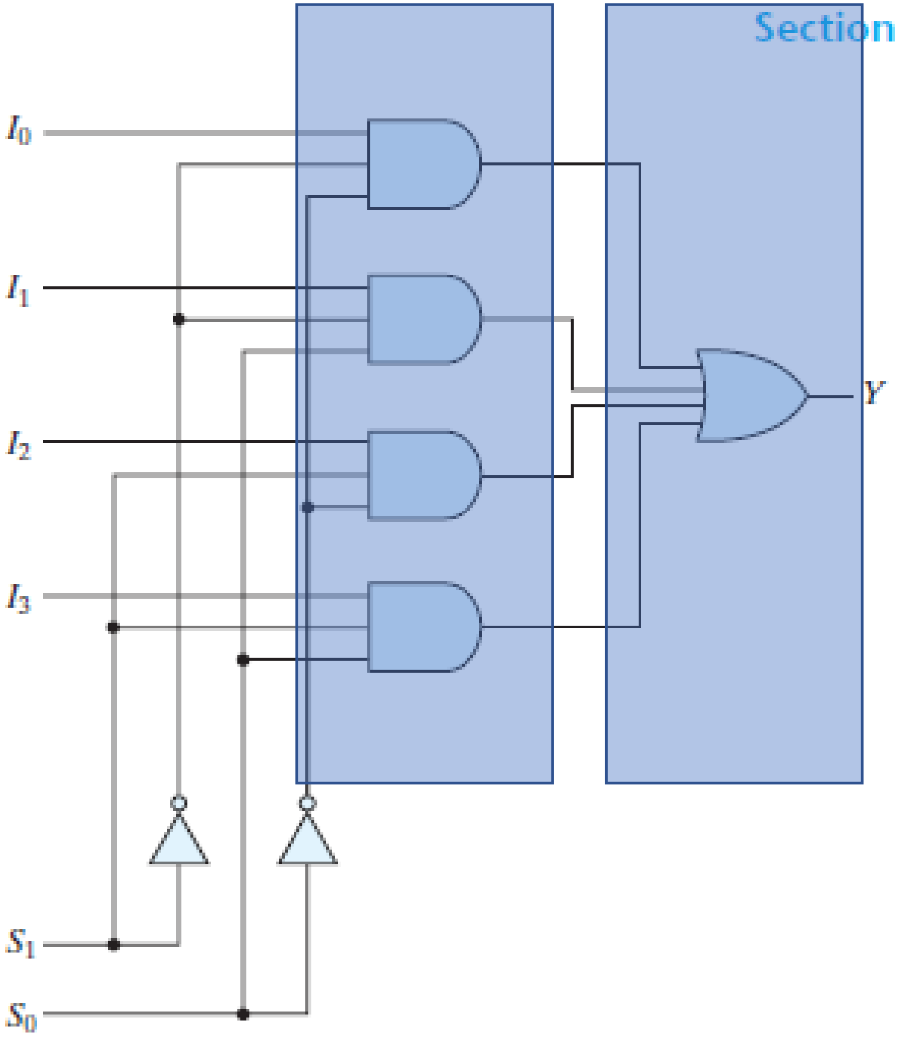


(a) Logic diagram

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(b) Function table

FIGURE 4.25
Four-to-one-line multiplexer



(a) Logic diagram

Sum of Products
2 Levels
ANDs-OR

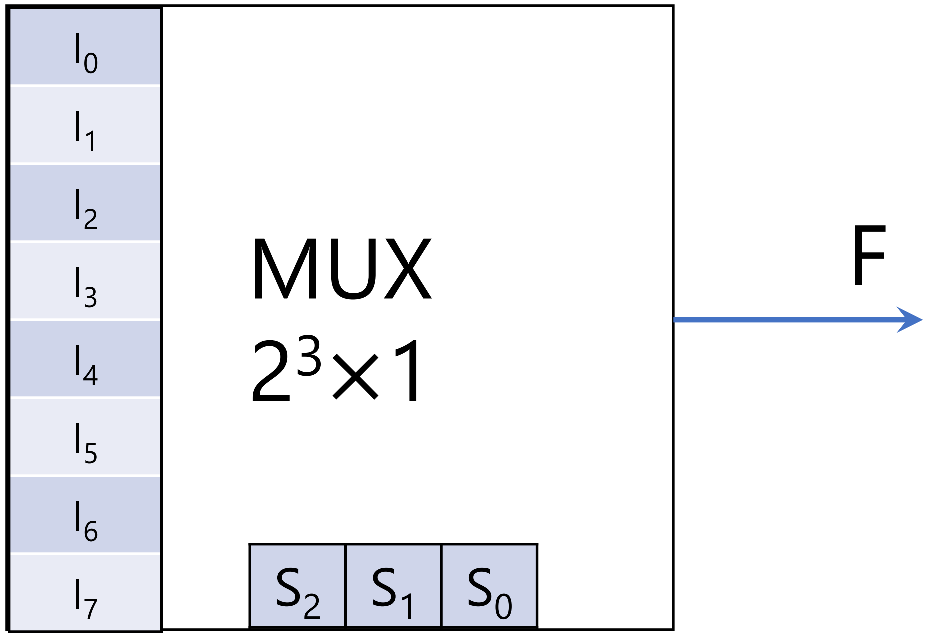
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(b) Function table

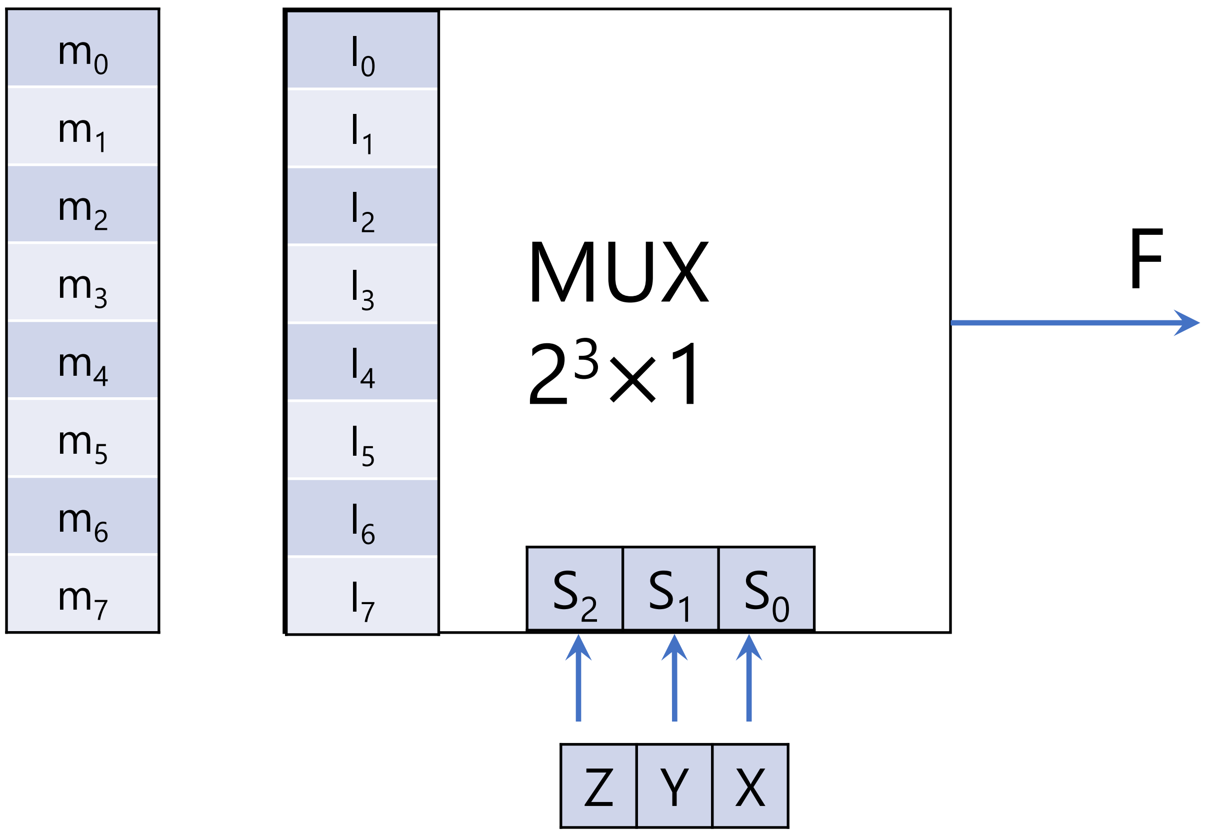
FIGURE 4.25
Four-to-one-line multiplexer

Multiplexer
Boolean Function

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



I_0
I_1
I_2
I_3
I_4
I_5
I_6
I_7
MUX
$2^3 \times 1$
S_2
S_1
S_0
F

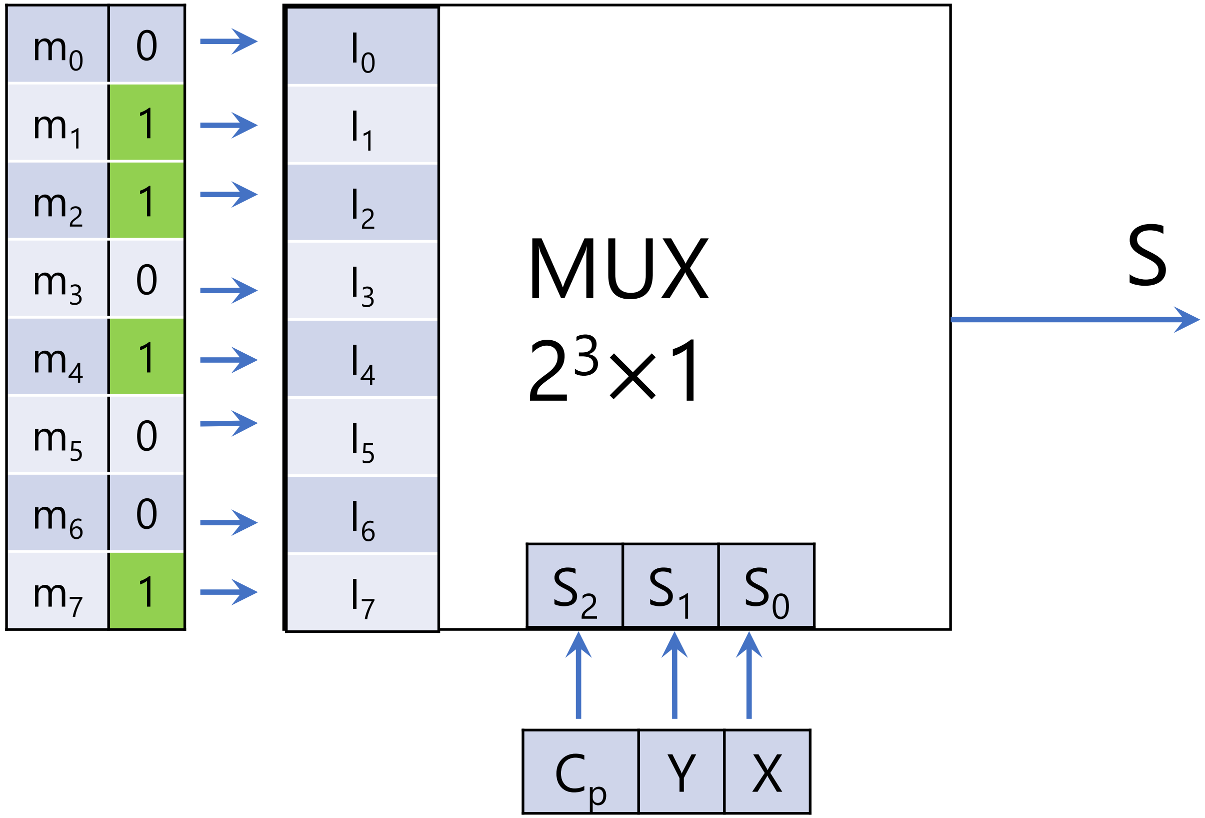


m_0
m_1
m_2
m_3
m_4
m_5
m_6
m_7
I_0
I_1
I_2
I_3
I_4
I_5
I_6
I_7
MUX
$2^3 \times 1$
F
S_2
S_1
S_0
Z
Y
X

MUX
Full Adder

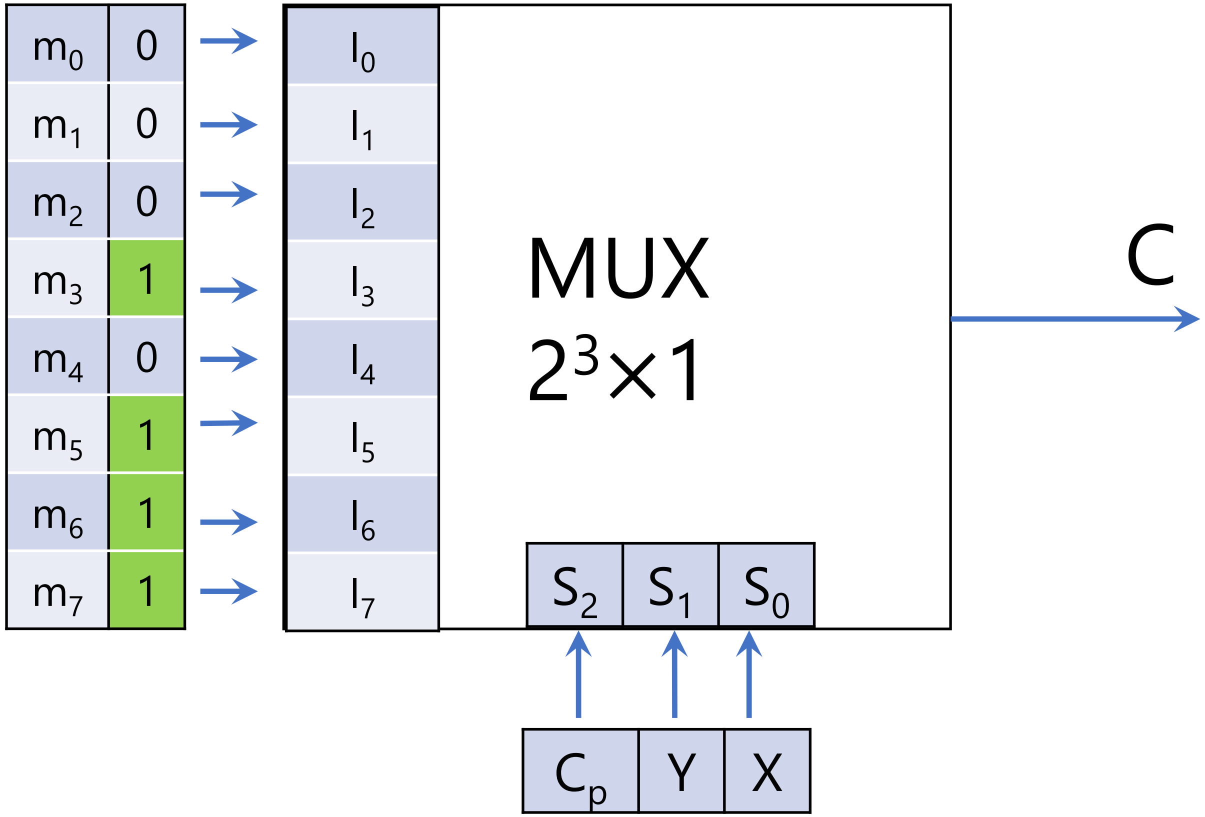
$$S = \sum m(1,2,4,7)$$
$$C = \sum m(3,5,6,7)$$

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



m_0 0
m_1 1
m_2 1
m_3 0
m_4 1
m_5 0
m_6 0
m_7 1
I_0
I_1
I_2
I_3
I_4
I_5
I_6
I_7
MUX
$2^3 \times 1$
S
S_2 S_1 S_0
C_p Y X

$$S = \sum m(1,2,4,7)$$

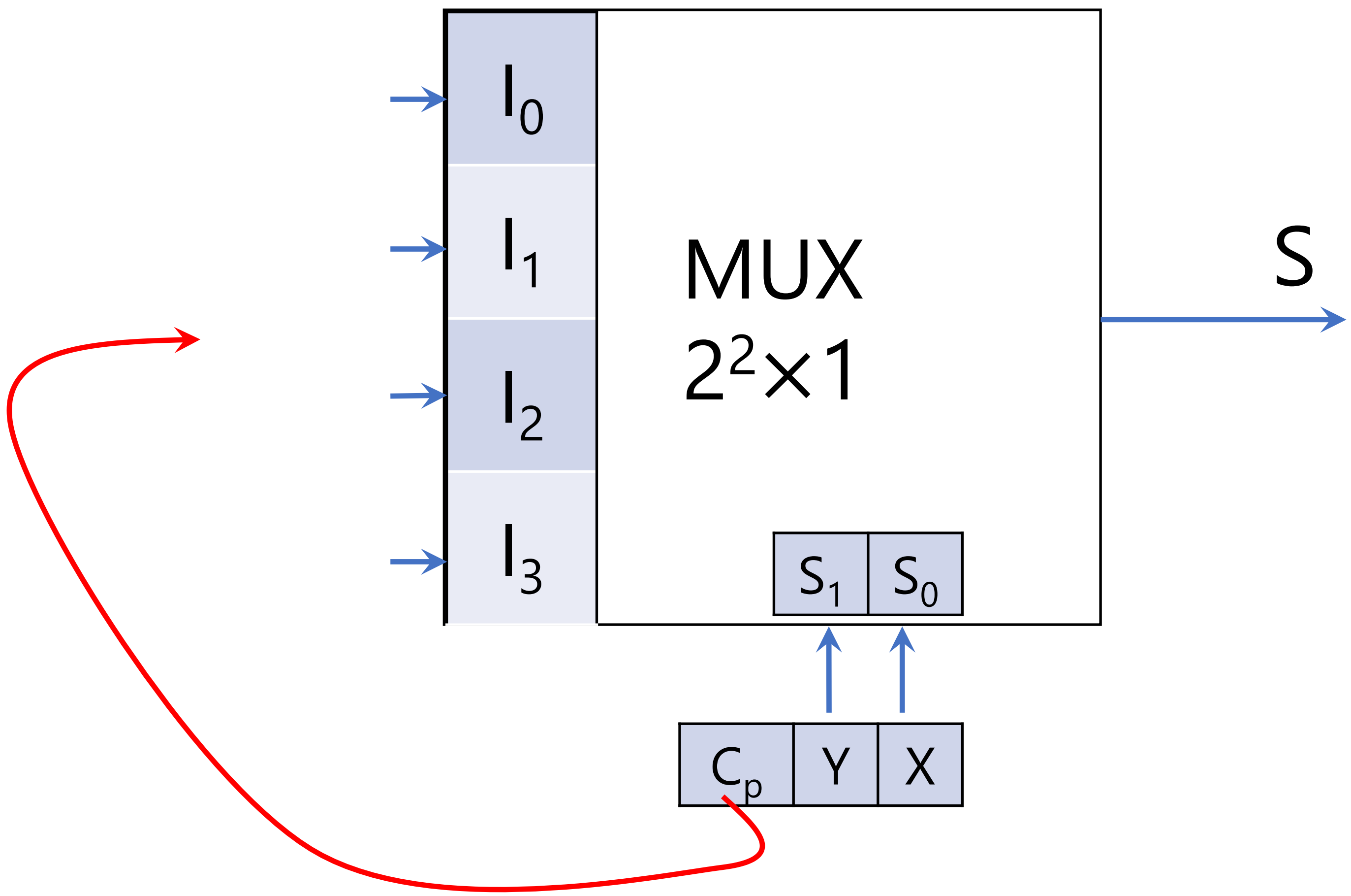


m0 0
m1 0
m2 0
m3 1
m4 0
m5 1
m6 1
m7 1
I0
I1
I2
I3
I4
I5
I6
I7
MUX
2³×1
C
S2 S1 S0
Cp Y X

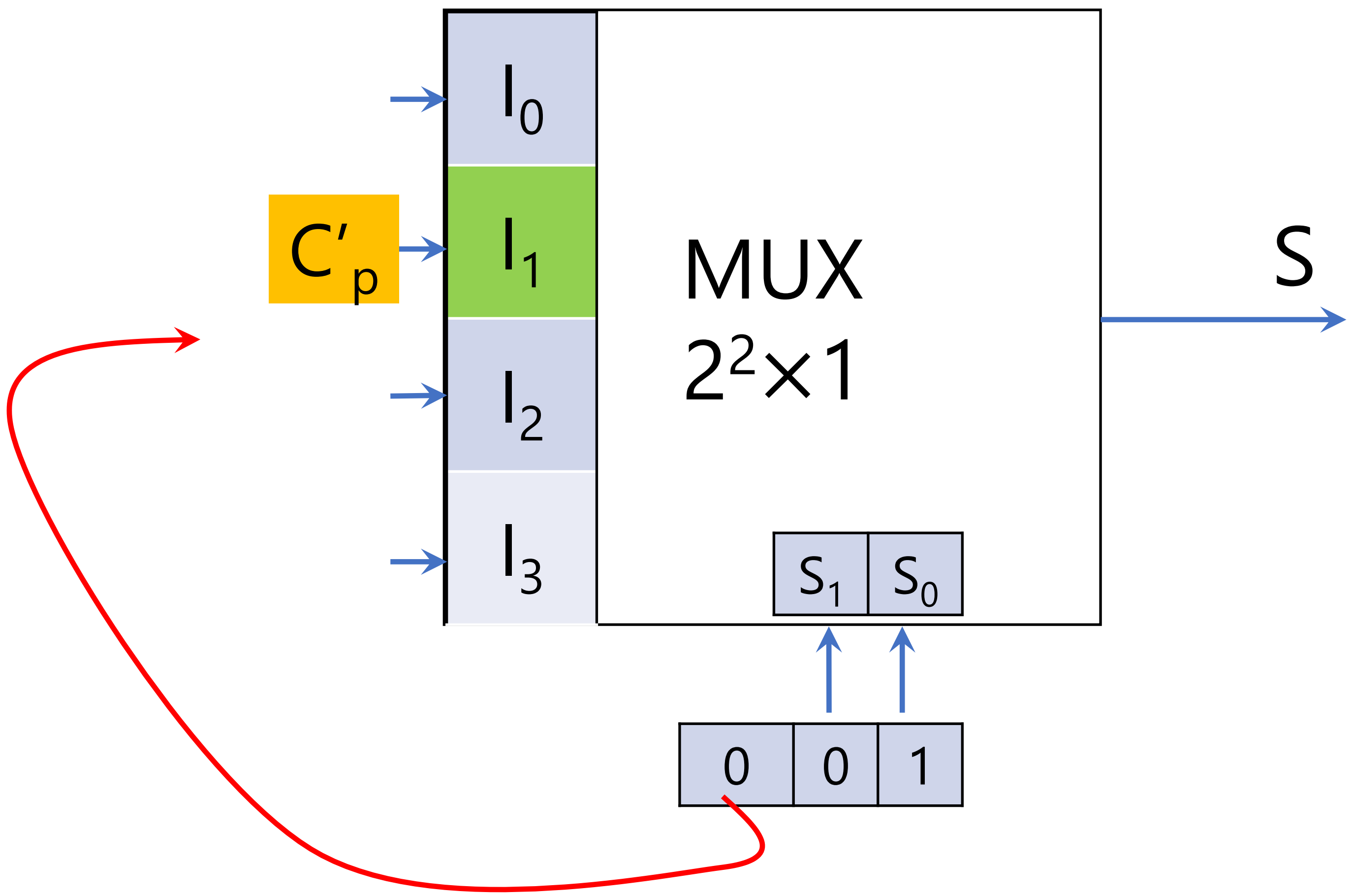
$$C = \sum m(3,5,6,7)$$

Multiplexer
Boolean Function II

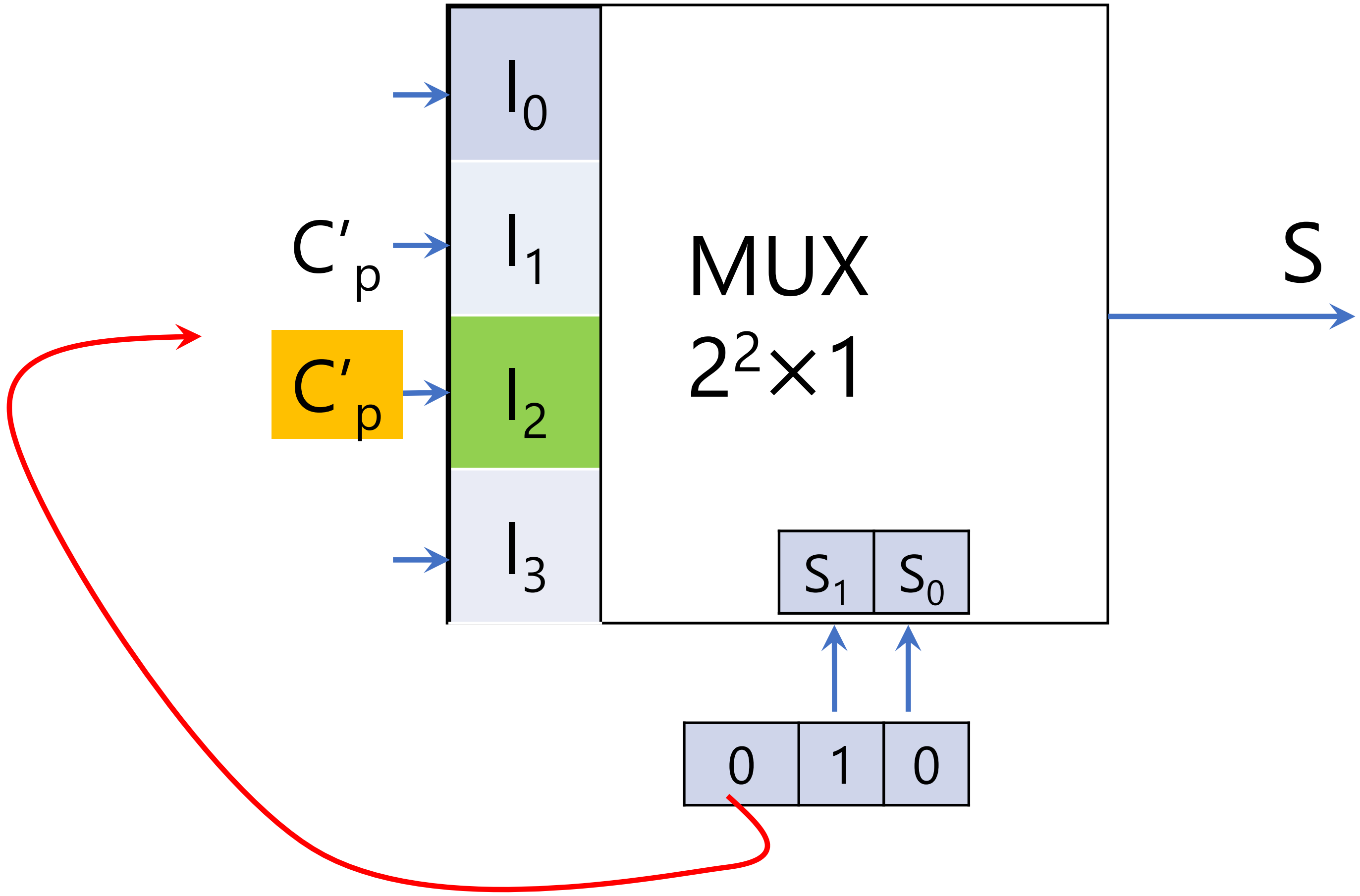
Book: Page 161



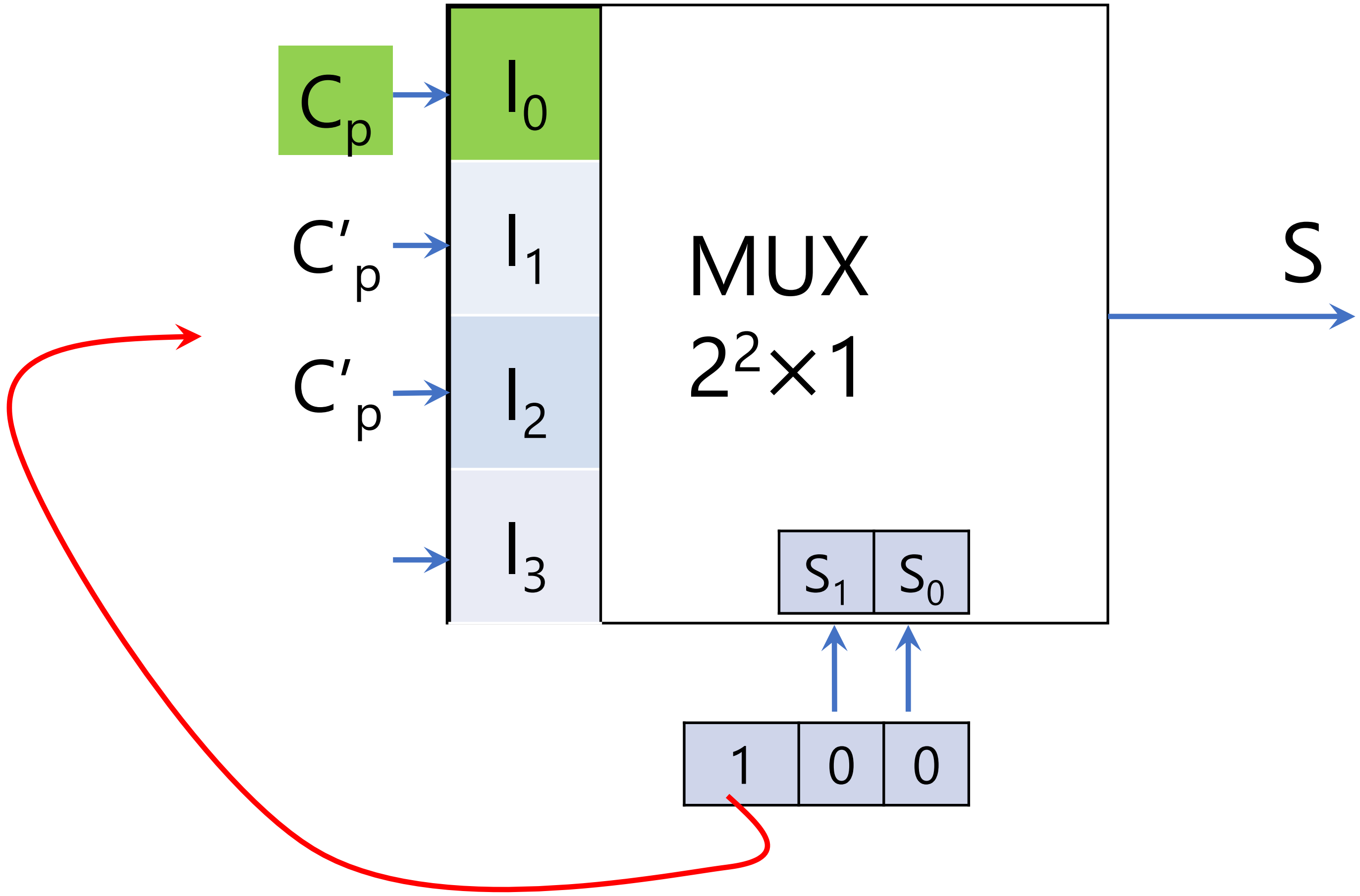
$$S = \sum m(1,2,4,7)$$



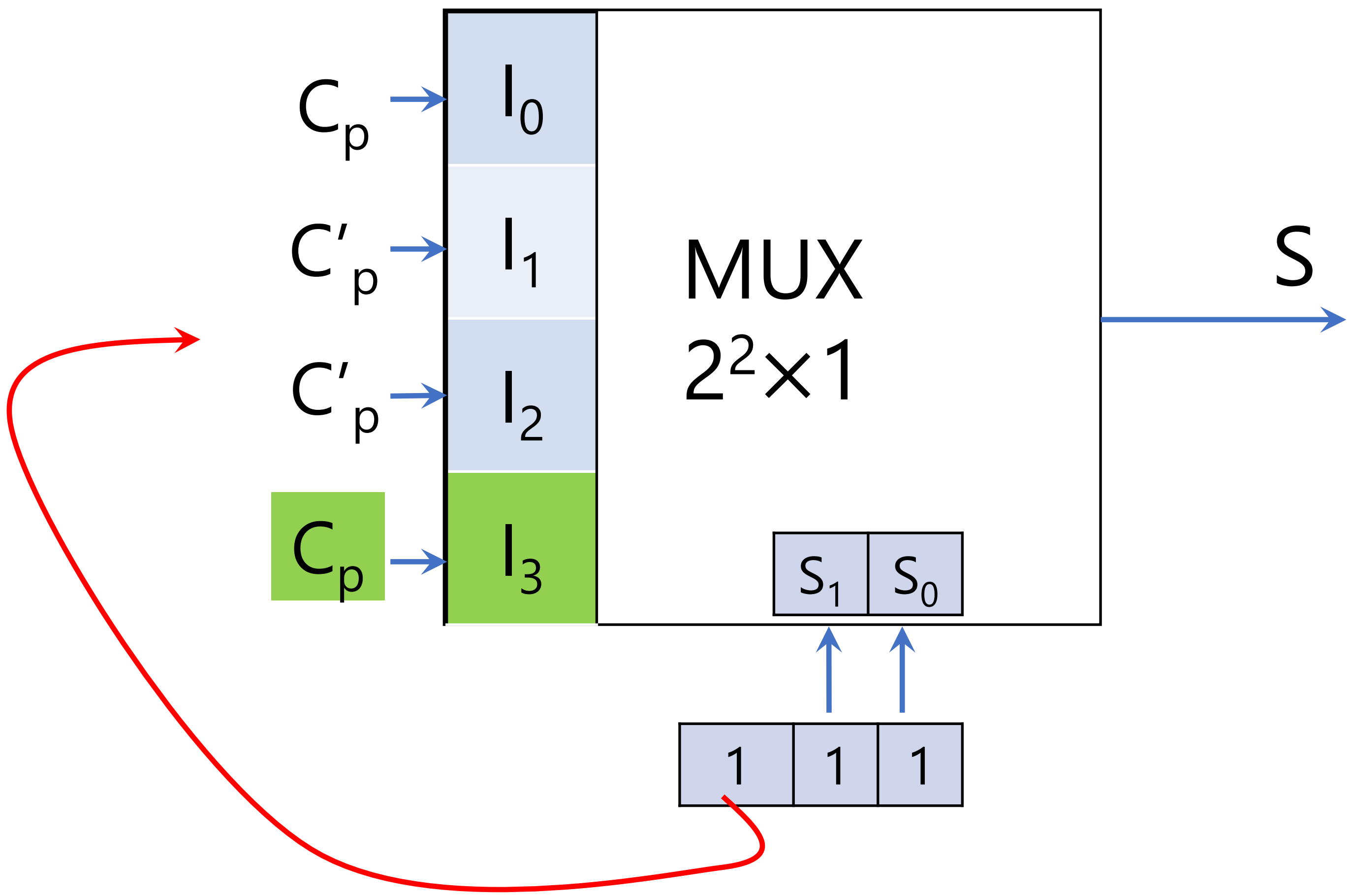
$$S = \sum m(1,2,4,7)$$



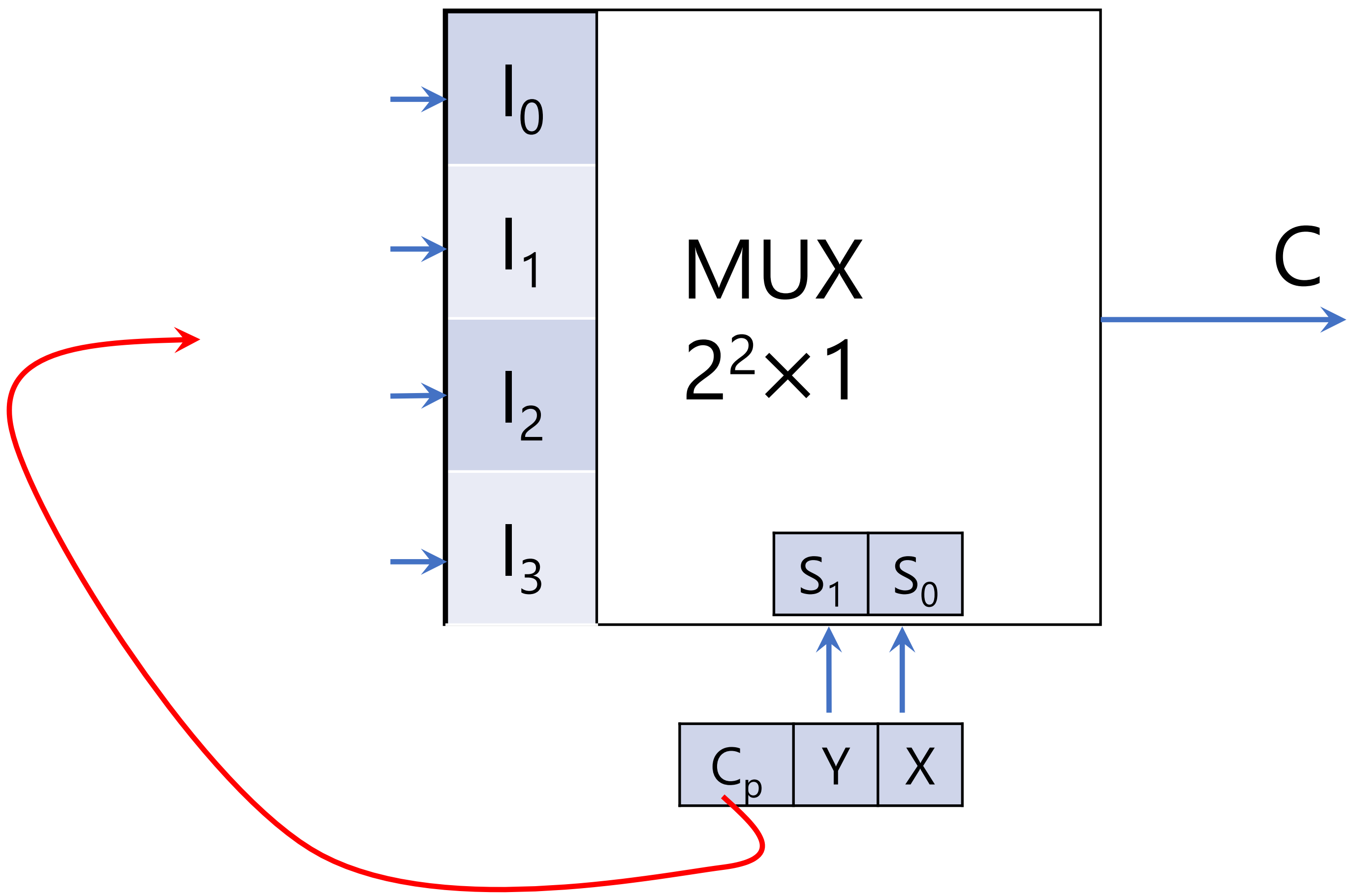
$$S = \sum m(1,2,4,7)$$



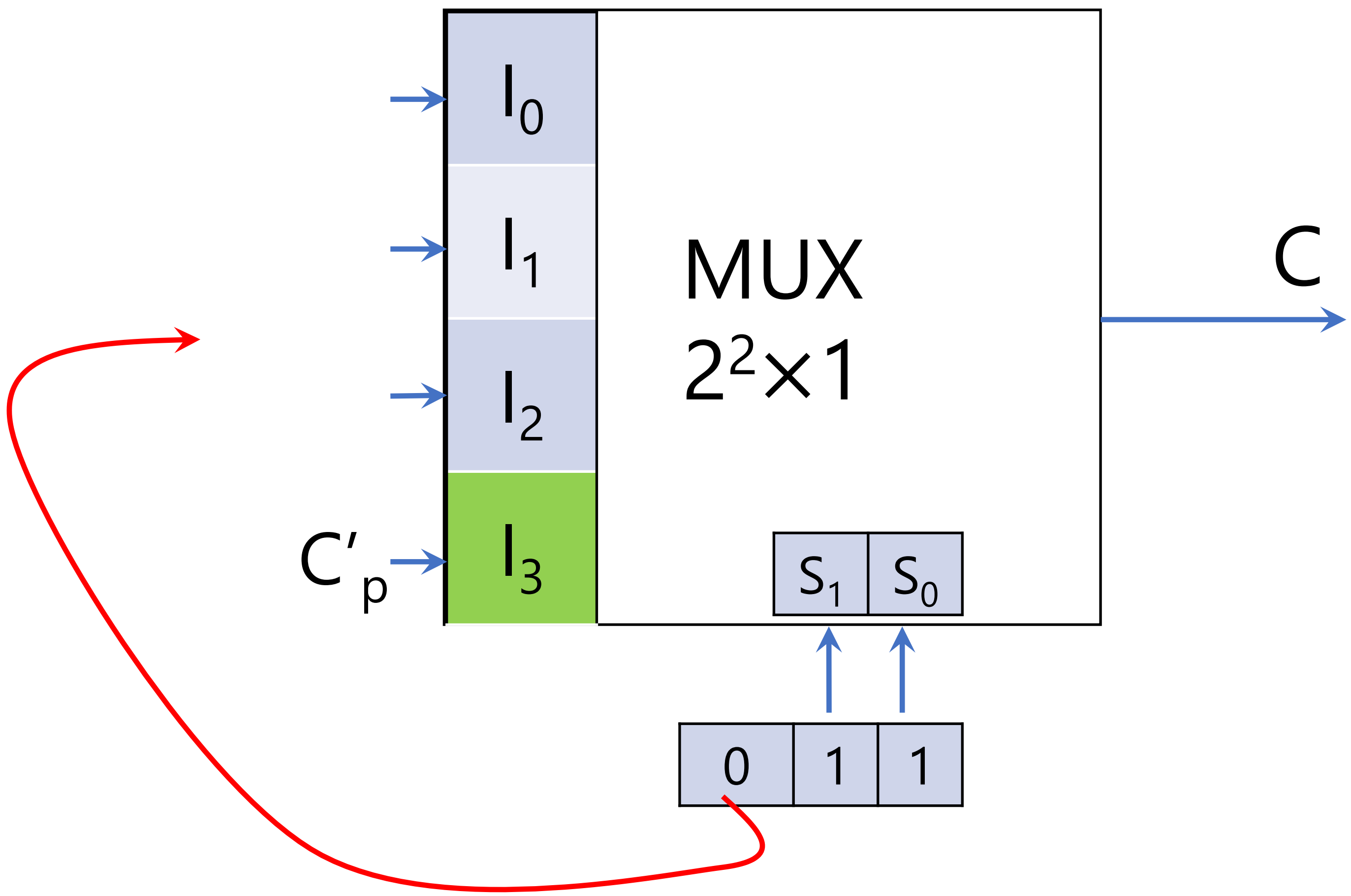
$$S = \sum m(1,2,4,7)$$



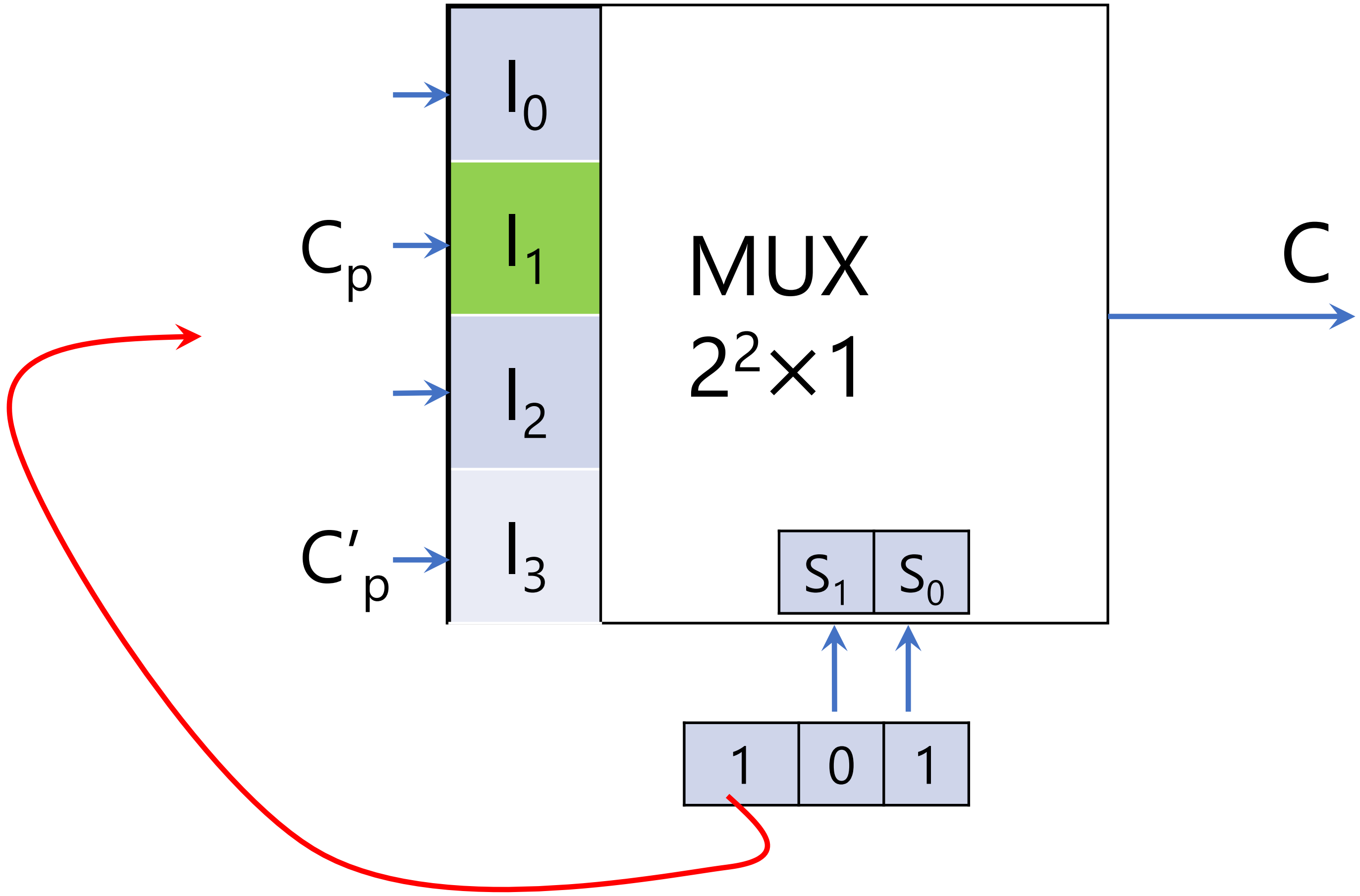
$$S = \sum m(1,2,4,7)$$



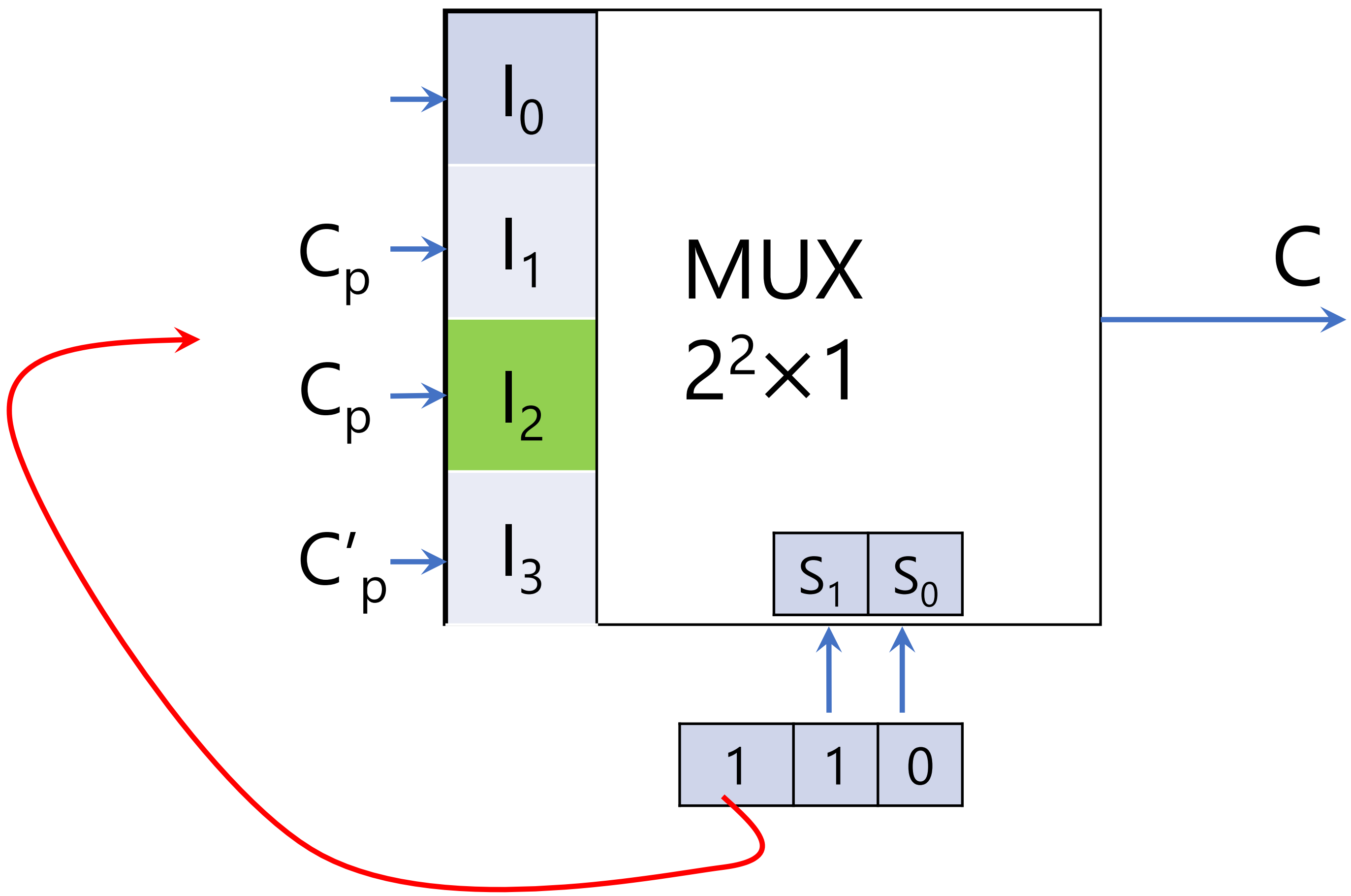
$$C = \sum m(3,5,6,7)$$



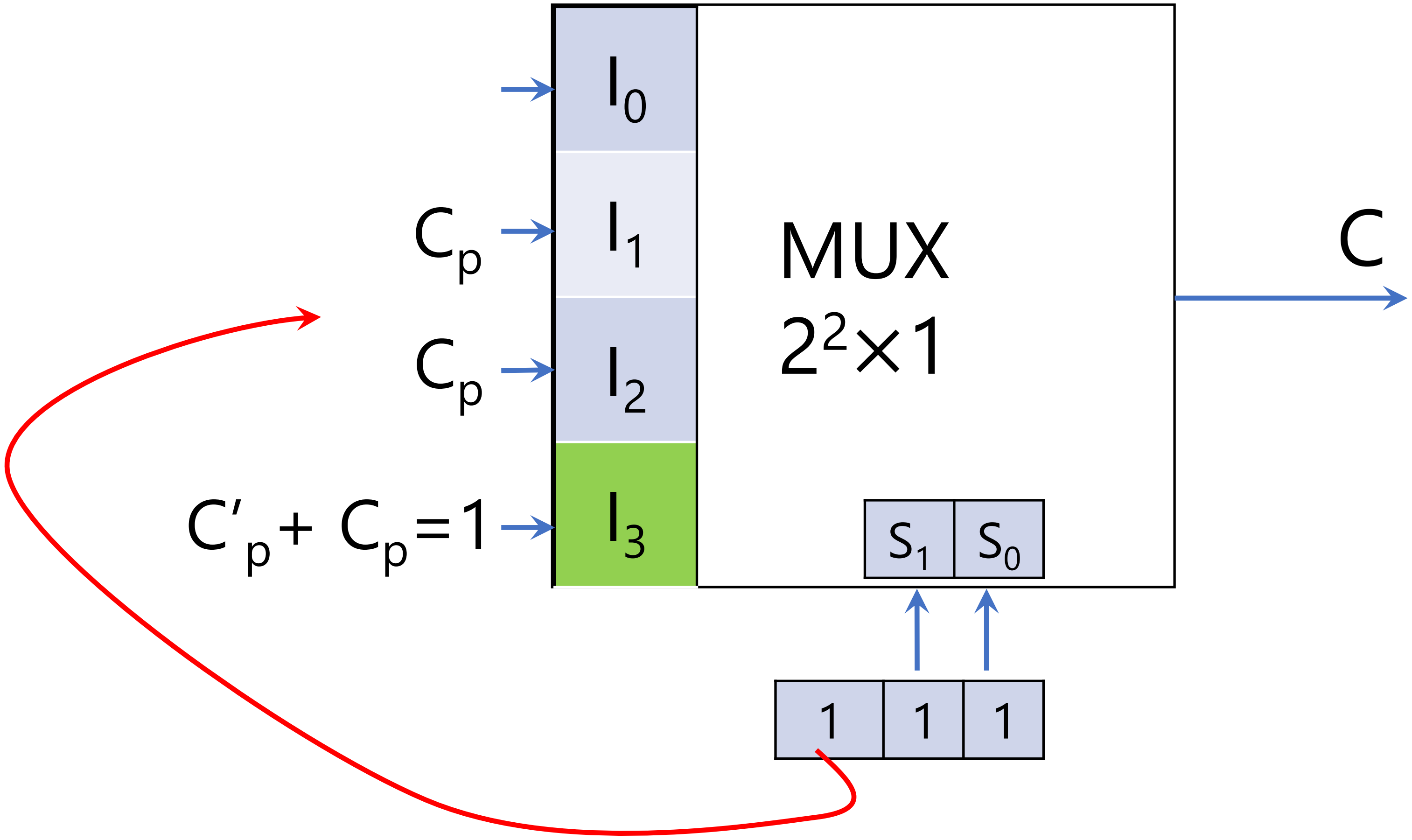
$$C = \sum m(3,5,6,7)$$



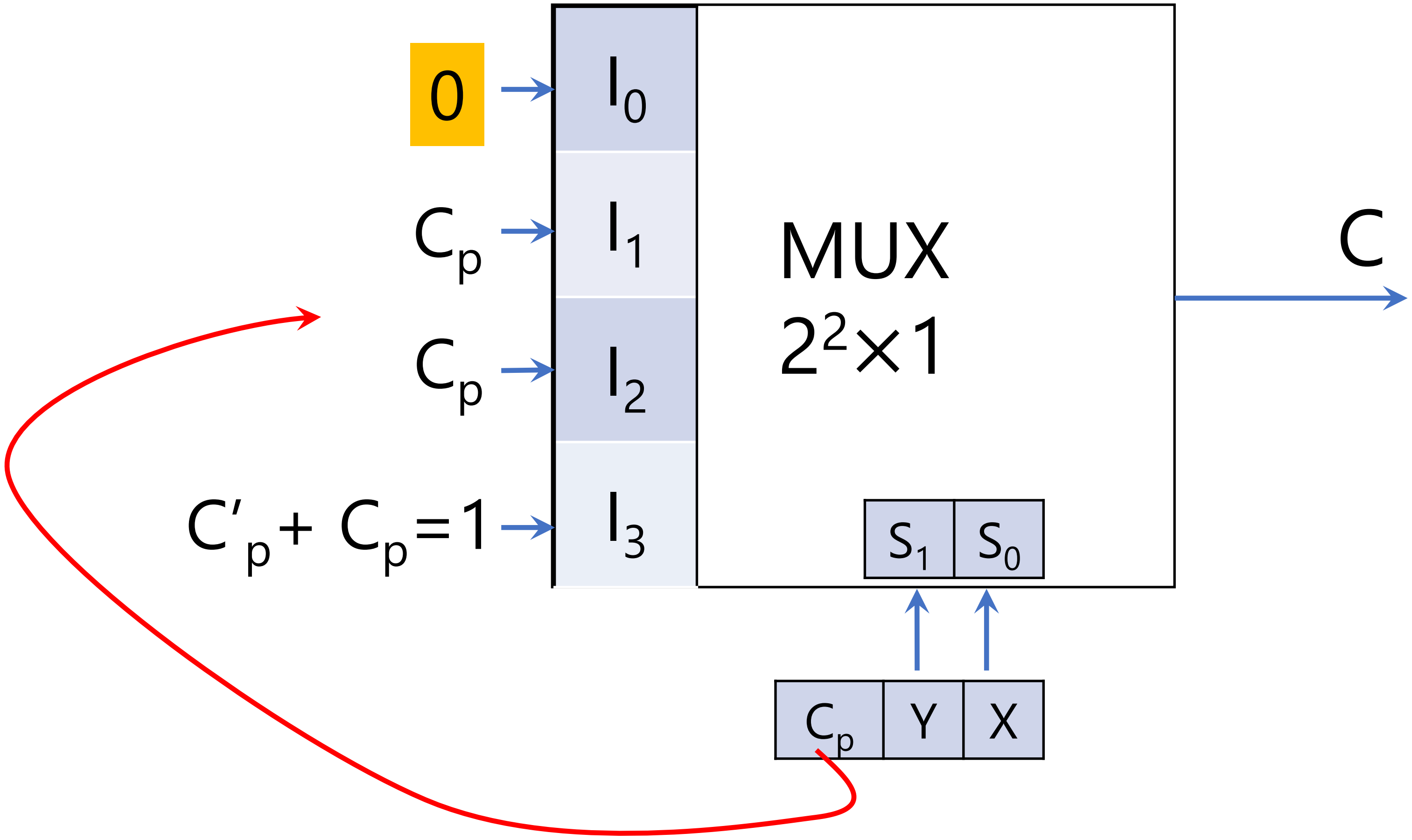
$$C = \sum m(3,5,6,7)$$



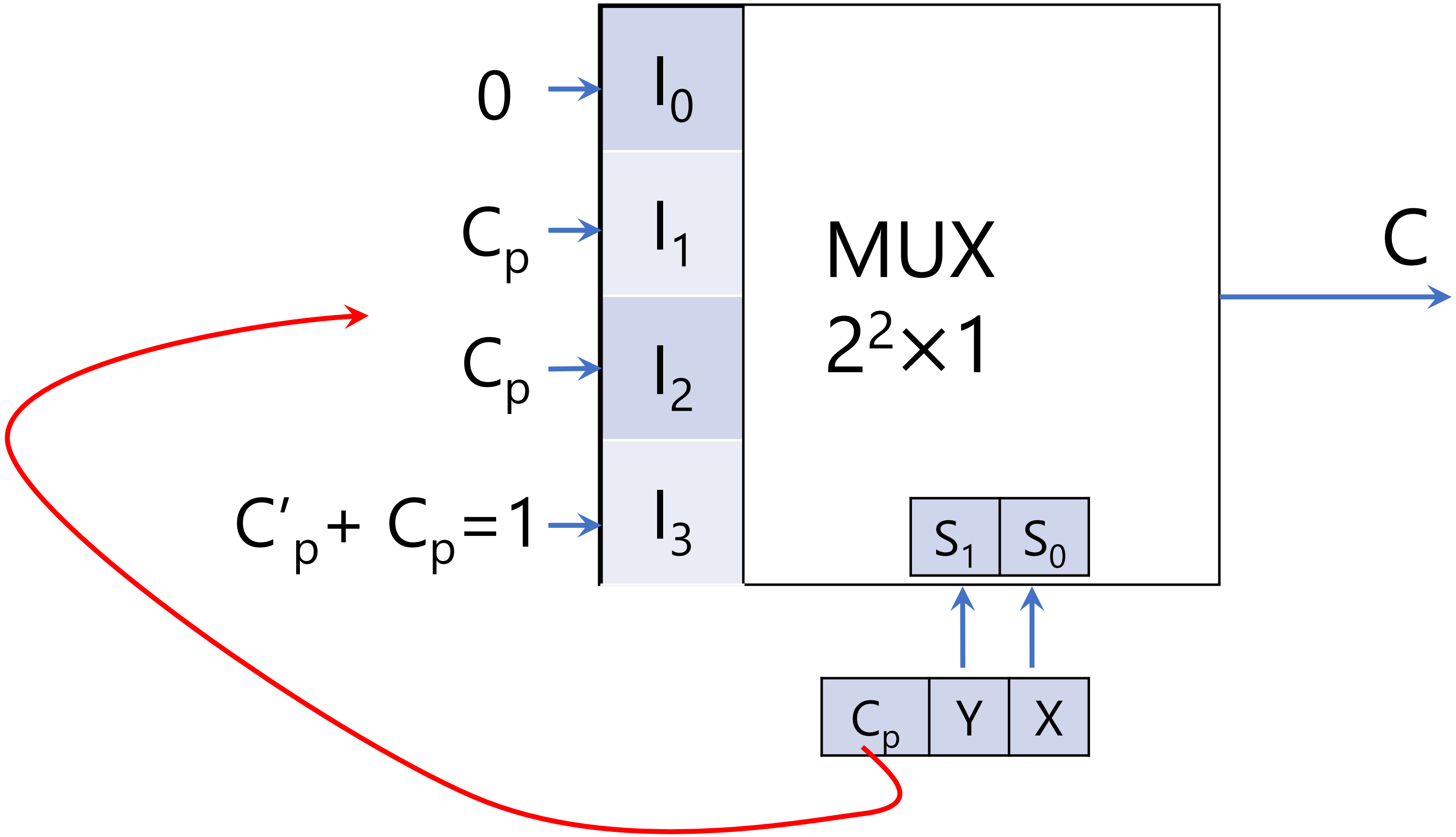
$$C = \sum m(3,5,6,7)$$



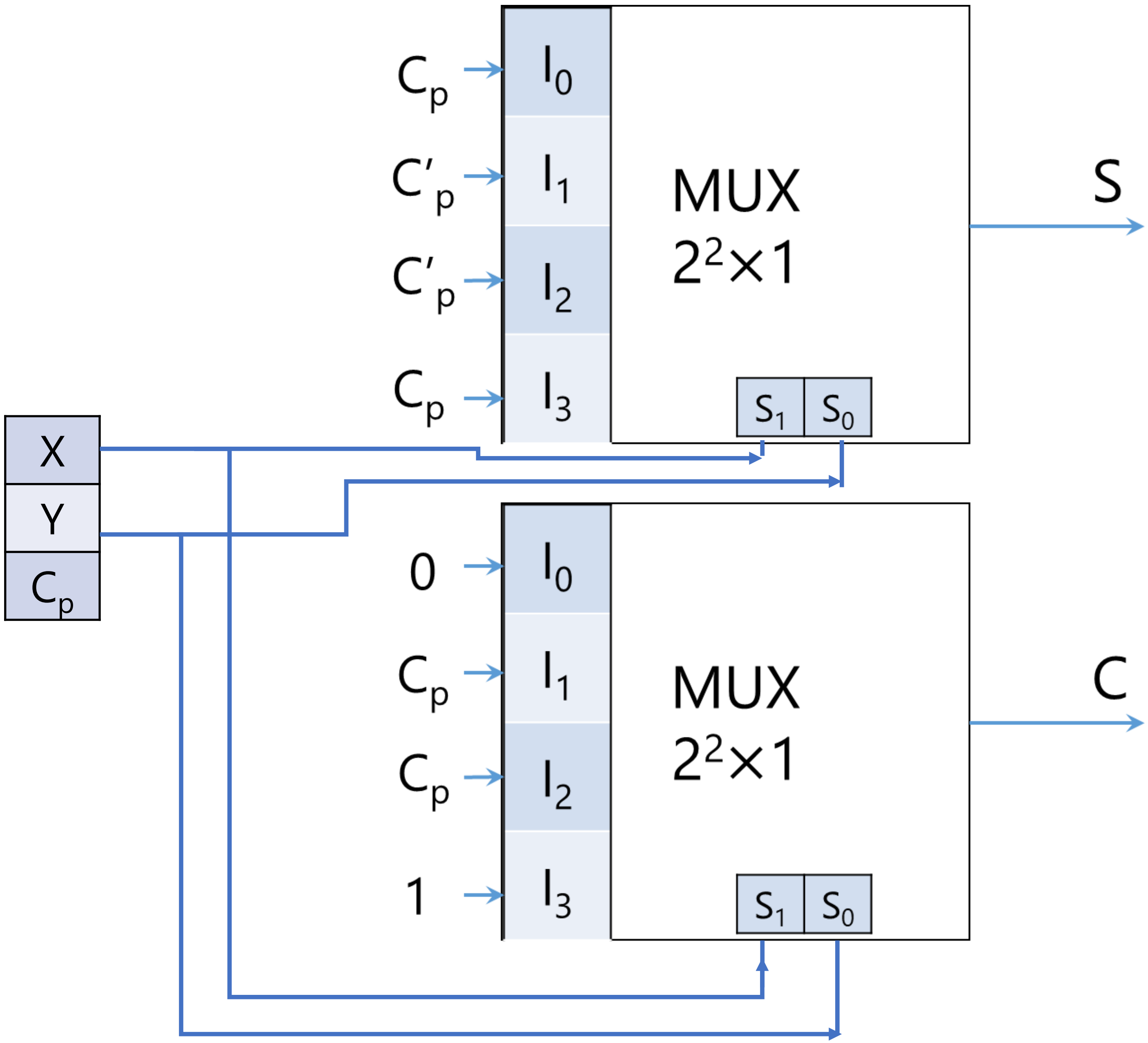
$$C = \sum m(3,5,6,7)$$



$$C = \sum m(3,5,6,7)$$



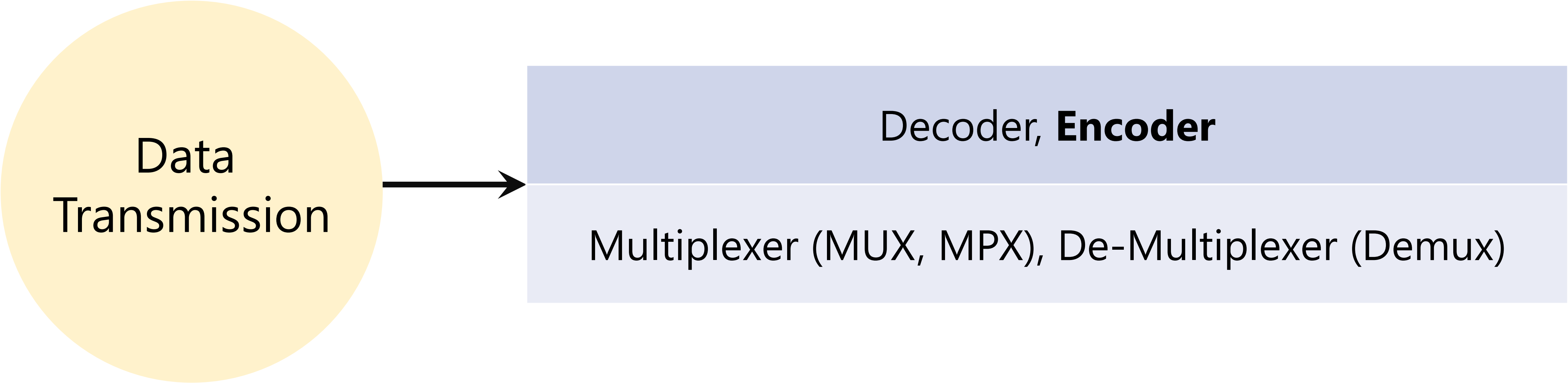
$$C = \sum m(3,5,6,7)$$



Cp
C'p
C'p
Cp
I0
I1
I2
I3
MUX
2²×1
S1
S0
S
X
Y
Cp
0
Cp
Cp
1
I0
I1
I2
I3
MUX
2²×1
S1
S0
C

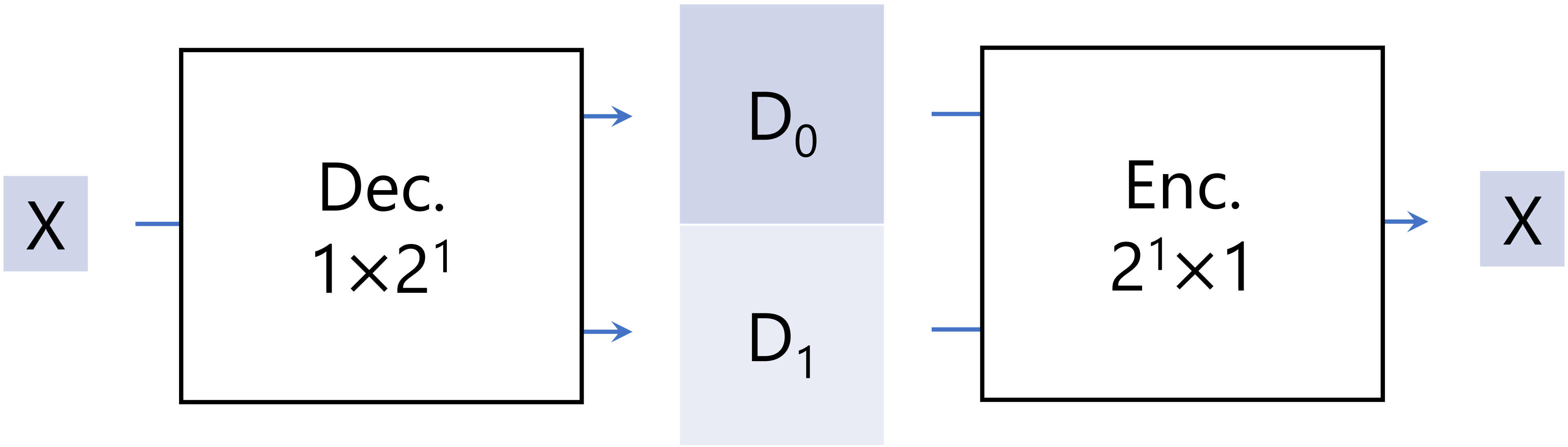
Multiplexer
Three-State Gates + Decoders

Book: Page 162-164

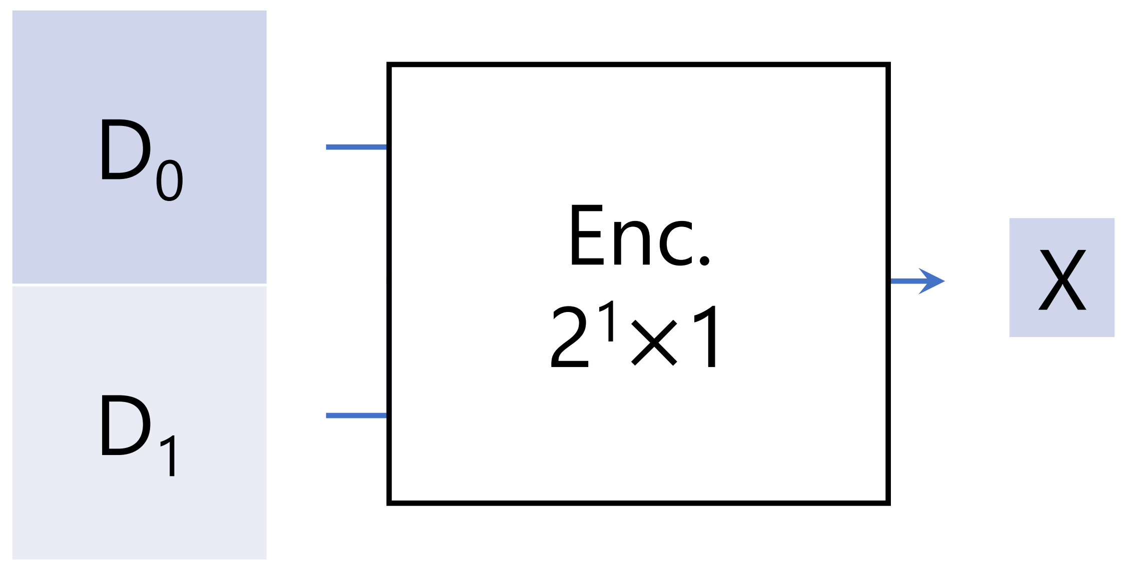

Data Transmission
Decoder, **Encoder**
Multiplexer (MUX, MPX), De-Multiplexer (Demux)

Encoder

Encoder
1-hot to Binary



X
Dec.
1×2^1
D0
D1
Enc.
2^1×1
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

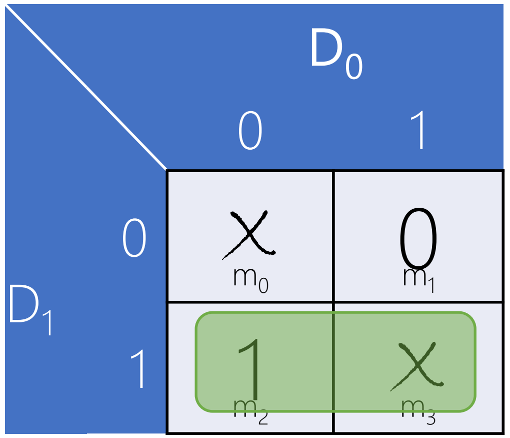
D_1 \ D_0	0	1
0	X (m_0)	0 (m_1)
1	1 (m_2)	X (m_3)

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

$D_1 \backslash D_0$	0	1
0	X m_0	0 m_1
1	1 m_2	X m_3

$$F_1 = D'_0$$

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



$$F_1 = D_1$$

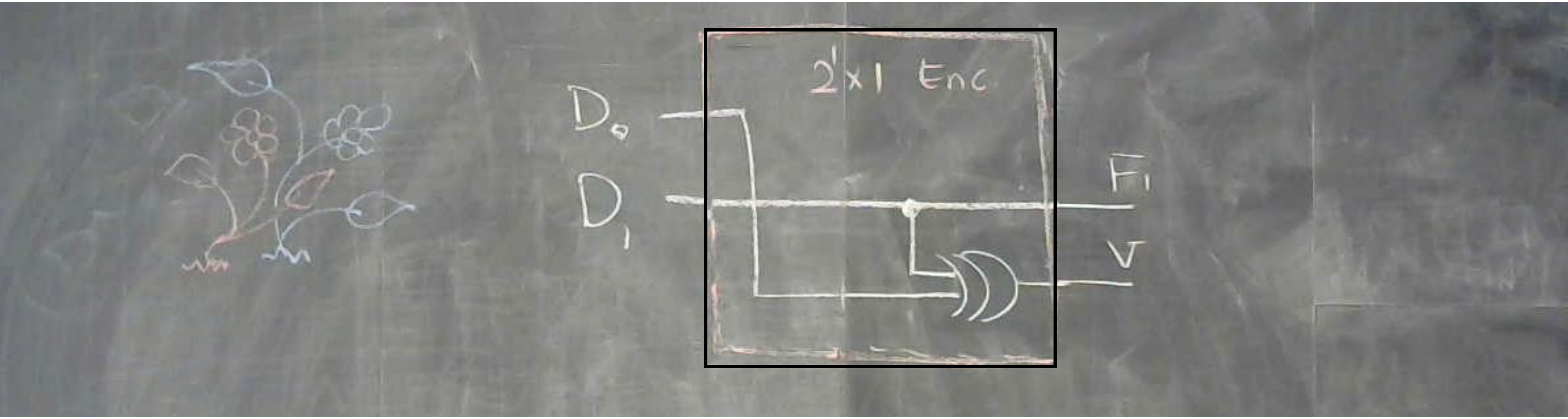
D_1	D_0	F_1	V
0	0	X	0
0	1	0	1
1	0	1	1
1	1	X	0

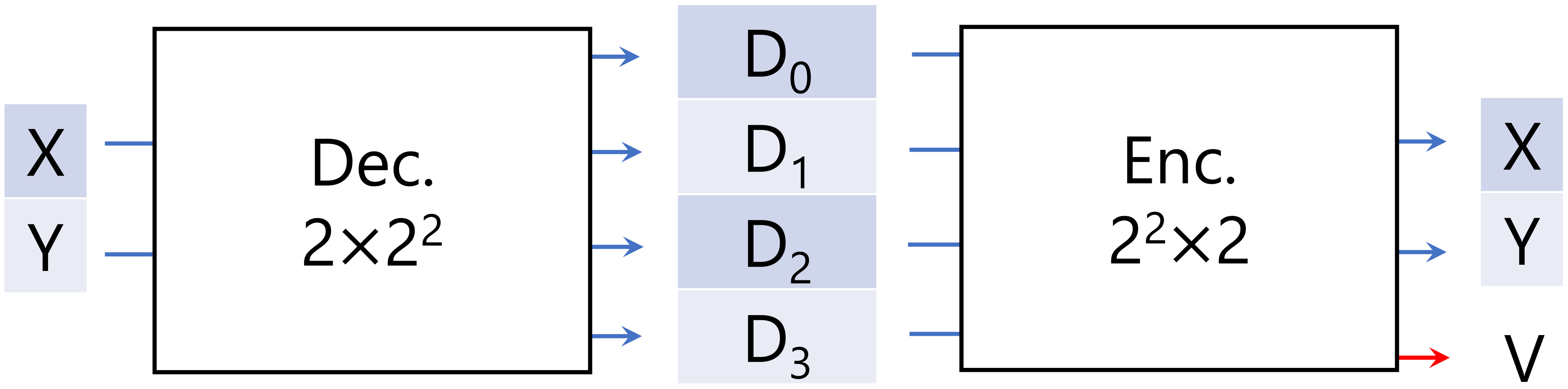
D_1 \ D_0	0	1
0	X (m_0)	0 (m_1)
1	1 (m_2)	X (m_3)

$$F_1 = D_1$$

D_1 \ D_0	0	1
0	0 (m_0)	1 (m_1)
1	1 (m_2)	0 (m_3)

$$V = D_0D'_1 + D'_0D_1$$
$$= D_0 \oplus D_1$$


2^1x1 Enc
D_0
D_1
F_1
V



X
Y
Dec.
2×2^2
D_0
D_1
D_2
D_3
Enc.
$2^2\times2$
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_3D_2 \ D_1D_0	00	01	11	10
00	X m_0	0 m_1	X m_3	1 m_2
01	0 m_4	X m_5	X m_7	X m_6
11	X m_{12}	X m_{13}	X m_{15}	X m_{14}
10	1 m_8	X m_9	X m_{11}	X m_{10}

$$F_1 = X = D_1 + D_3$$

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

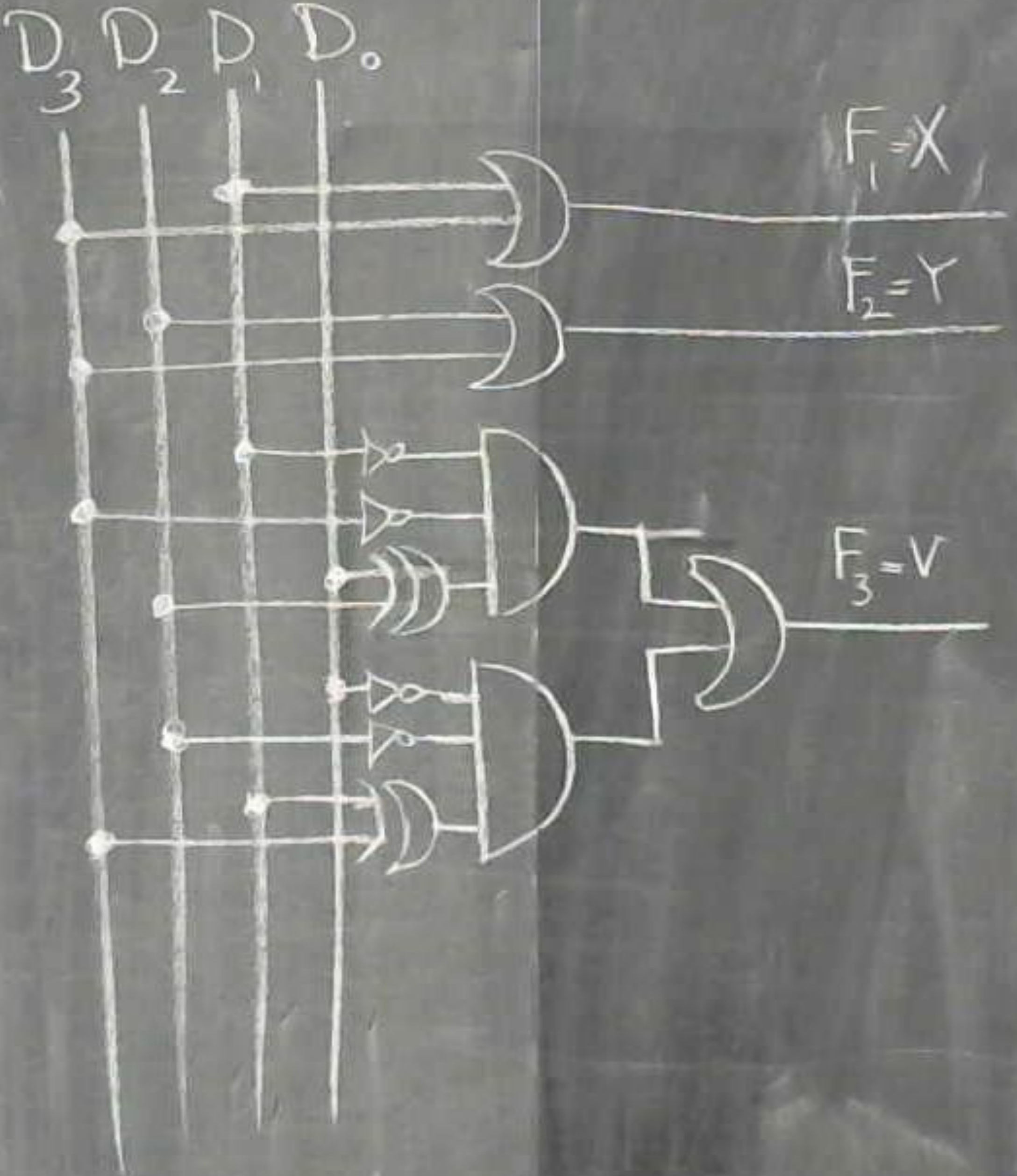
$$F_2 = Y = D_2 + D_3$$

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

$$F_3 = V = D'_3D'_2D'_1D_0 + D'_3D_2D'_1D'_0 + D'_3D'_2D_1D'_0 + D_3D'_2D'_1D'_0$$

$$= D'_3D'_1(D'_2D_0 + D_2D'_0) + D'_2D'_0(D'_3D_1 + D_3D'_1)$$

$$= D'_3D'_1(D_2 \oplus D_0) + D'_2D'_0(D_3 \oplus D_1)$$



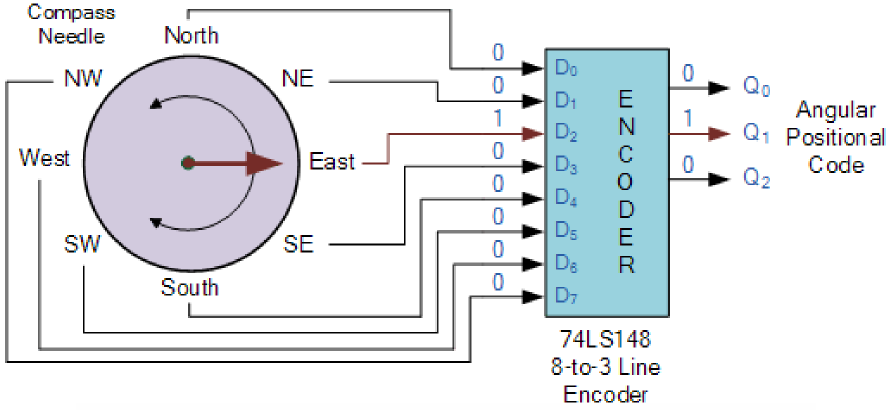
D3 D2 D1 D0
F1 = X
F2 = Y
F3 = V

Priority Encoder

at home!

Positional Encoders

Priority Encoder Navigation

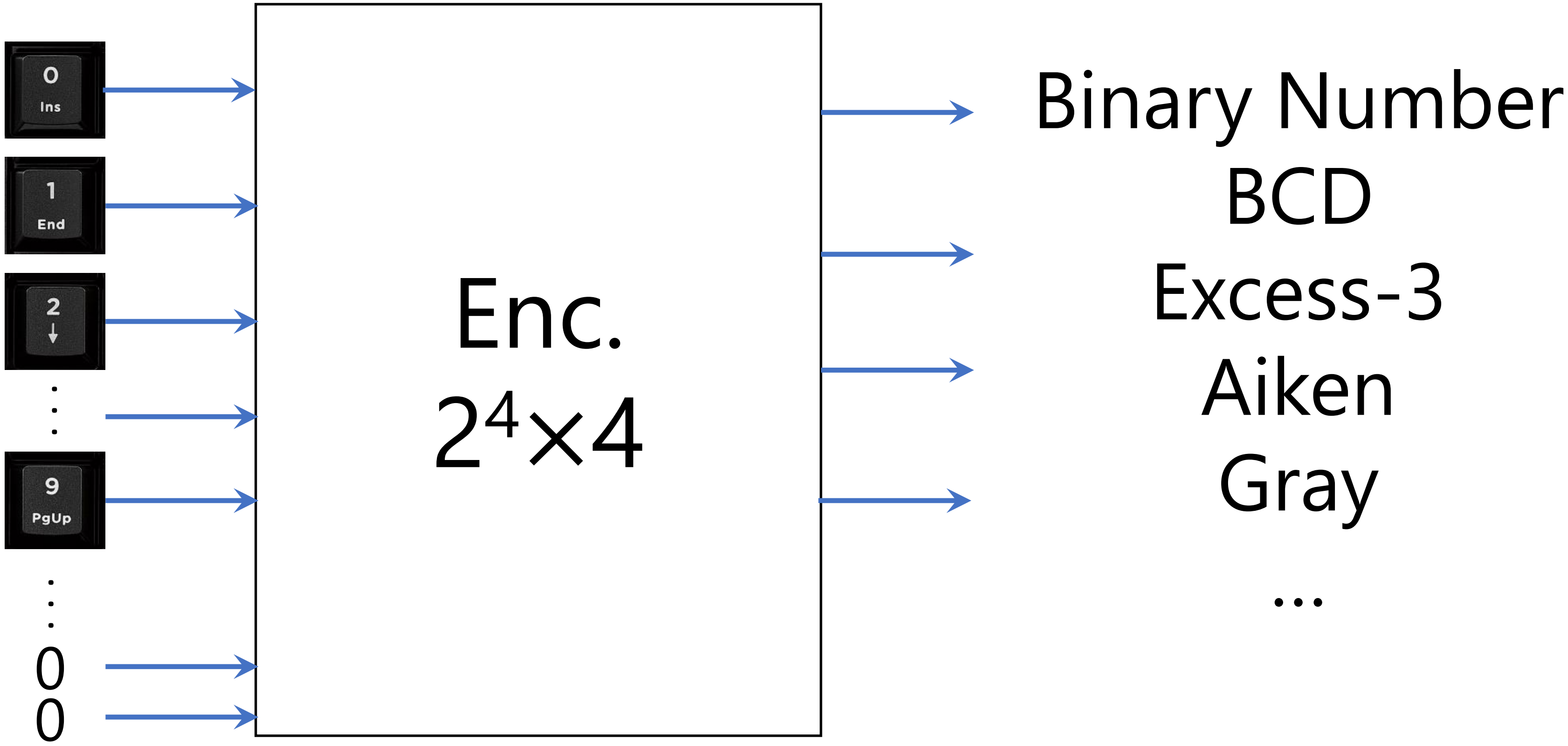


Compass Direction	Binary Output		
	Q_0	Q_1	Q_2
North	0	0	0
North-East	0	0	1
East	0	1	0
South-East	0	1	1
South	1	0	0
South-West	1	0	1
West	1	1	0
North-West	1	1	1

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

Keyboard Encoders

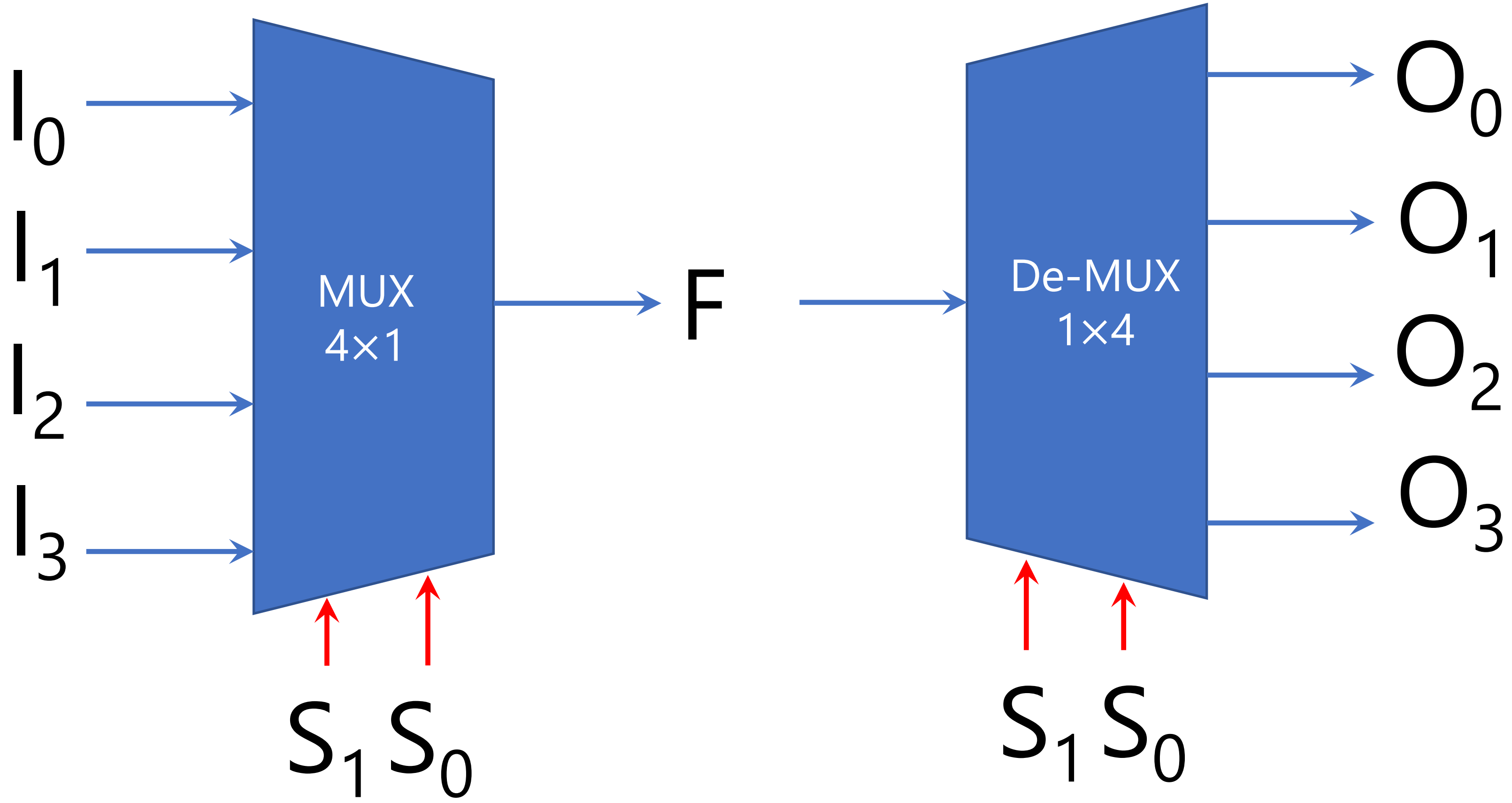
Num Lock
/
*
Back Space
7 Home
8
9 PgUp
−
4
5
6
+
1 End
2
3 PgDn
Enter
0 Ins
000
.
Del



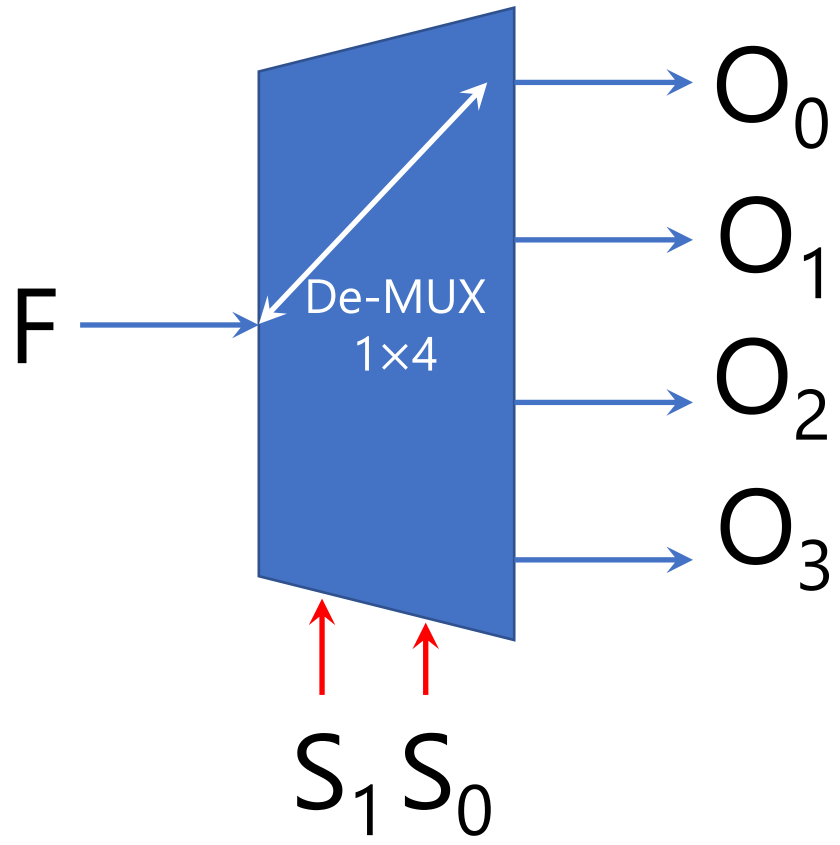
0 Ins
1 End
2
9 PgUp
0
0
Enc.
$2^4 \times 4$
Binary Number
BCD
Excess-3
Aiken
Gray
...


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

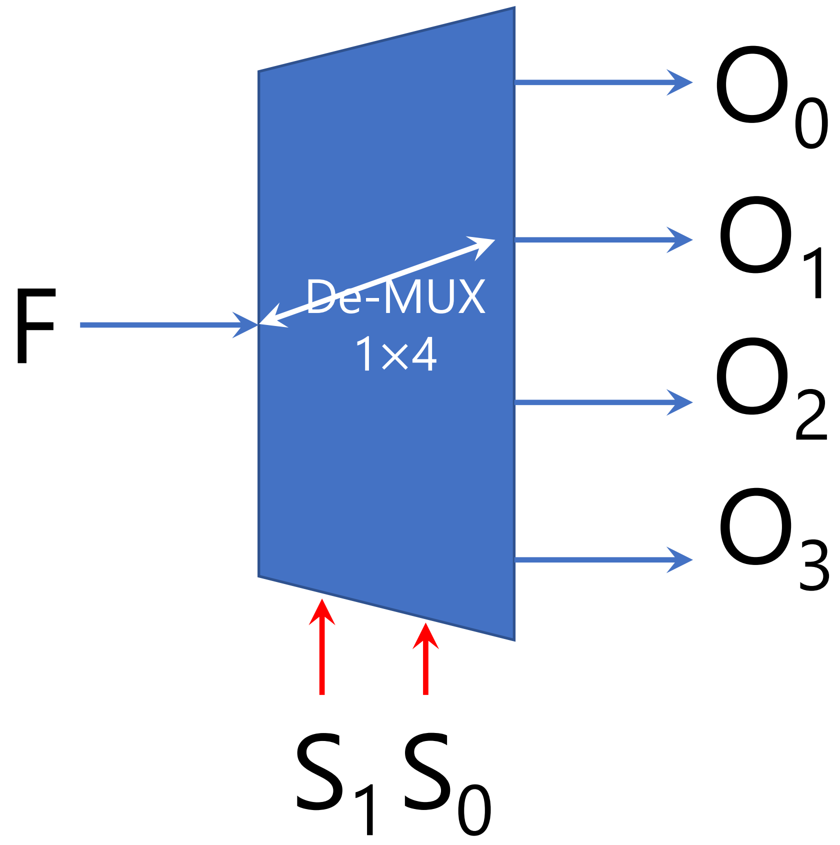
De-multiplexer



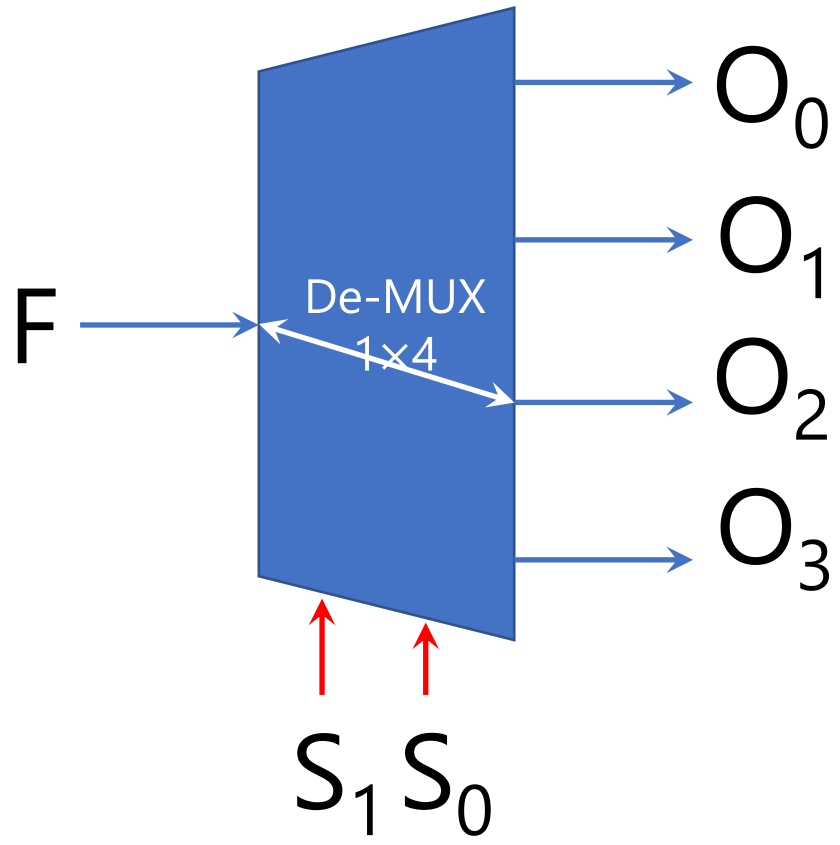
I_0
I_1
I_2
I_3
MUX
4×1
S_1 S_0
F
De-MUX
1×4
O_0
O_1
O_2
O_3
S_1 S_0



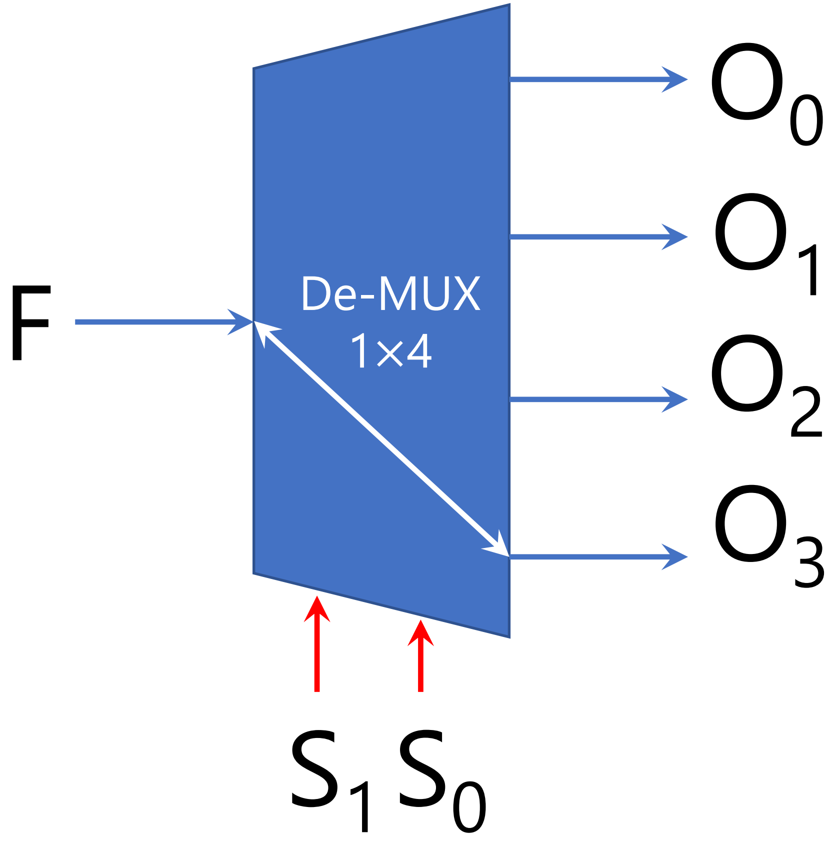
S_1	S_0	F	O_0	O_1	O_2	O_3
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	1
1	1	1	0	0	0	0



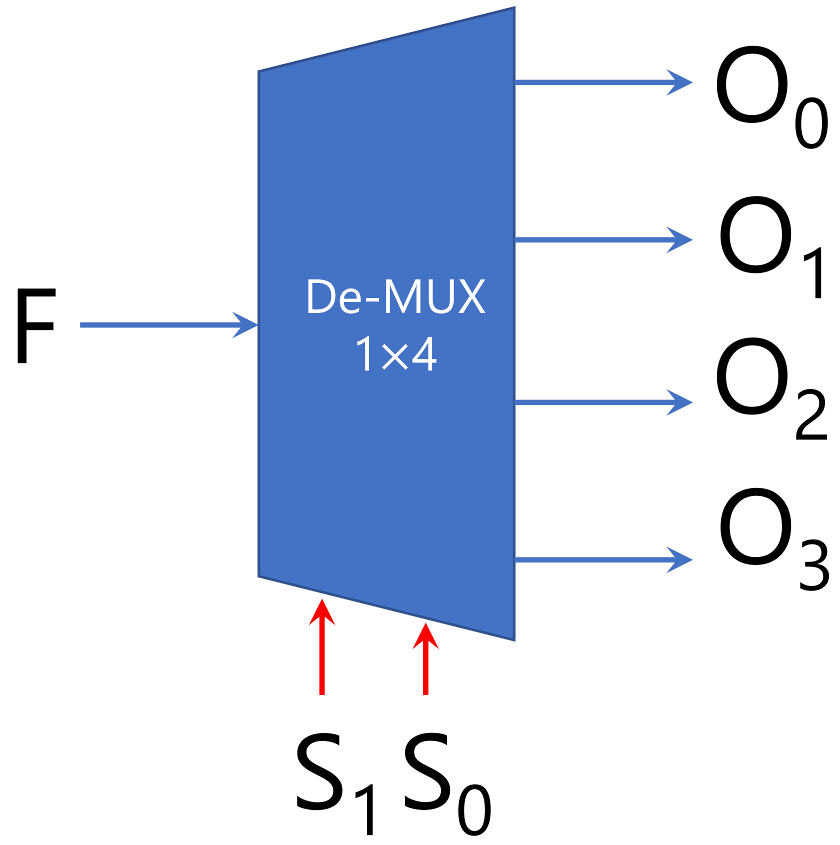
S_1	S_0	F	O_0	O_1	O_2	O_3
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	1
1	1	1	0	0	0	0



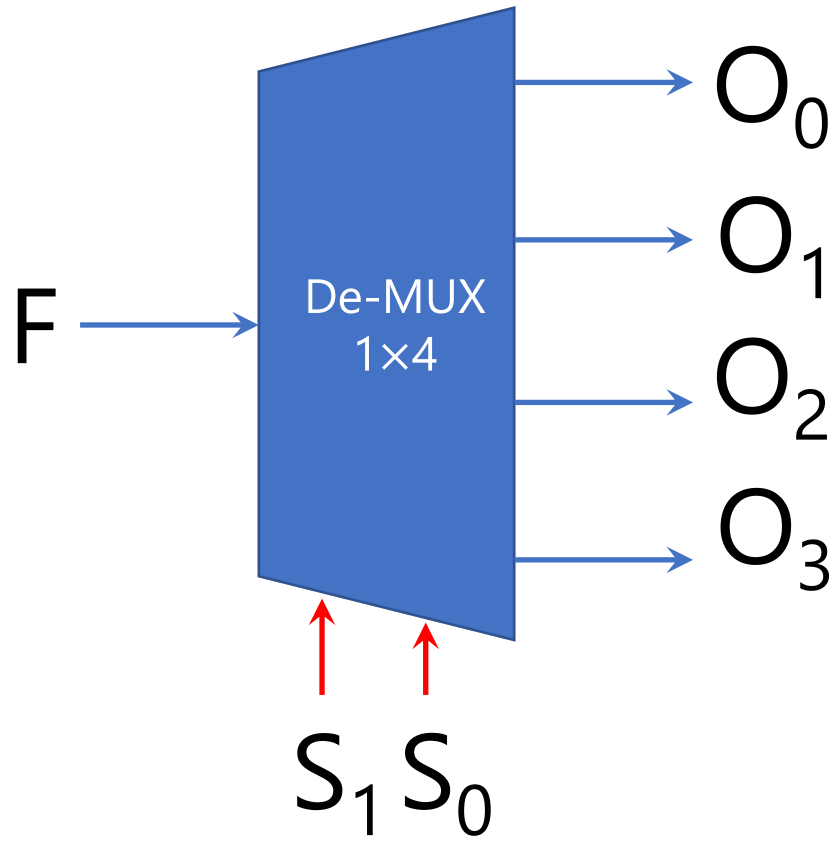
S_1	S_0	F	O_0	O_1	O_2	O_3
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	1
1	1	1	0	0	0	0



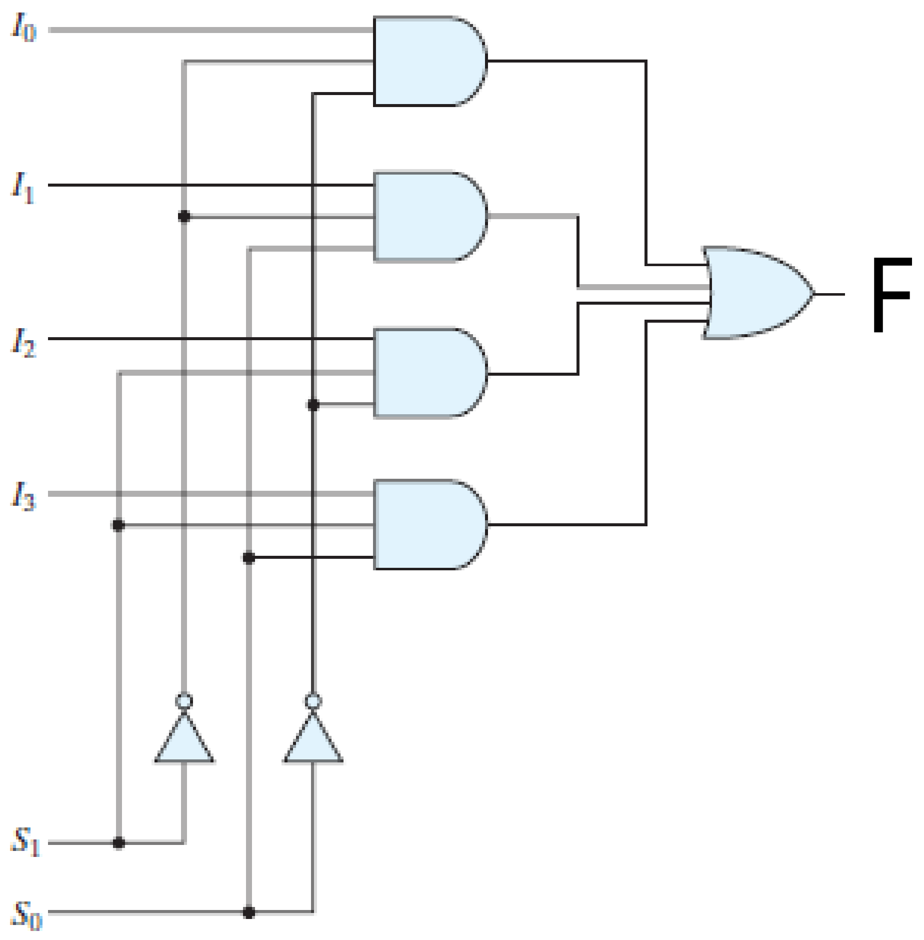
S_1	S_0	F	O_0	O_1	O_2	O_3
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	0
1	1	1	0	0	0	1



S_1	S_0	O_0	O_1	O_2	O_3
0	0	F	0	0	0
0	1	0	F	0	0
1	0	0	0	F	0
1	1	0	0	0	F

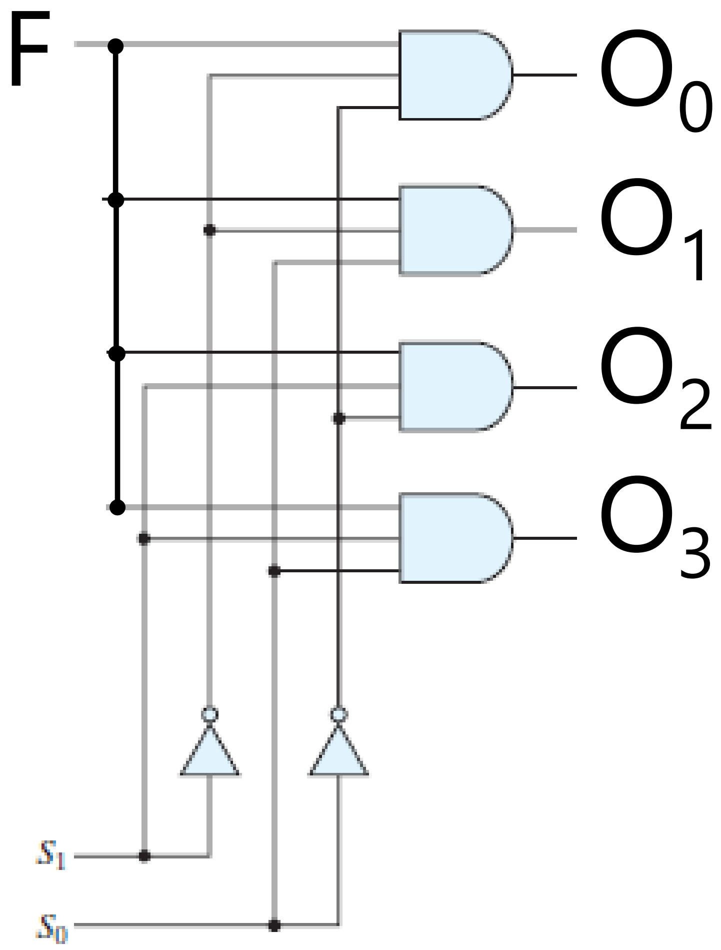


S_1	S_0	$O_0 = S'_1 S'_0 F$	$O_1 = S'_1 S_0 F$	$O_2 = S_1 S'_0 F$	$O_3 = S_1 S_0 F$
0	0	F	0	0	0
0	1	0	F	0	0
1	0	0	0	F	0
1	1	0	0	0	F



(a) Logic diagram

FIGURE 4.25
Four-to-one-line multiplexer



1-to-4 De-mux

De-multiplexer = Decoder w/ Enable input

How come?!