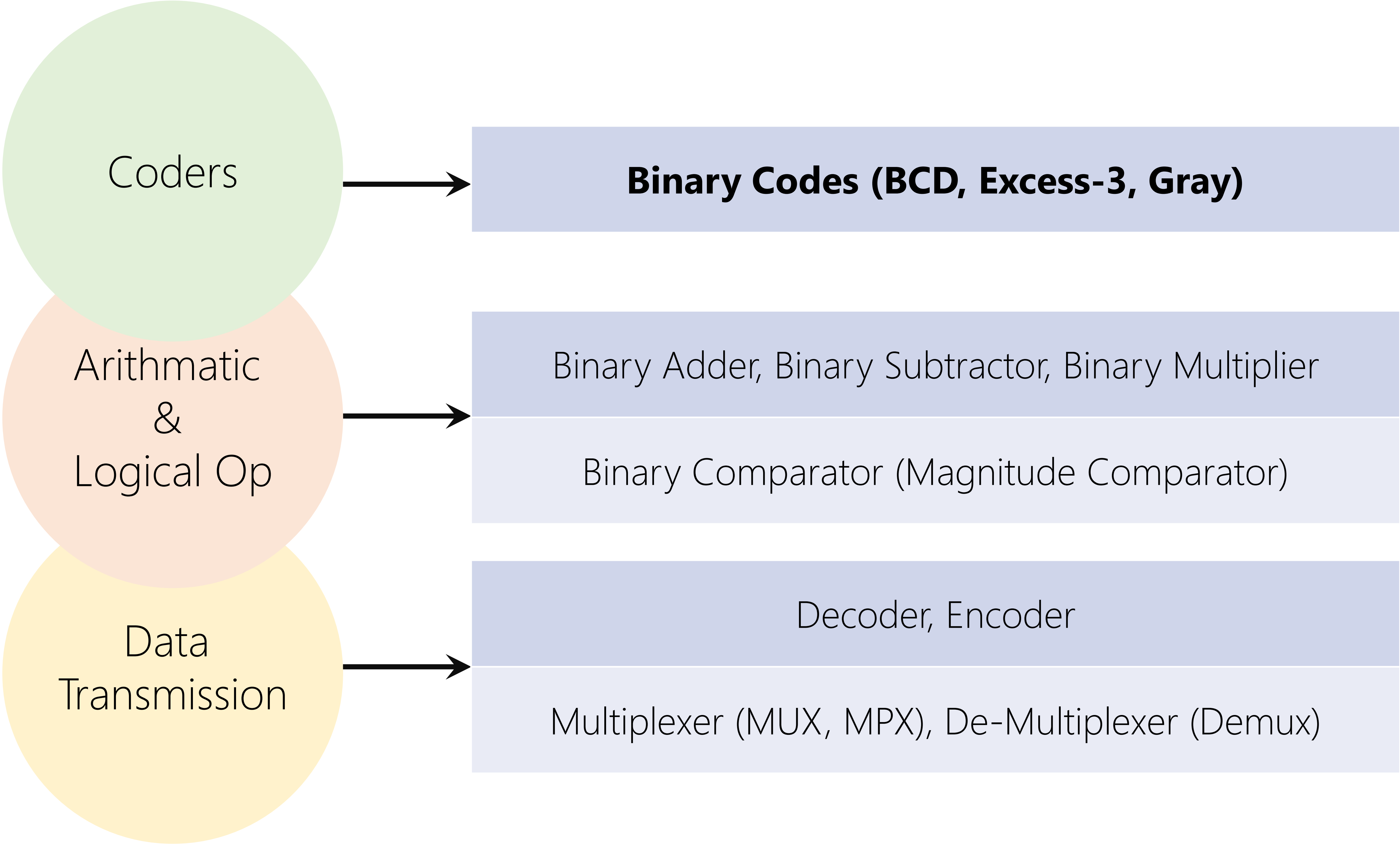
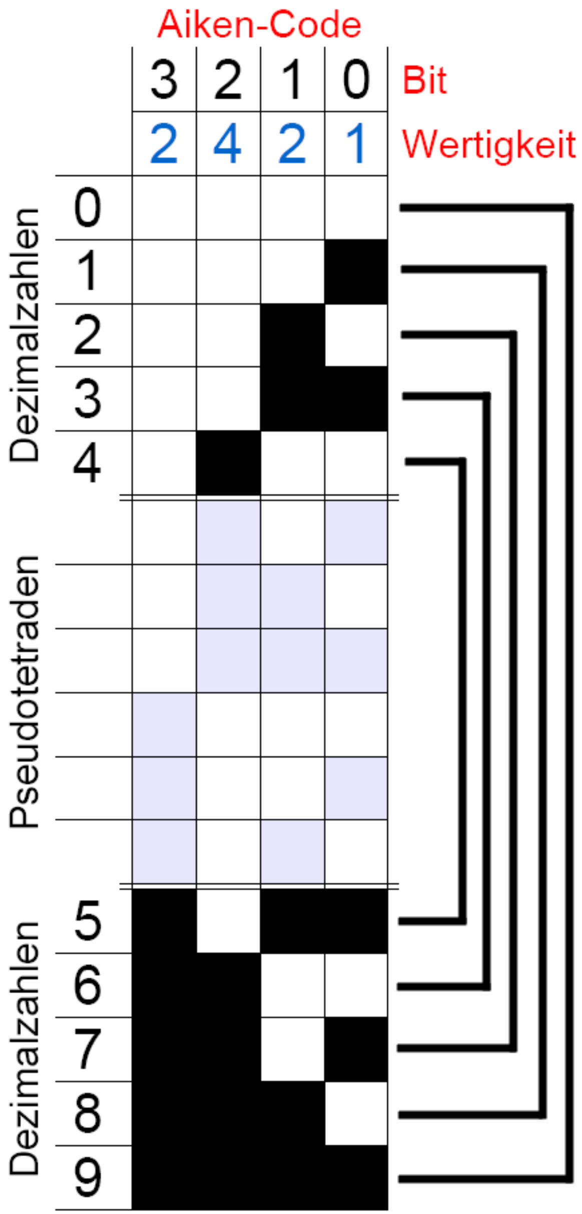
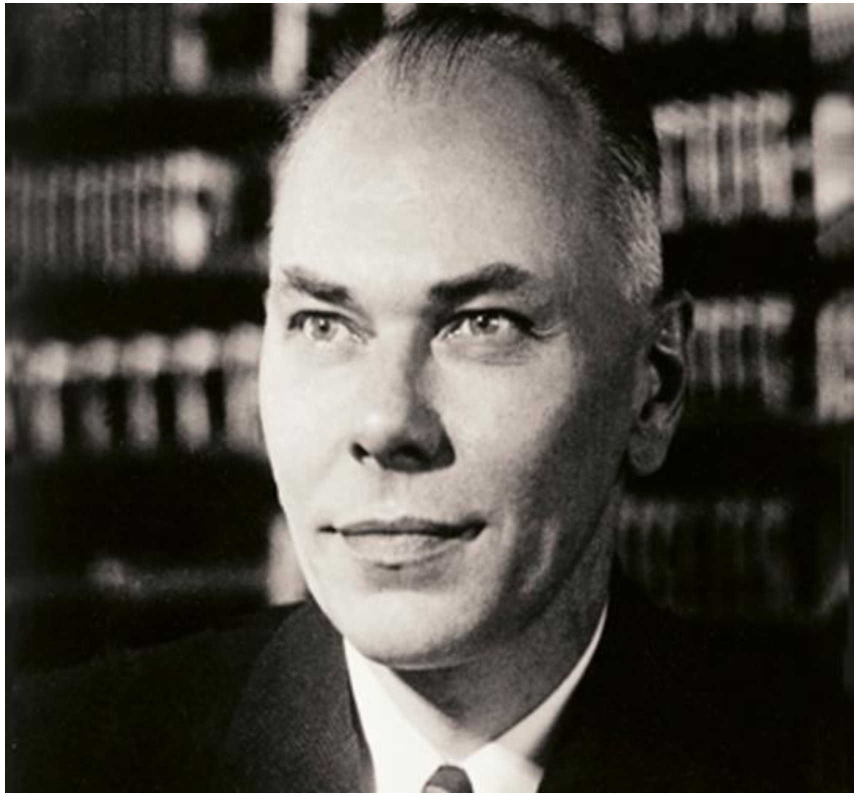

Coders
Binary Codes (BCD, Excess-3, Gray)
Arithmatic
&
Logical Op
Binary Adder, Binary Subtractor, Binary Multiplier
Binary Comparator (Magnitude Comparator)
Data
Transmission
Decoder, Encoder
Multiplexer (MUX, MPX), De-Multiplexer (Demux)





Howard Hathaway Aiken

(March 8, 1900 – March 14, 1973)

Physicist

Pioneer in computing

Original conceptual designer behind IBM's Harvard Mark I

George Robert Stibitz

(April 30, 1904 – January 31, 1995)

Bell Labs researcher

One of the fathers of the modern first digital computer

Table 1.5

Four Different Binary Codes for the Decimal Digits

Decimal Digit	BCD 8421	Aiken 2421	+3 Excess-3	8, 4, −2, −1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111

Self-complementing

The 9’s complement of the decimal number

=

The 1’s complement (NOT) of its binary code

THE INTERNATIONAL
Calculator Collector

Spring 1993 Issue No. 1

The Cal Tech circa 1967 *Photo Courtesy Texas Instruments*

The Beginning

If you're past your mid-30s, you probably remember your first simple hand-held calculator costing over $50 (in early 1970's dollars). Depending how much older you are, your first could have been upwards to $400. And we're just talking the basic four functions here — addition, subtraction, multiplication, and division. Percentage and memory features were extra (if they were even available at that point in time).

Company Profile:

Bowmar

Who can forget the "Bowmar Brain" series of calculators from the early '70s?

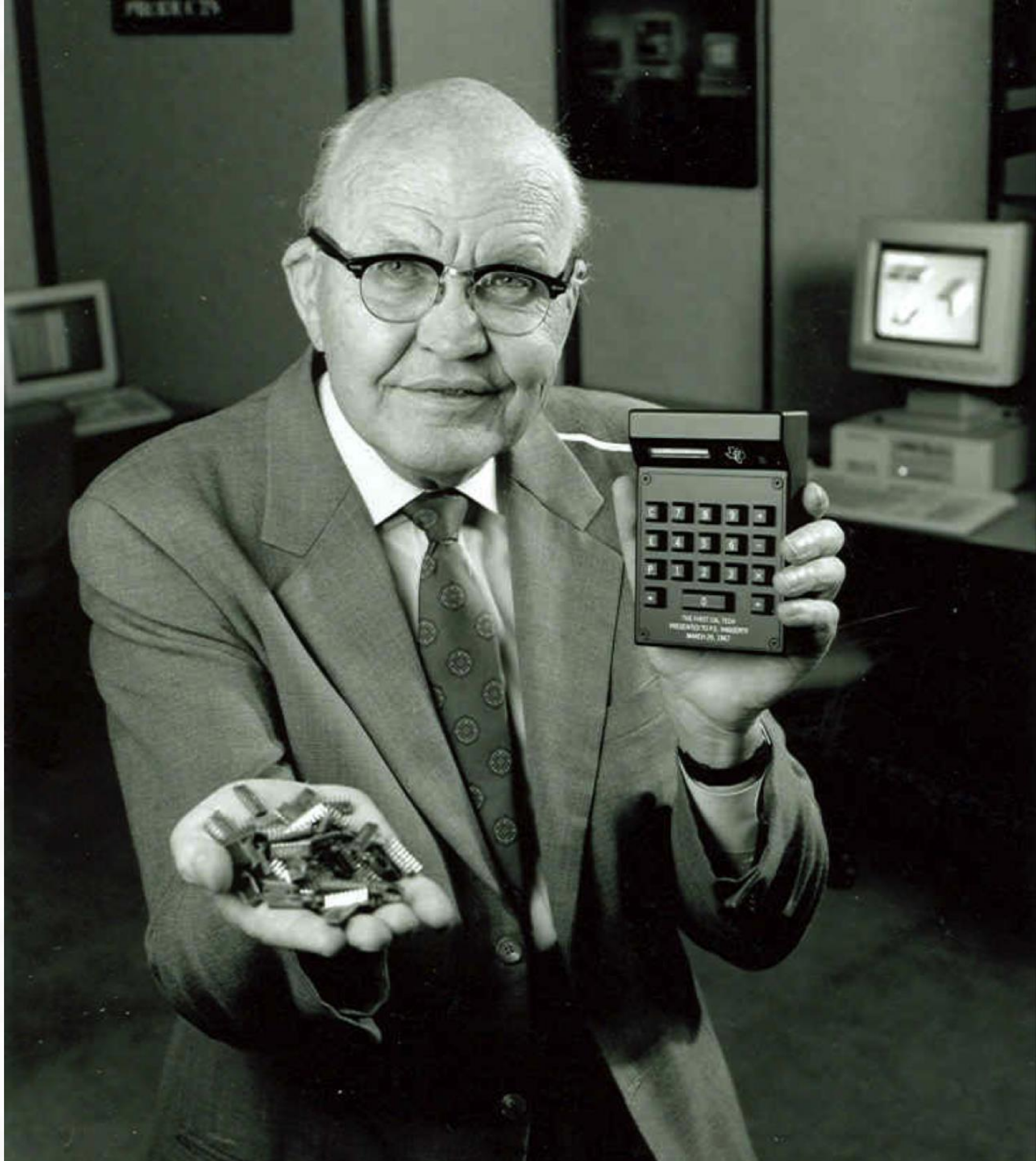
Bowmar was the first American company that made and sold their own line of portable electronic machines.

The story starts around 1970 when Bowmar, then a manufacturer of Light Emitting Diodes (LEDs), tried to sell their numeric display product to Japanese manufacturers for use in their electronic products.

Bowmar wasn't too successful. The Japanese were using a flourescent style display that was cheaper and had a few design features the manufacturers liked better.

So, president Ed White, a consummate entrepreneur, and his staff came up with an even better idea — make the whole electronic calculator themselves.

Up to now, most of the so-called "portable" calculators



Gray Code

Analog → Digital

Straight binary number sequence for 7 to 8: 0111 → 1000; causes all four bits to change values.

Gray code for 7 → 8: 0100 to 1100; only the first bit changes from 0 to 1; the other three bits remain the same.

Gray Code
Algorithm

Step 0: Convert the decimal number to binary number.
Step 1: The MSB (Most Significant Bit) of a gray code and binary code is **the same**.
Step 2: The next digit of gray code is the XOR of the previous and current digit in the binary code.

ASCII Code

American Standard Code for Information Interchange

USASCII code chart

Bits				$b_7 b_6 b_5$	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4	b_3	b_2	b_1	Column → / Row ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	−	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

“0” $= (011\ 0000)_2 = (48)_{10}$

Table 1.7
American Standard Code for Information Interchange (ASCII)

	$b_7b_6b_5$							
$b_4b_3b_2b_1$	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	–	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Control Characters

NUL	Null	DLE	Data-link escape
SOH	Start of heading	DC1	Device control 1
STX	Start of text	DC2	Device control 2
ETX	End of text	DC3	Device control 3
EOT	End of transmission	DC4	Device control 4
ENQ	Enquiry	NAK	Negative acknowledge
ACK	Acknowledge	SYN	Synchronous idle
BEL	Bell	ETB	End-of-transmission block
BS	Backspace	CAN	Cancel
HT	Horizontal tab	EM	End of medium
LF	Line feed	SUB	Substitute
VT	Vertical tab	ESC	Escape
FF	Form feed	FS	File separator
CR	Carriage return	GS	Group separator
SO	Shift out	RS	Record separator
SI	Shift in	US	Unit separator
SP	Space	DEL	Delete

Combinational Logic
Binary Codes

Combinational Logic

Code Conversion

Decimal Equivalent	BCD 8421	Aiken 2421	Excess-3	8, 4, −2, −1	Gray Code
0	0000	0000	0011	0000	0000
1	0001	0001	0100	0111	0001
2	0010	0010	0101	0110	0011
3	0011	0011	0110	0101	0010
4	0100	0100	0111	0100	0110
5	0101	1011	1000	1011	0111
6	0110	1100	1001	1010	0101
7	0111	1101	1010	1001	0100
8	1000	1110	1011	1000	1100
9	1001	1111	1100	1111	1101
10					1111
11					1110
12					1010
13					1011
14					1001
15					1000

Decimal Equivalent	BCD 8421	Aiken 2421	Excess-3	8, 4, −2, −1	Gray Code
0	0000	0000	0011	0000	0000
1	0001	0001	0100	0111	0001
2	0010	0010	0101	0110	0011
3	0011	0011	0110	0101	0010
4	0100	0100	0111	0100	0110
5	0101	1011	1000	1011	0111
6	0110	1100	1001	1010	0101
7	0111	1101	1010	1001	0100
8	1000	1110	1011	1000	1100
9	1001	1111	1100	1111	1101
10	0001 0000	0001 0000	You fill it at home	You fill it at home	1111
11	0001 0001	0001 0001			1110
12	0001 0010	0001 0010			1010
13	0001 0011	0001 0011			1011
14	0001 0100	0001 0100			1001
15	0001 0101	0001 1011			1000

Decimal Equivalent	BCD 8421	Aiken 2421	Excess-3	8, 4, −2, −1	Gray Code
0	0000	0000	0011	0000	0000
1	0001	0001	0100	0111	0001
2	0010	0010	0101	0110	0011
3	0011	0011	0110	0101	0010
4	0100	0100	0111	0100	0110
5 ↔	0101 ↔	1011 ↔	1000 ↔	1011 ↔	0111
6	0110	1100	1001	1010	0101
7	0111	1101	1010	1001	0100
8	1000	1110	1011	1000	1100
9	1001	1111	1100	1111	1101
10	0001 0000	0001 0000	You fill it at home	You fill it at home	1111
11	0001 0001	0001 0001			1110
12	0001 0010	0001 0010			1010
13	0001 0011	0001 0011			1011
14	0001 0100	0001 0100			1001
15	0001 0101	0001 1011			1000

Combinational Logic

Code Conversion

BCD (8421) → Excess-3

Table 4.2
Truth Table for Code Conversion Example

Input BCD				Output Excess-3 Code			
A	*B*	*C*	*D*	*w*	*x*	*y*	*z*
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	?	?	?	?
1	0	1	1	?	?	?	?
1	1	0	0	?	?	?	?
1	1	0	1	?	?	?	?
1	1	1	0	?	?	?	?
1	1	1	1	?	?	?	?

Don’t Care Conditions

In practice, in some applications the function is not specified for certain combinations of the variables.

Don’t Care Conditions

Functions that have unspecified outputs for some input combinations are called *incompletely specified functions*.

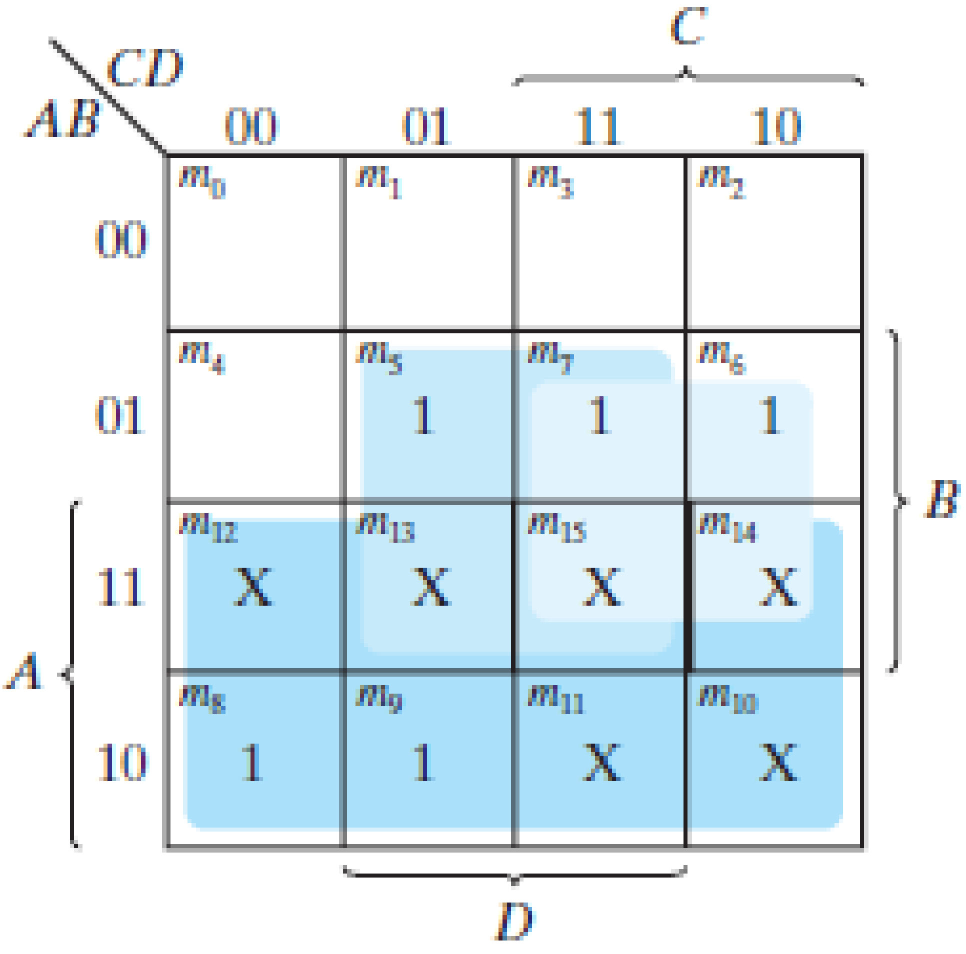
Don’t-care conditions can be used on a map to provide further simplification of the Boolean expression.

Don’t Care Conditions

To distinguish the don’t-care condition from 1’s and 0’s, an x is used.

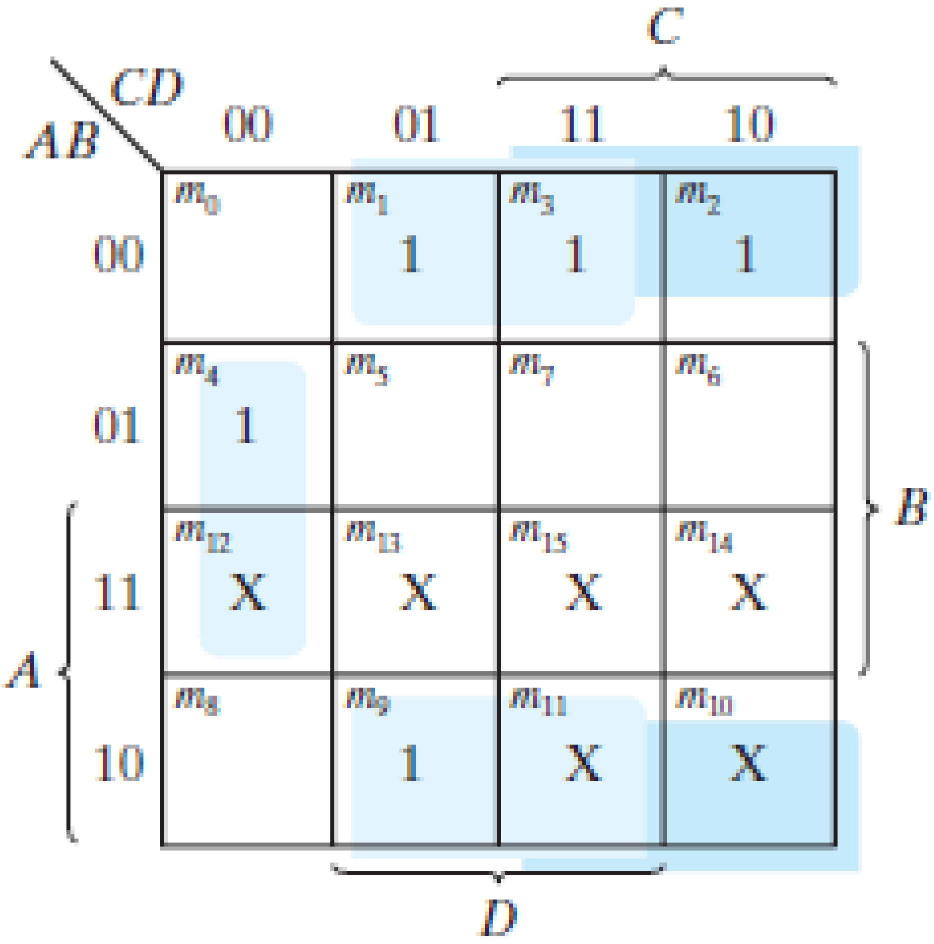
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

$W(A,B,C,D) = \sum(5,6,7,8,9) + d(10,11,12,13,14,15)$



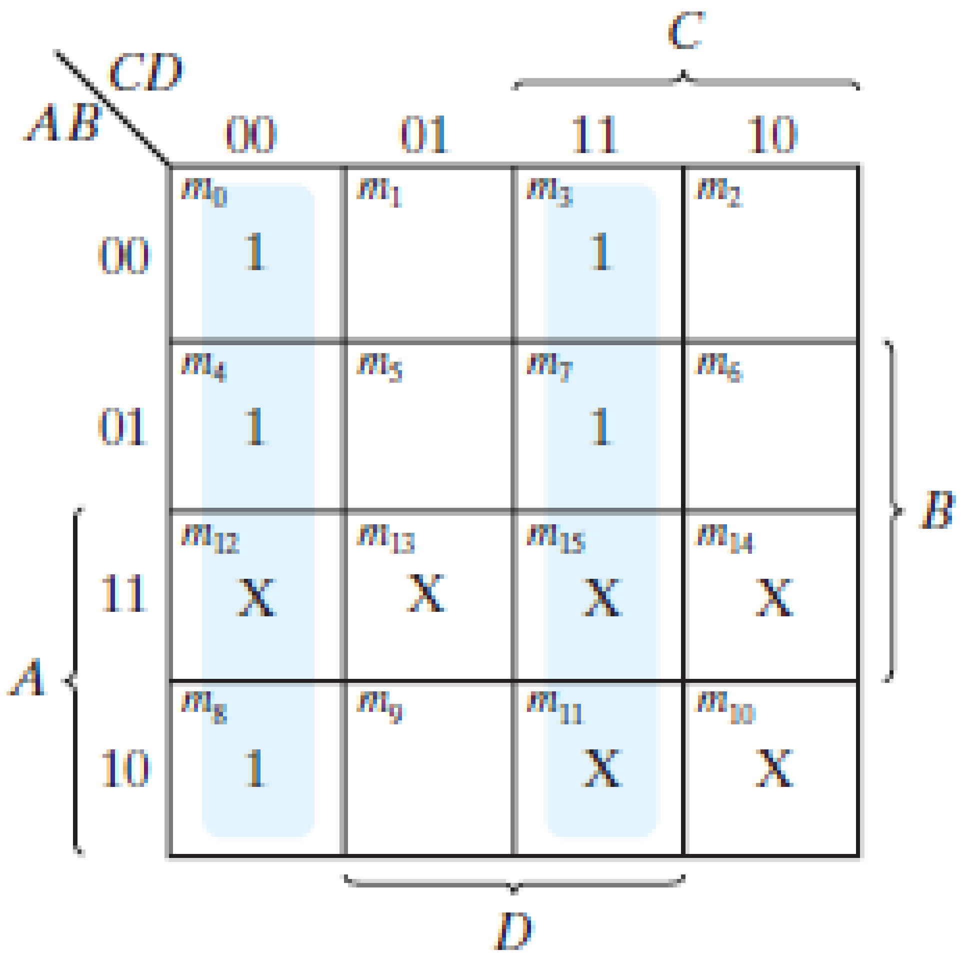
$w = A + BC + BD$

$$X(A,B,C,D) = \sum(1,2,3,4,9) + d(10,11,12,13,14,15)$$



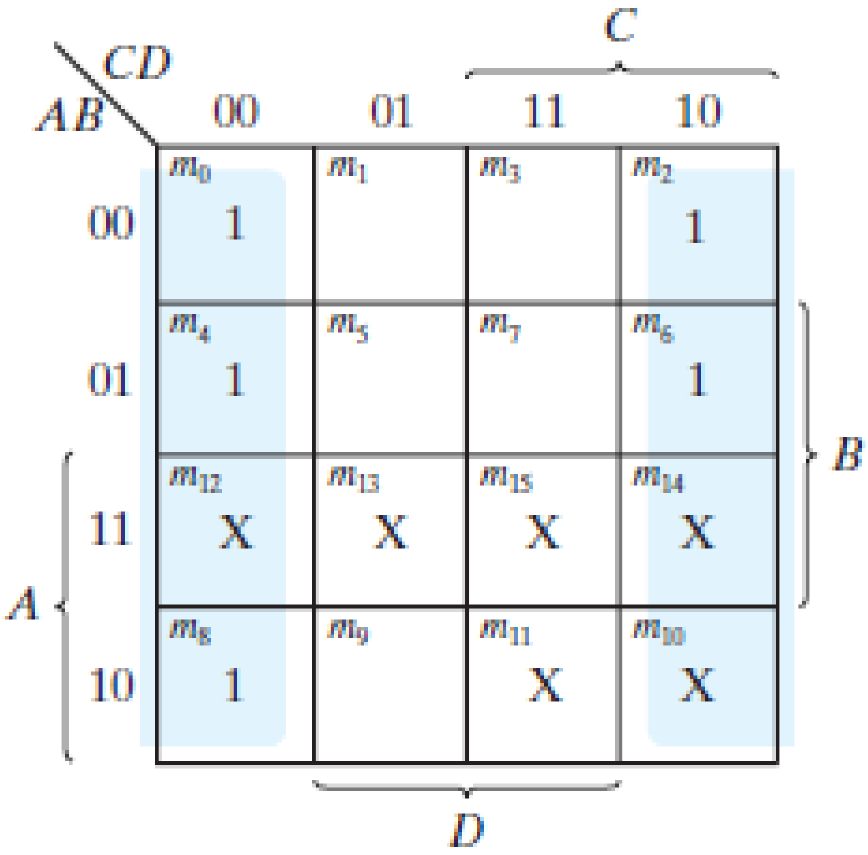
$$x = B'C + B'D + BC'D'$$

Y(A,B,C,D) = $\sum(0,3,4,7,8) + d(10,11,12,13,14,15)$



$y = CD + C'D'$

$Z(A,B,C,D) = \sum(0,2,4,6,8) + d(10,11,12,13,14,15)$



$z = D'$

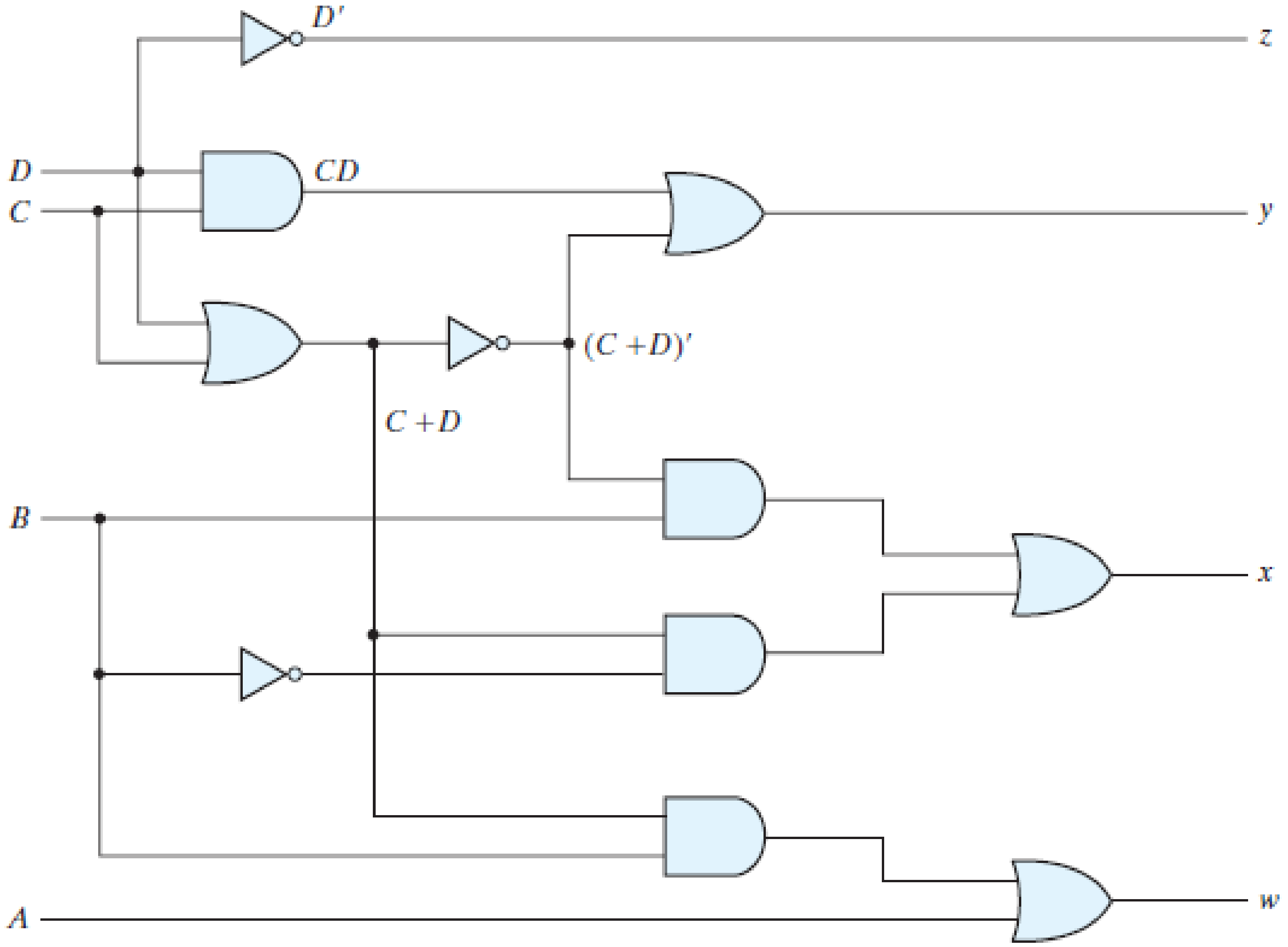


FIGURE 4.4
Logic diagram for BCD-to-excess-3 code converter

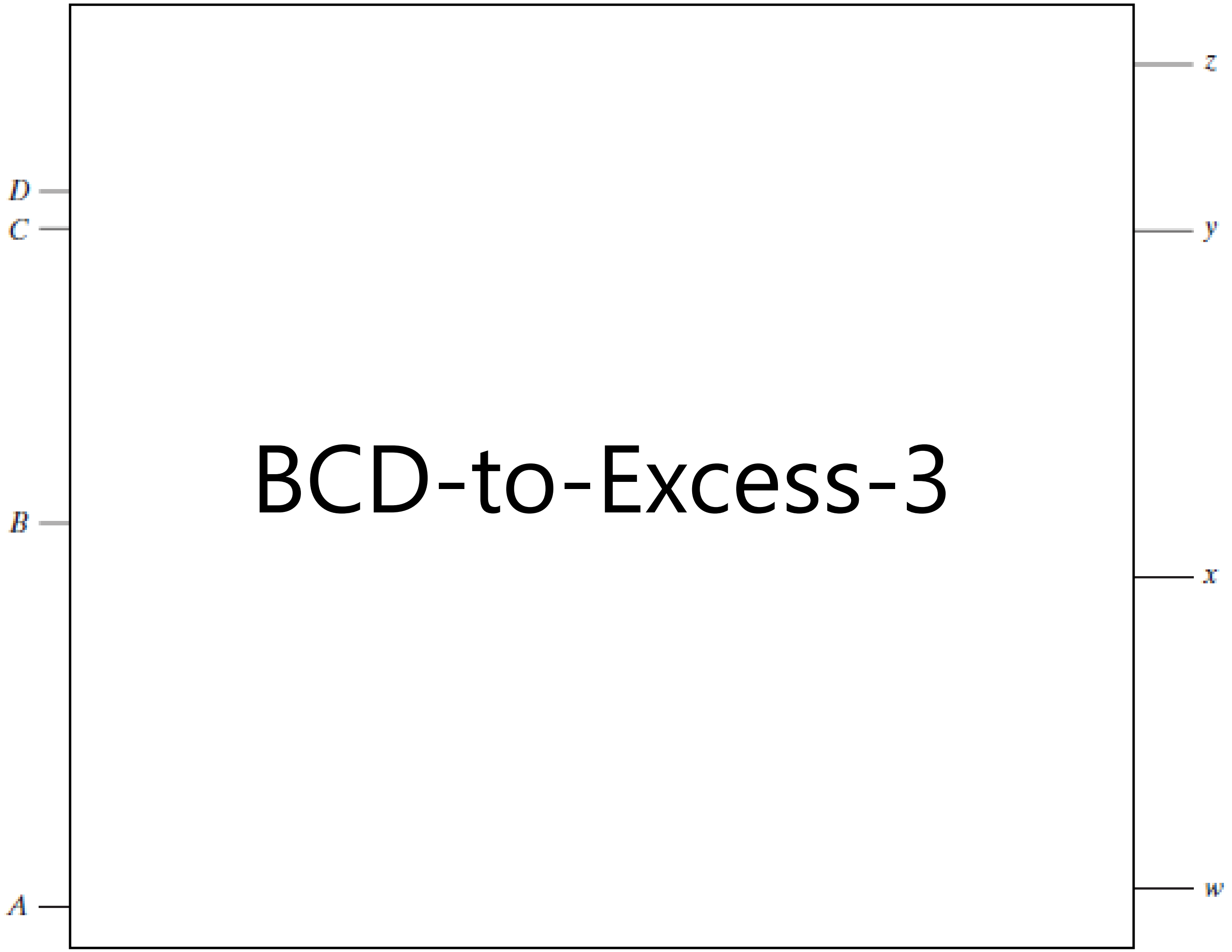


FIGURE 4.4
Logic diagram for BCD-to-excess-3 code converter

MAXTERM

$$
\begin{aligned}
W(A,B,C,D) &= \sum(5,6,7,8,9) + d(10,11,12,13,14,15) \\
&= \prod(?)
\end{aligned}
$$

$$
\begin{aligned}
W(A,B,C,D) &= \sum(5,6,7,8,9) + d(10,11,12,13,14,15) \\
&= \prod(0,1,2,3,4)
\end{aligned}
$$

$$W(A,B,C,D) = \sum(5,6,7,8,9) + d(10,11,12,13,14,15)$$
$$= \prod(0,1,2,3,4) + D(10,11,12,13,14,15)$$

We can assume the don't care conditions are 0 if they help to more simplification

$$
\begin{aligned}
W(A,B,C,D) &= \sum(5,6,7,8,9) + d(10,11,12,13,14,15) \\
&= \prod(0,1,2,3,4) + D(10,11,12,13,14,15) \\
&= ()'
\end{aligned}
$$

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

$$W(A,B,C,D) = \sum(5,6,7,8,9) + d(10,11,12,13,14,15)$$
$$= \prod(0,1,2,3,4) + D(10,11,12,13,14,15)$$
$$= ((A'B'))'$$

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

$$
\begin{aligned}
W(A,B,C,D) &= \sum(5,6,7,8,9) + d(10,11,12,13,14,15) \\
&= \prod(0,1,2,3,4) + D(10,11,12,13,14,15) \\
&= ((A'B')+(A'C'D'))'
\end{aligned}
$$

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

$$
\begin{aligned}
W(A,B,C,D) &= \sum(5,6,7,8,9) + d(10,11,12,13,14,15) \\
&= \prod(0,1,2,3,4) + D(10,11,12,13,14,15) \\
&= ((A'B')+(A'C'D'))' \quad \text{Here the “don’t care conditions” did not help ☹} \\
&= (A+B)(A+C+D)
\end{aligned}
$$

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

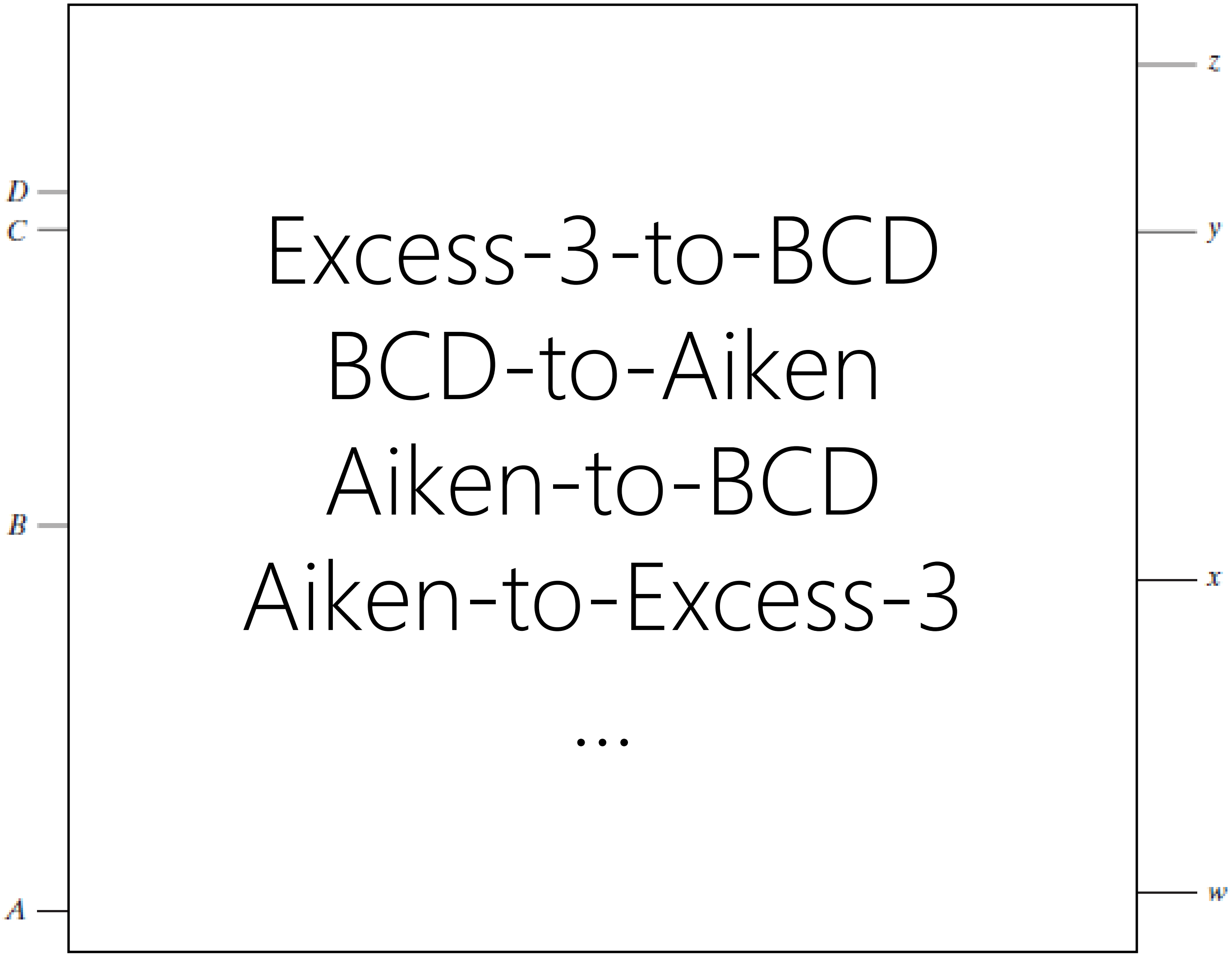
$$X(A,B,C,D) = \sum(1,2,3,4,9) + d(10,11,12,13,14,15)$$
$$= \prod(0,5,6,7,8) + D(10,11,12,13,14,15)$$
$$= ((BD)+(BC)+(B'C'D'))'$$ Here the "don't care conditions" helped ☺
$$= (B'+D')(B'+C')(B+C+D)$$

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

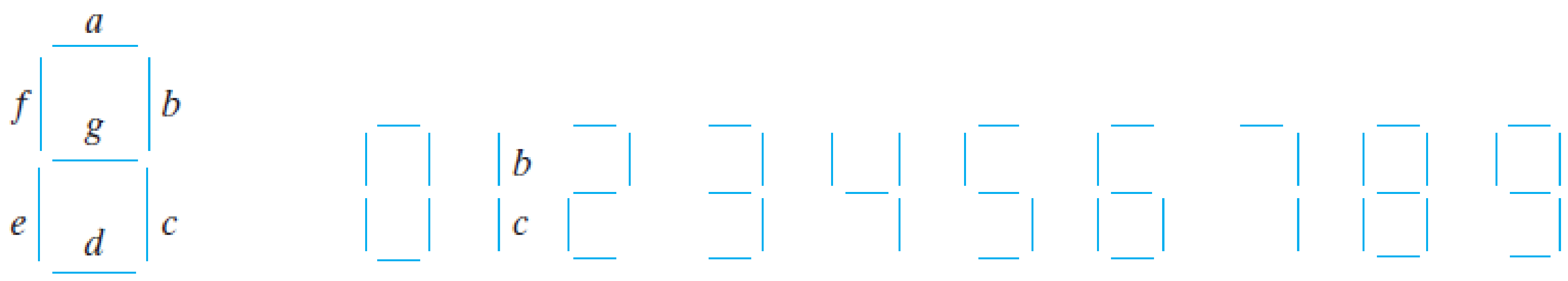
$$Y(A,B,C,D) = \sum(0,3,4,7,8) + d(10,11,12,13,14,15)$$
$$= ?$$

$$Z(A,B,C,D) = \sum(0,2,4,6,8) + d(10,11,12,13,14,15)$$
$$= ?$$

Your Turn!


D
C
B
A
Excess-3-to-BCD
BCD-to-Aiken
Aiken-to-BCD
Aiken-to-Excess-3
...
z
y
x
w

4.9 An ABCD-to-seven-segment decoder is a combinational circuit that converts a decimal digit in BCD to an appropriate code for the selection of segments in an indicator used to display the decimal digit in a familiar form. The seven outputs of the decoder *(a, b, c, d, e, f, g)* select the corresponding segments in the display, as shown in Fig. P4.9(a). The numeric display chosen to represent the decimal digit is shown in Fig. P4.9(b). Using a truth table and Karnaugh maps, design the BCD-to-seven-segment decoder using a minimum number of gates. The six invalid combinations should result in a blank display. (HDL—see Problem 4.51.)



(a) Segment designation

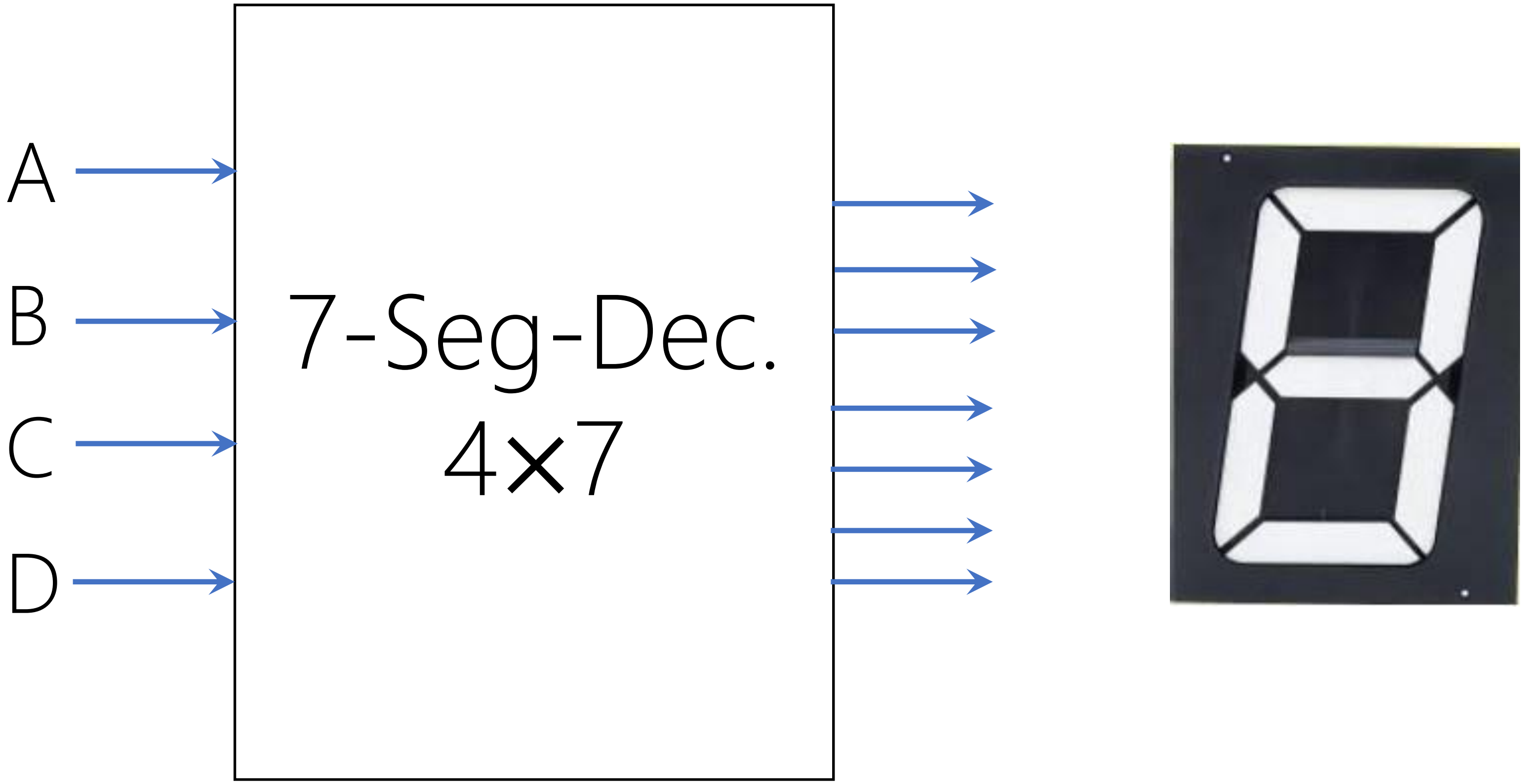
(b) Numerical designation for display

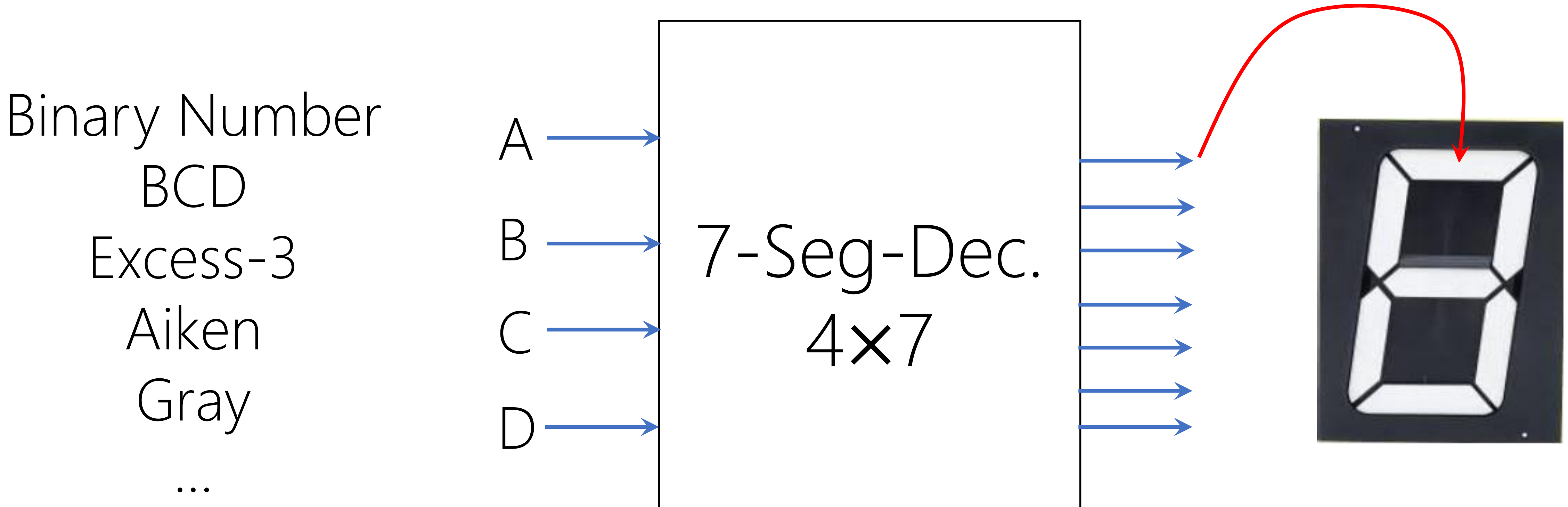
FIGURE P4.9

Combinational Logic

Display Decoder

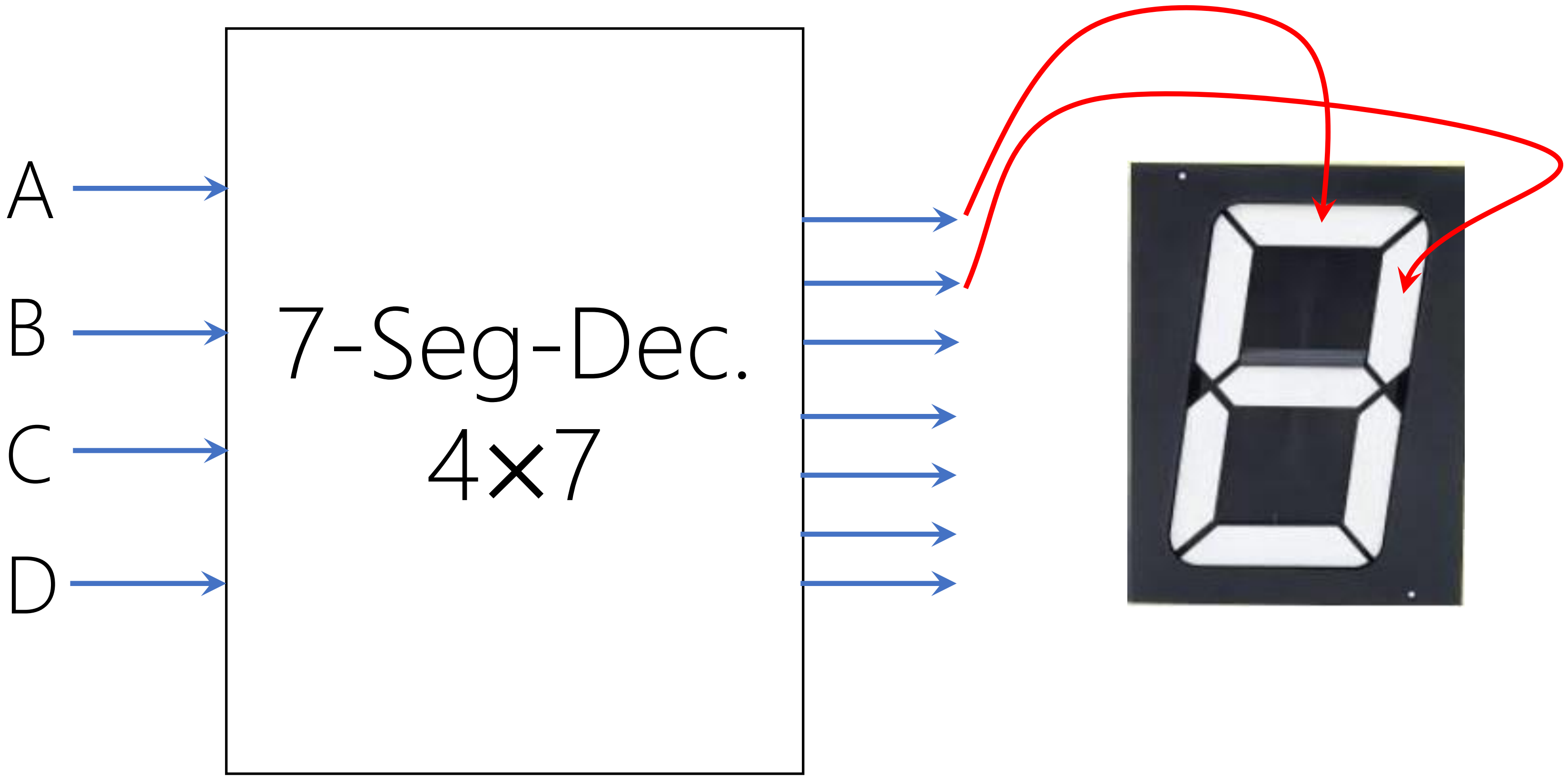
Binary Number
BCD
Excess-3
Aiken
Gray
...



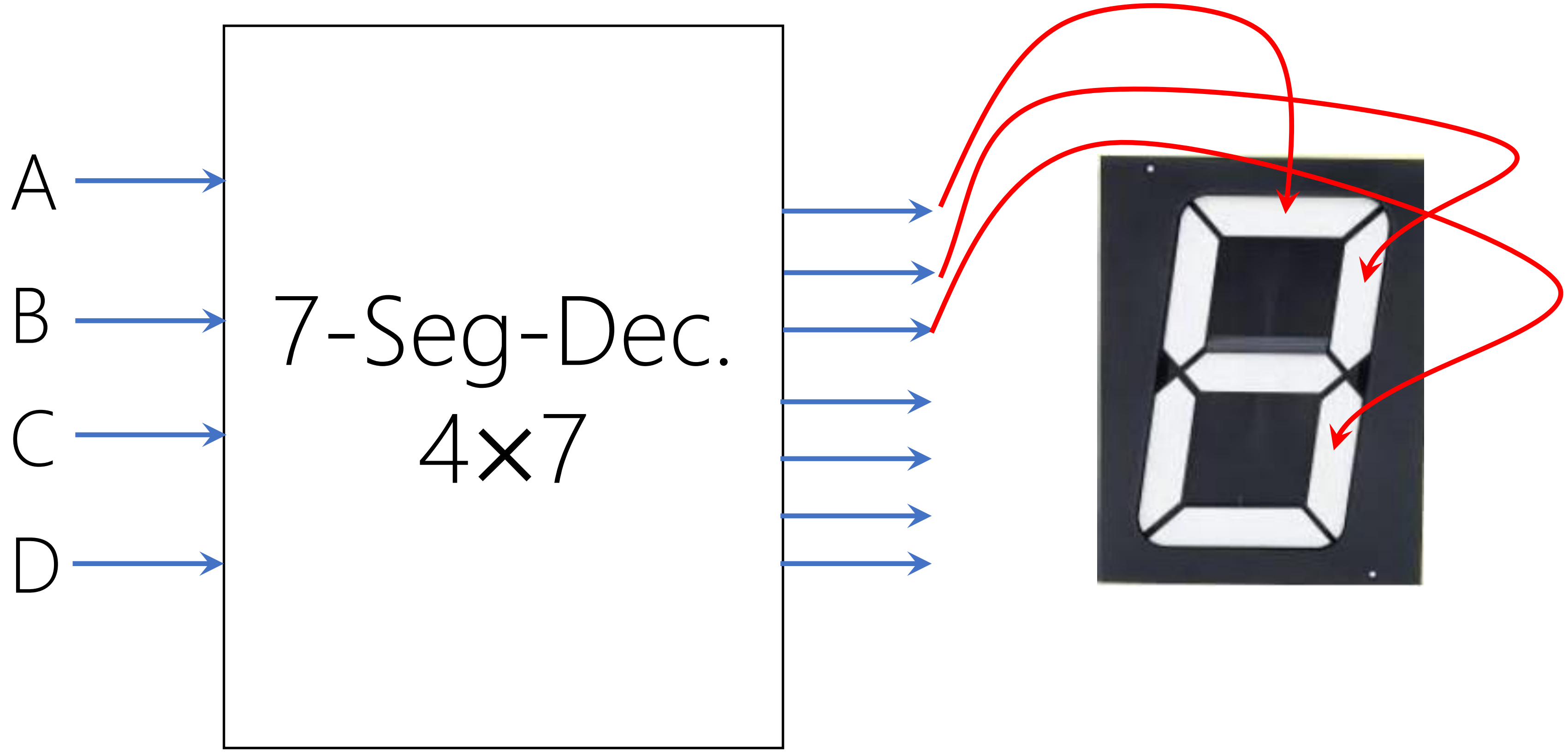


Binary Number
BCD
Excess-3
Aiken
Gray
...
A
B
C
D
7-Seg-Dec.
4×7

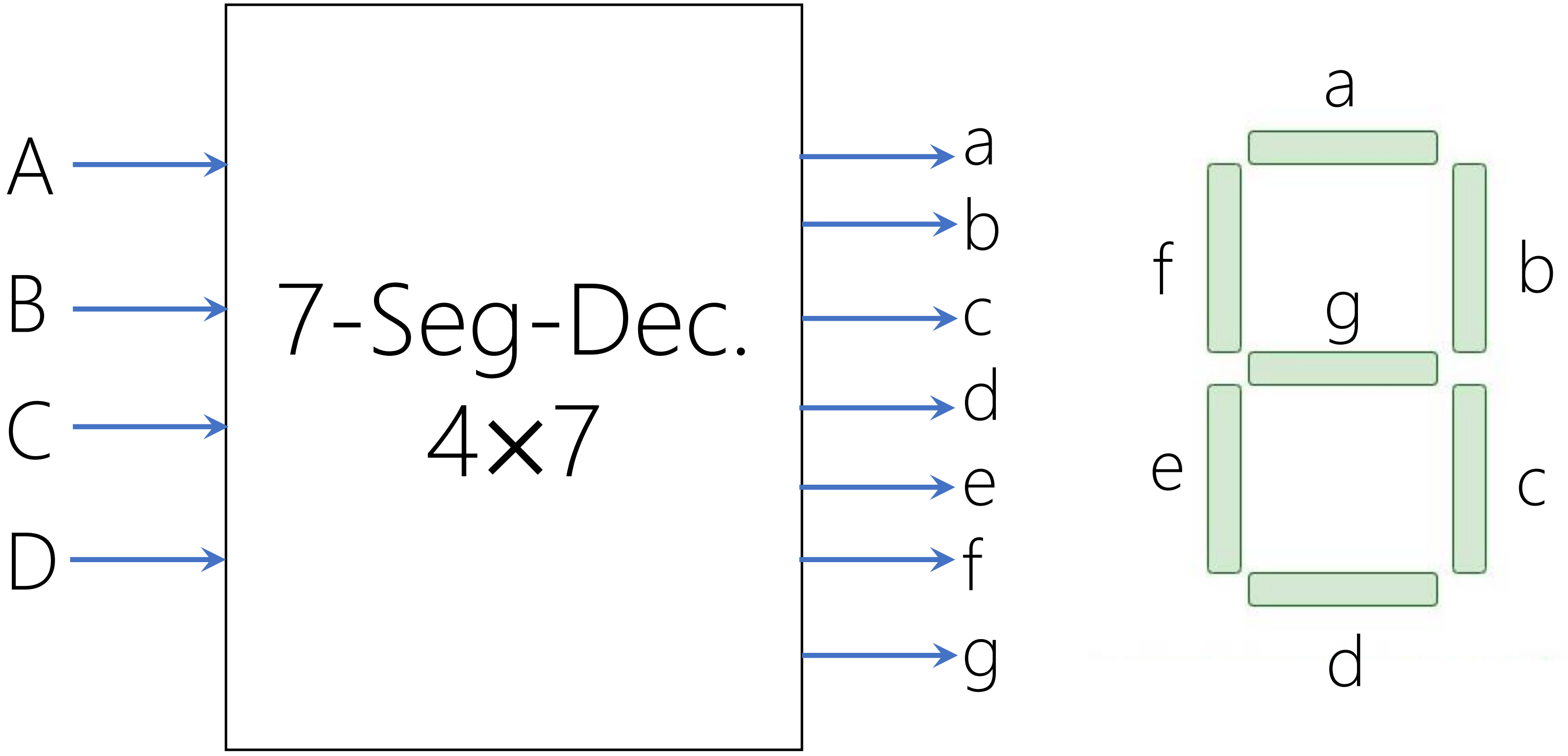
Binary Number
BCD
Excess-3
Aiken
Gray
...



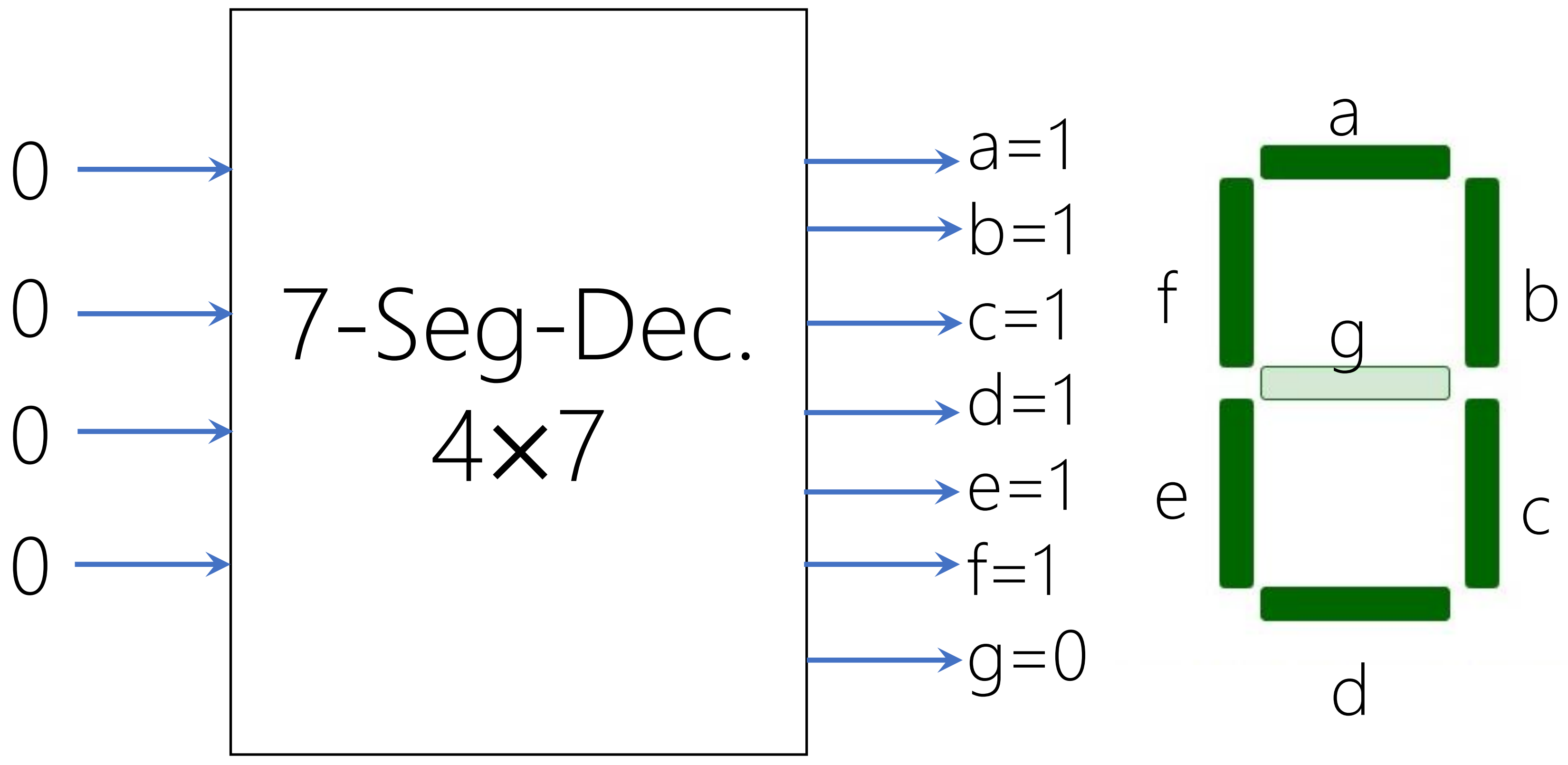
Binary Number
BCD
Excess-3
Aiken
Gray
...



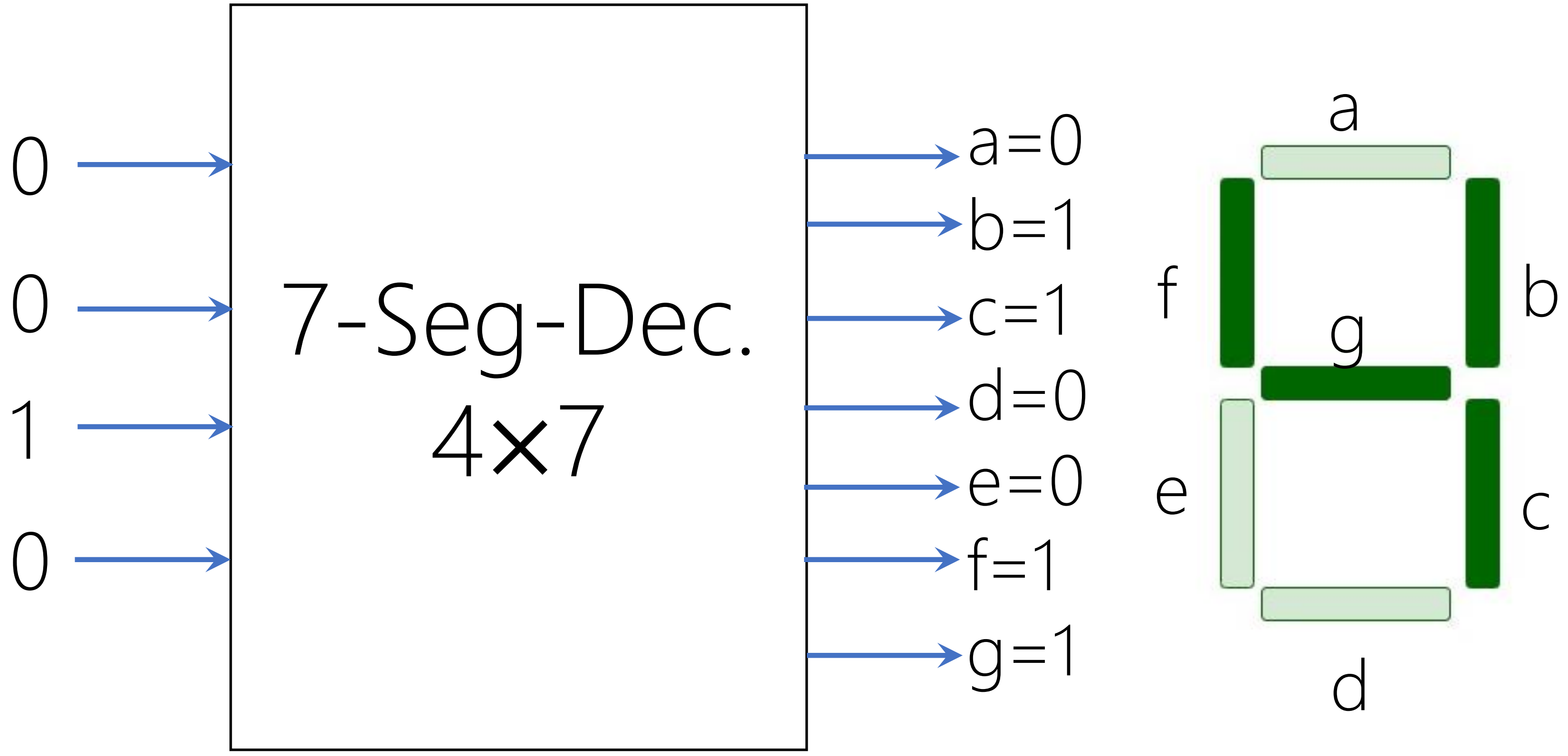
Binary Number
BCD
Excess-3
Aiken
Gray
...



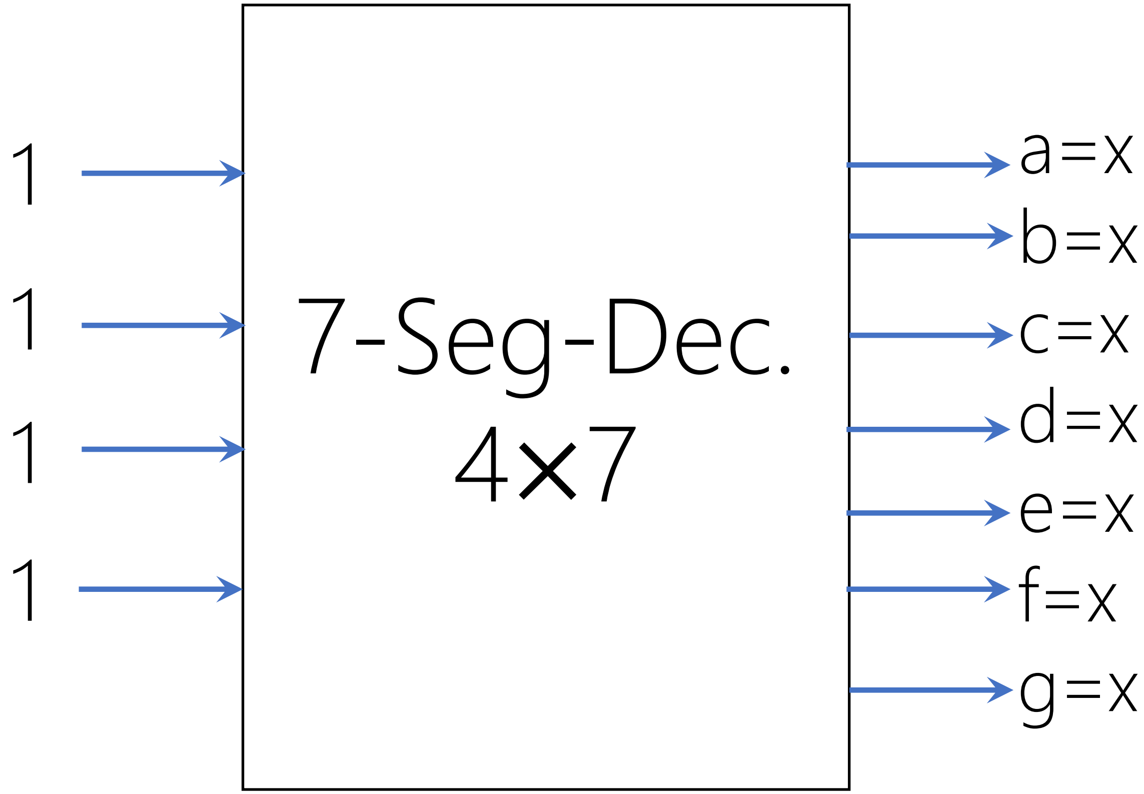
Binary Number



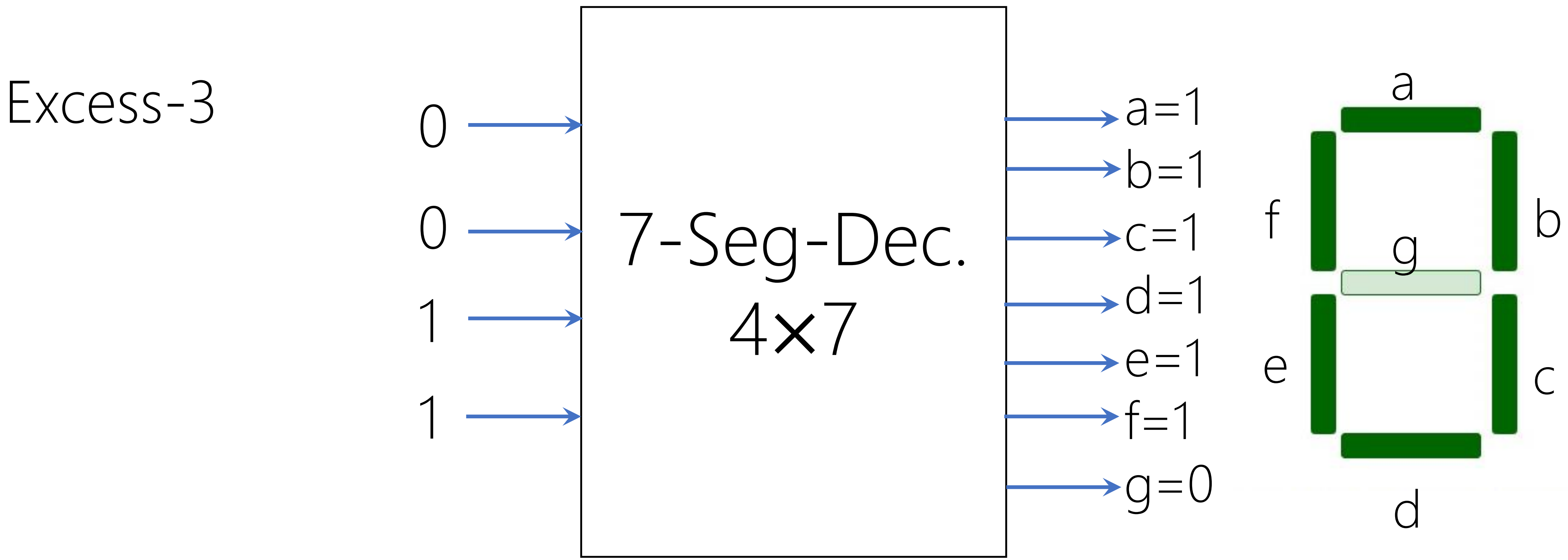
Binary Number



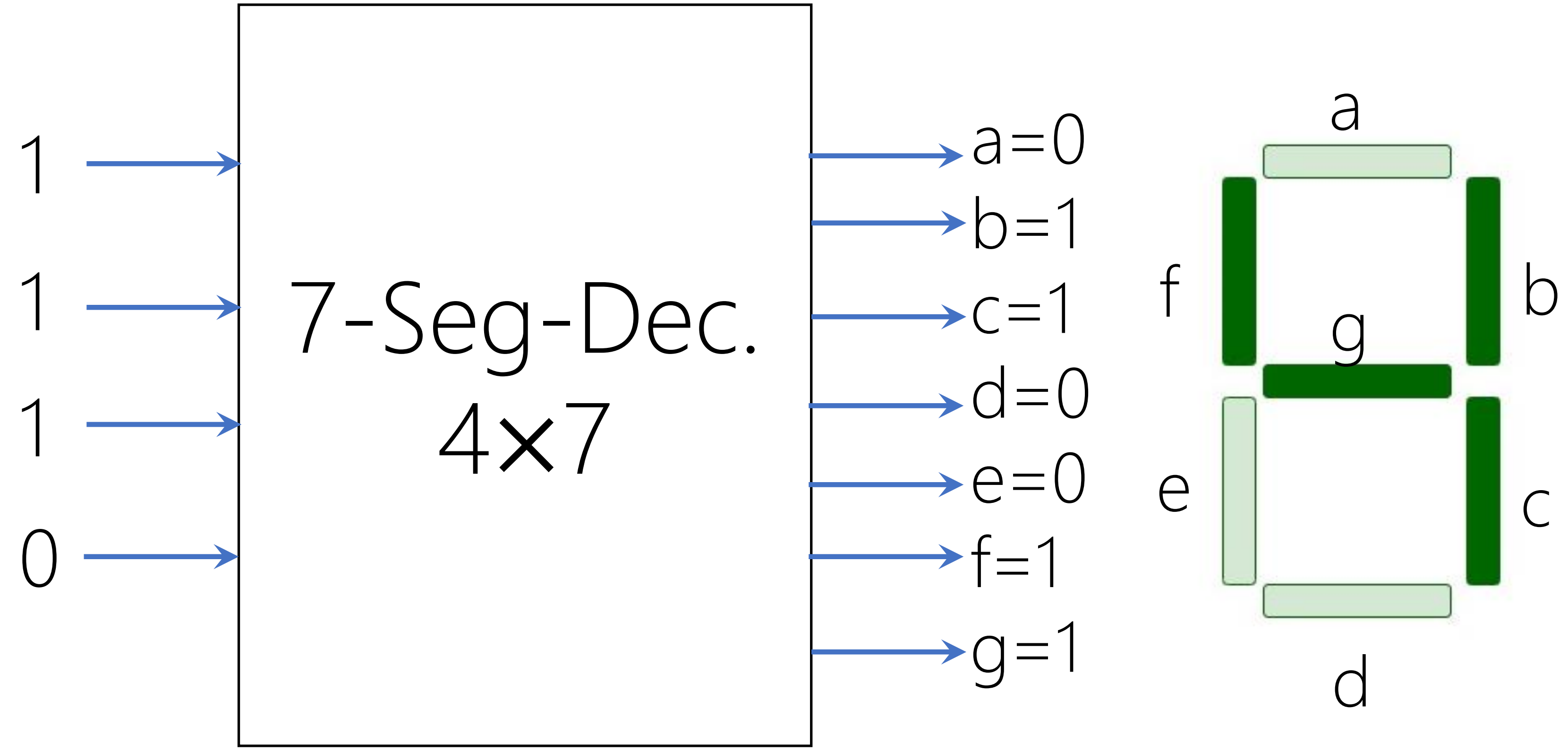
Binary Number



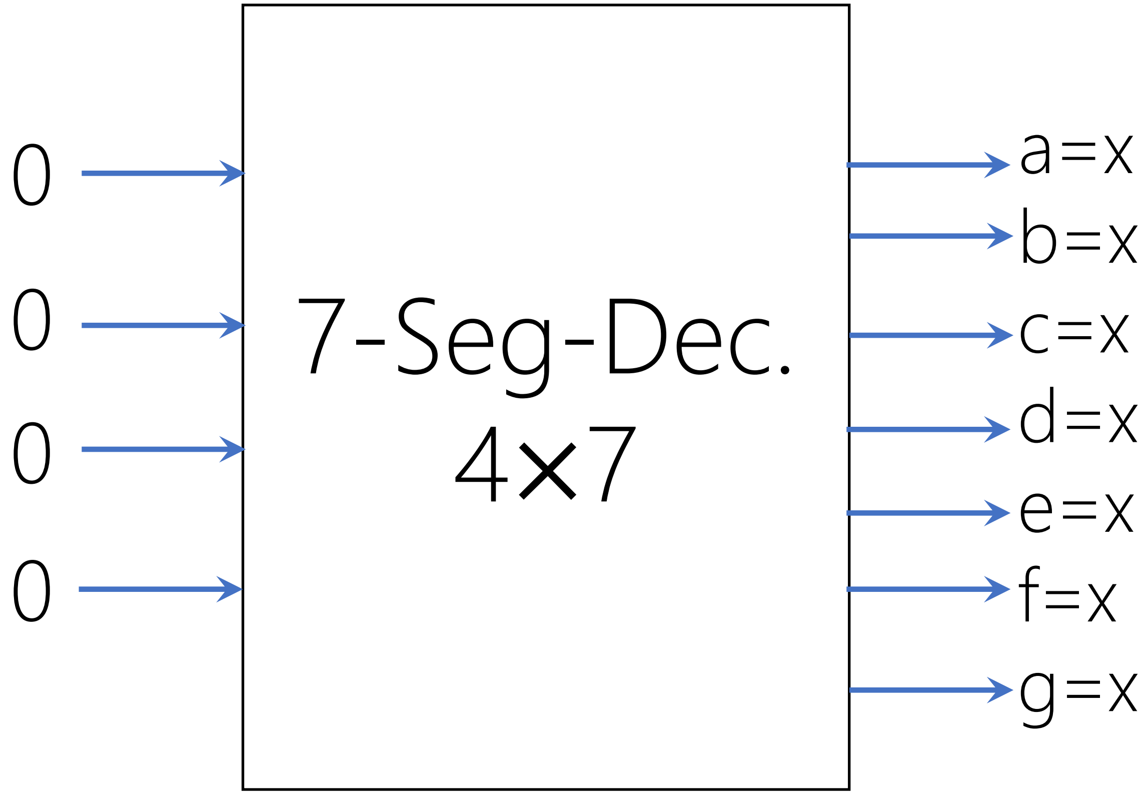
From 10 to 15, don’t care conditions!


Excess-3
0
0
1
1
7-Seg-Dec.
4×7
a=1
b=1
c=1
d=1
e=1
f=1
g=0
a
b
c
d
e
f
g

Excess-3



Excess-3



0,1,2,13,14,15, don’t care conditions!

Combinational Logic

Binary Code Arithmetic

Combinational Logic

BCD Adder

Book: Page 144-146