5 Binary Codes and Combinational Circuit

Binary Codes

Digital communication prefer to use binary codes to represent information such as numbers and alphabets:

- Number: Binary Coded Decimal (BCD), Grey Code etc.
- Alphabets, Symbols and Basic Commands: ASCII and EBCDIC.

Binary Codes

Numbers:

- Human being daily use: Decimal.
- Machine, Equipment Computer use: Binary.
- Then Human being prefer to use CODES to communicate with the Machine, and vice versa.
- There comes **the BCD (Binary Coded Decimal).**

Binary Coded Decimal

- BCD is the binary code to represent a decimal.
- Requires the Binary code for 10 numbers: 0 - 9.
- Therefore, it must be a 4-bit code for 1 Digit, i.e.

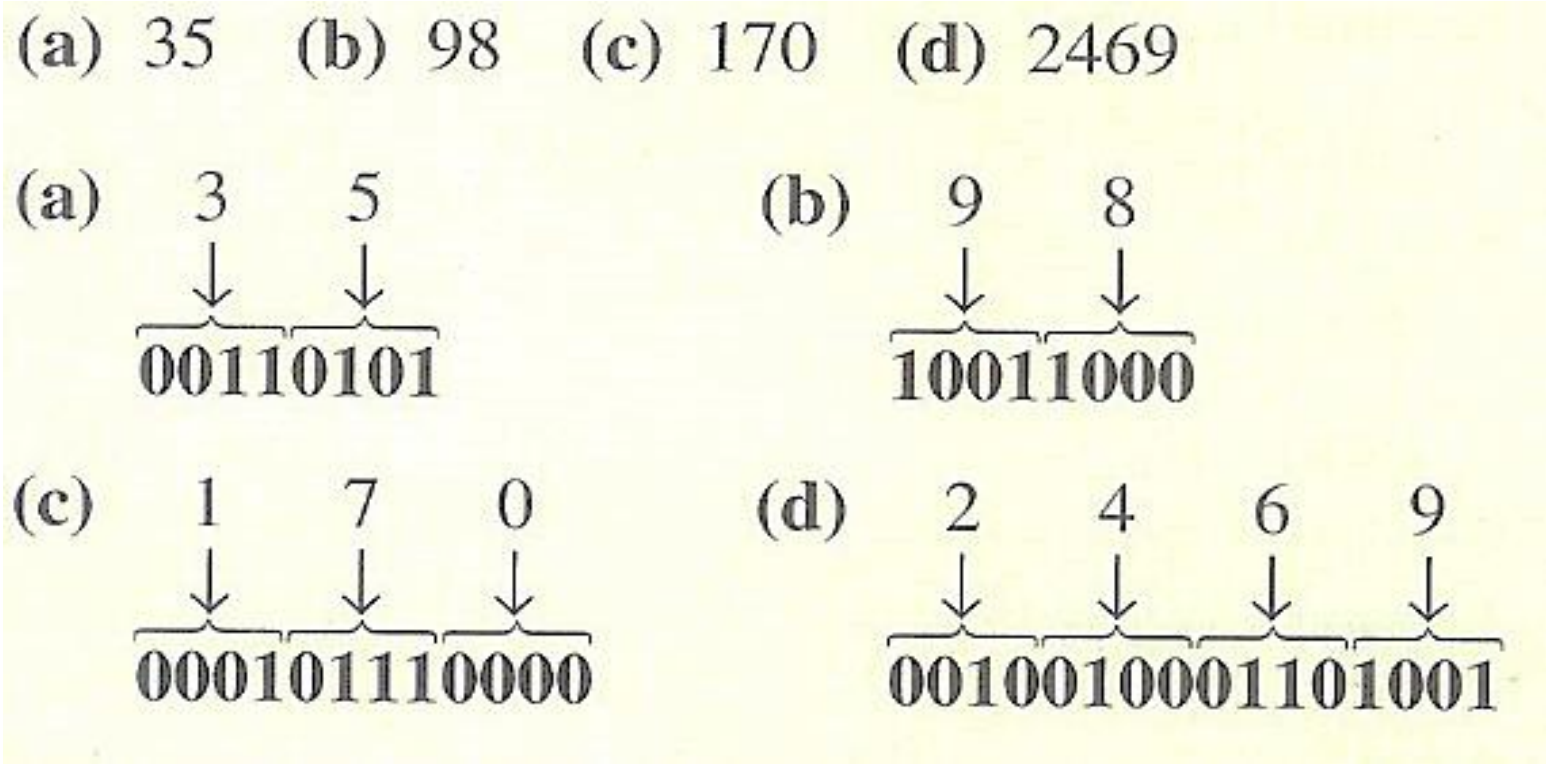
DECIMAL DIGIT	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

- Sometimes it is called 8421-BCD
- For multiple digits, the code must be multiple of 4-bit binary groups too.
- For example: A Decimal Number: 1 3 5

  has BCD : 0001 0011 0101

8421 Binary Coded Decimal

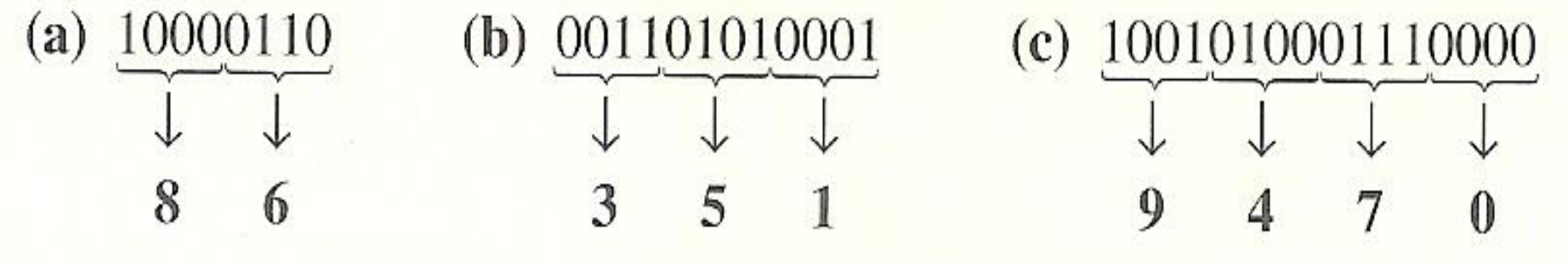
Example: Change the following decimals to BCD.



8421 Binary Coded Decimal

Example: Change the following BCDs to decimals:

(a) 10000110 (b) 001101010001 (c) 1001010001110000



8421 Binary Coded Decimal

- Since the BCD is the 4-bit binary code, which represent 16 digits.
- But it employs only 10 numbers: 0-9.
- The rest of 6 numbers (A-F) are ignored,

  and they would be considered as Don’t Care in the truth table.

8421 BCD: Addition

BCD Addition Method:

Add both binary numbers together bit by bit:

If it is Not more than 9 (1001): The result is feasible;

If it is more than 9 (1001): The result is **NOT** feasible,

need to be corrected by:

- A carry must be produced;
- The result has to be changed to the decimal digit

  by offsetting of 6 (0110)

8421 BCD: Addition

Example: Add the following BCDs:

(a) 0011 + 0100

(b) 00100011 + 00010101

(c) 10000110 + 00010011

(d) 010001010000 + 010000010111

The decimal number additions are shown for comparison.

(a)

	0011	3
	+ 0100	+ 4
	0111	7

(b)

	0010	0011	23
	+ 0001	0101	+ 15
	0011	**1000**	38

(c)

	1000	0110	86
	+ 0001	0011	+ 13
	1001	**1001**	99

(d)

	0100	0101	0000	450
	+ 0100	0001	0111	+ 417
	1000	**0110**	**0111**	867

8421 BCD: Addition

Example: Add the following BCDs:

(a) 1001 + 0100 (b) 1001 + 1001

(c) 00010110 + 00010101 (d) 01100111 + 01010011

The decimal number additions are shown for comparison.

(a)

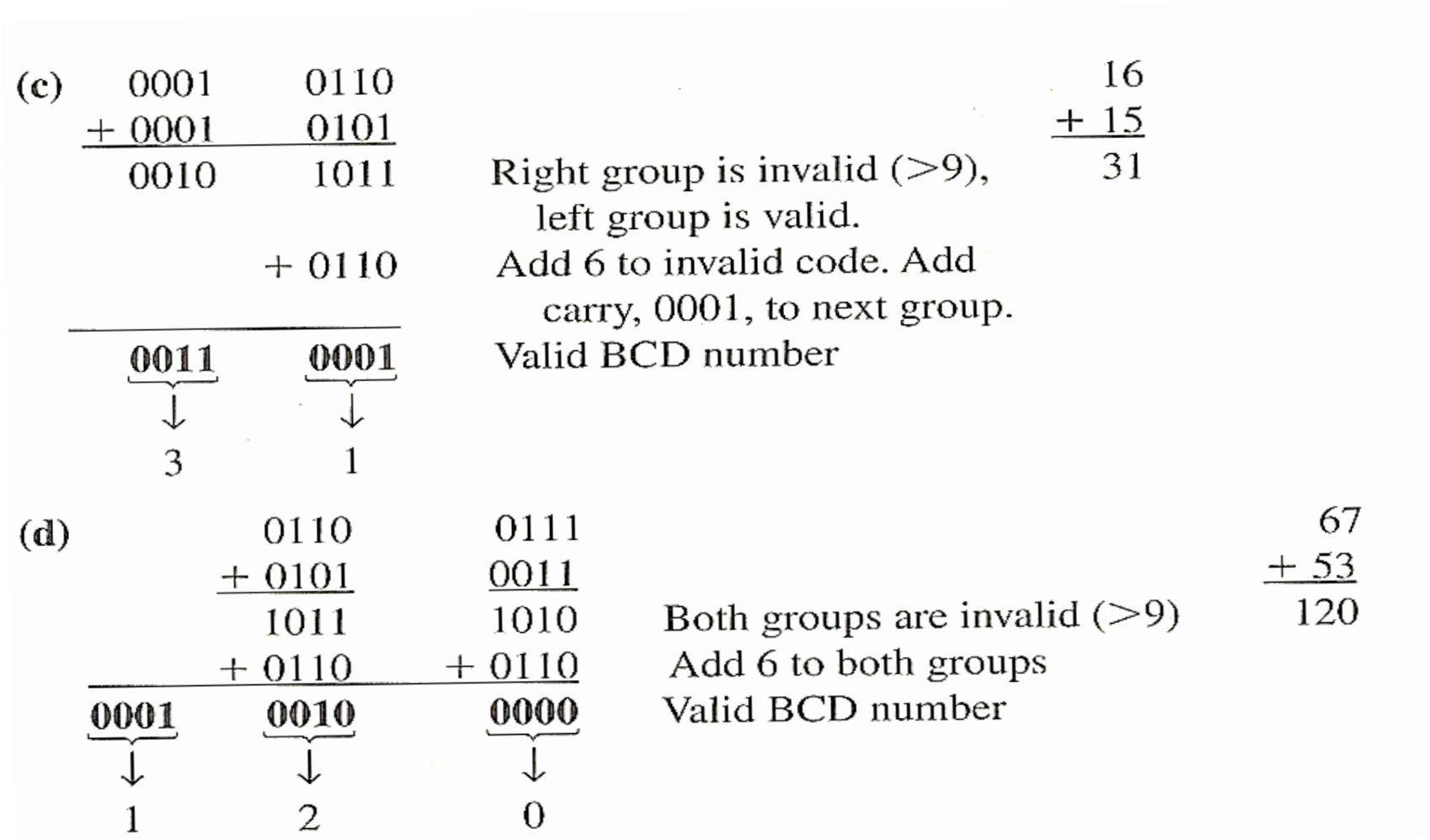
```
   1001                                        9
 + 0100                                      +4
   1101   Invalid BCD number (>9)            13
 + 0110   Add 6
---------
 0001 0011  Valid BCD number
   ↓    ↓
   1    3
```

(b)

```
   1001                                        9
 + 1001                                      + 9
1  0010   Invalid because of carry            18
 + 0110   Add 6
---------
 0001 1000  Valid BCD number
   ↓    ↓
   1    8
```

8421 BCD: Addition

Example: Add the following BCDs:



American Standard Coded for Information Interchange (ASCII)

ASCII is the 7-bit standard code used for representing information:

- Write in hexadecimal from 00 – 7F
- Represents information upto 2^7 = 128 codes;

- First 32 (00-1F) for Basic Commands: “null”, “line feed”, “start of text”, “carriage return” etc.
- Hex (30-39) for Numbers: 0-9;
- Hex (41-5A) for Capital Letters: A - Z;
- Hex (61-7A) for Small Letters: a - z;
- The rest for Symbols: +, - , * , / , ?, etc.

American Standard Code for Information Interchange (ASCII).

CONTROL CHARACTERS				GRAPHIC SYMBOLS											
NAME	DEC	BINARY	HEX	SYMBOL	DEC	BINARY	HEX	SYMBOL	DEC	BINARY	HEX	SYMBOL	DEC	BINARY	HEX
NUL	0	0000000	00	space	32	0100000	20	@	64	1000000	40	`	96	1100000	60
SOH	1	0000001	01	!	33	0100001	21	A	65	1000001	41	a	97	1100001	61
STX	2	0000010	02	"	34	0100010	22	B	66	1000010	42	b	98	1100010	62
ETX	3	0000011	03	#	35	0100011	23	C	67	1000011	43	c	99	1100011	63
EOT	4	0000100	04	$	36	0100100	24	D	68	1000100	44	d	100	1100100	64
ENQ	5	0000101	05	%	37	0100101	25	E	69	1000101	45	e	101	1100101	65
ACK	6	0000110	06	&	38	0100110	26	F	70	1000110	46	f	102	1100110	66
BEL	7	0000111	07	'	39	0100111	27	G	71	1000111	47	g	103	1100111	67
BS	8	0001000	08	(	40	0101000	28	H	72	1001000	48	h	104	1101000	68
HT	9	0001001	09	)	41	0101001	29	I	73	1001001	49	i	105	1101001	69
LF	10	0001010	0A	*	42	0101010	2A	J	74	1001010	4A	j	106	1101010	6A
VT	11	0001011	0B	+	43	0101011	2B	K	75	1001011	4B	k	107	1101011	6B
FF	12	0001100	0C	,	44	0101100	2C	L	76	1001100	4C	l	108	1101100	6C
CR	13	0001101	0D	–	45	0101101	2D	M	77	1001101	4D	m	109	1101101	6D
SO	14	0001110	0E	.	46	0101110	2E	N	78	1001110	4E	n	110	1101110	6E
SI	15	0001111	0F	/	47	0101111	2F	O	79	1001111	4F	o	111	1101111	6F
DLE	16	0010000	10	0	48	0110000	30	P	80	1010000	50	p	112	1110000	70
DC1	17	0010001	11	1	49	0110001	31	Q	81	1010001	51	q	113	1110001	71
DC2	18	0010010	12	2	50	0110010	32	R	82	1010010	52	r	114	1110010	72
DC3	19	0010011	13	3	51	0110011	33	S	83	1010011	53	s	115	1110011	73
DC4	20	0010100	14	4	52	0110100	34	T	84	1010100	54	t	116	1110100	74
NAK	21	0010101	15	5	53	0110101	35	U	85	1010101	55	u	117	1110101	75
SYN	22	0010110	16	6	54	0110110	36	V	86	1010110	56	v	118	1110110	76
ETB	23	0010111	17	7	55	0110111	37	W	87	1010111	57	w	119	1110111	77
CAN	24	0011000	18	8	56	0111000	38	X	88	1011000	58	x	120	1111000	78
EM	25	0011001	19	9	57	0111001	39	Y	89	1011001	59	y	121	1111001	79
SUB	26	0011010	1A	:	58	0111010	3A	Z	90	1011010	5A	z	122	1111010	7A
ESC	27	0011011	1B	;	59	0111011	3B	[	91	1011011	5B	{	123	1111011	7B
FS	28	0011100	1C	<	60	0111100	3C	\	92	1011100	5C	|	124	1111100	7C
GS	29	0011101	1D	=	61	0111101	3D	]	93	1011101	5D	}	125	1111101	7D
RS	30	0011110	1E	>	62	0111110	3E	^	94	1011110	5E	~	126	1111110	7E
US	31	0011111	1F	?	63	0111111	3F	_	95	1011111	5F	Del	127	1111111	7F

Parity Bit

- The error that occurred in sending binary data due to

  - Interference

  - The small signal.

- To identify the occurrence of an error, achieved by adding a special bit to the data set by counting the number of ‘1’ in the data set. We call this special bit as the parity bit.

Parity Bit

Even Parity

(To make the number of ‘1’ in the data sending to be **Even**)

The bit is 0 if the number of 1 in the data is even, and

The bit is 1 if the number of 1 in the data is odd.

Odd Parity

(To make the number of ‘1’ in the data sending to be **Odd**)

The bit is 0 if the number of 1 in the data is odd, and

The bit is 1 if the number of 1 in the data is even.

Parity Bit

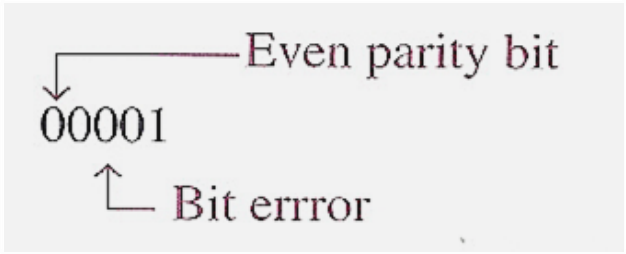
Four bit Message $D_3D_2D_1D_0$	*Even Parity* (P_E)	*Odd Parity* (P_o)
0000	0	1
0001	1	0
0010	1	0
0011	0	1
0100	1	0
0101	0	1
0110	0	1
0111	1	0
1000	1	0
1001	0	1
1010	0	1
1011	1	0
1100	0	1
1101	1	0
1110	1	0
1111	0	1

Parity Bit

Example: Sending 0101 with the protocol of even parity bit:

Sending 0101 →

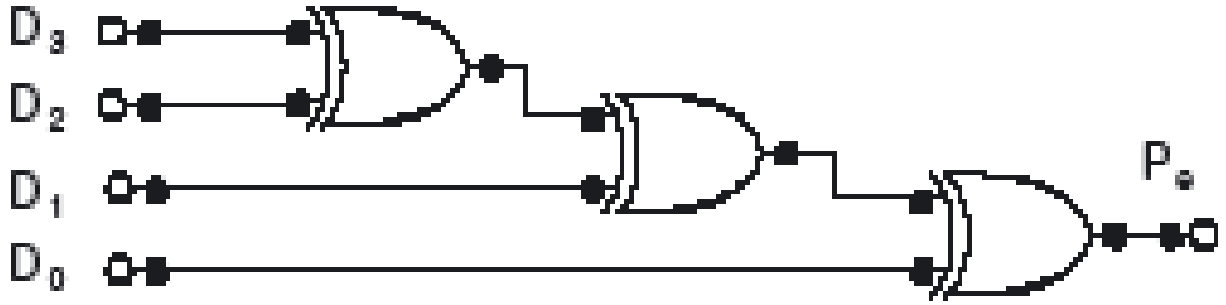
If receive 00001



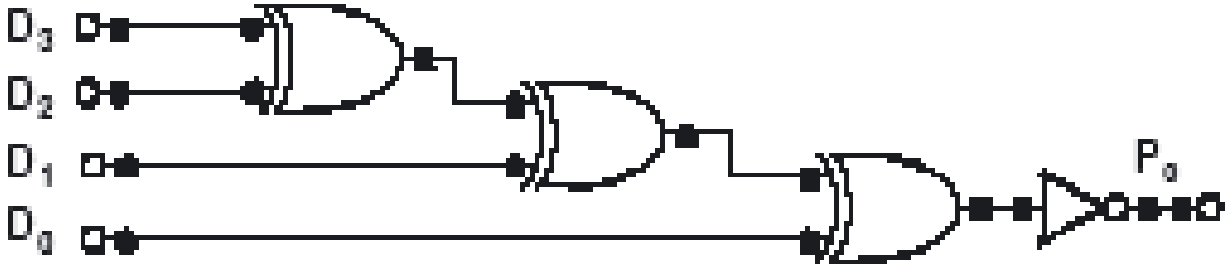
- Use even parity bit, so the number of 1's must be even.
- But there is just a single 1 (odd), so the data sending has error.
- However, we cannot not know which position is error.
- However, we cannot use this data.
- So, we have to request for resending the data.

PARITY GENERATOR/CHECKER

A 9-bit parity generator / checker.



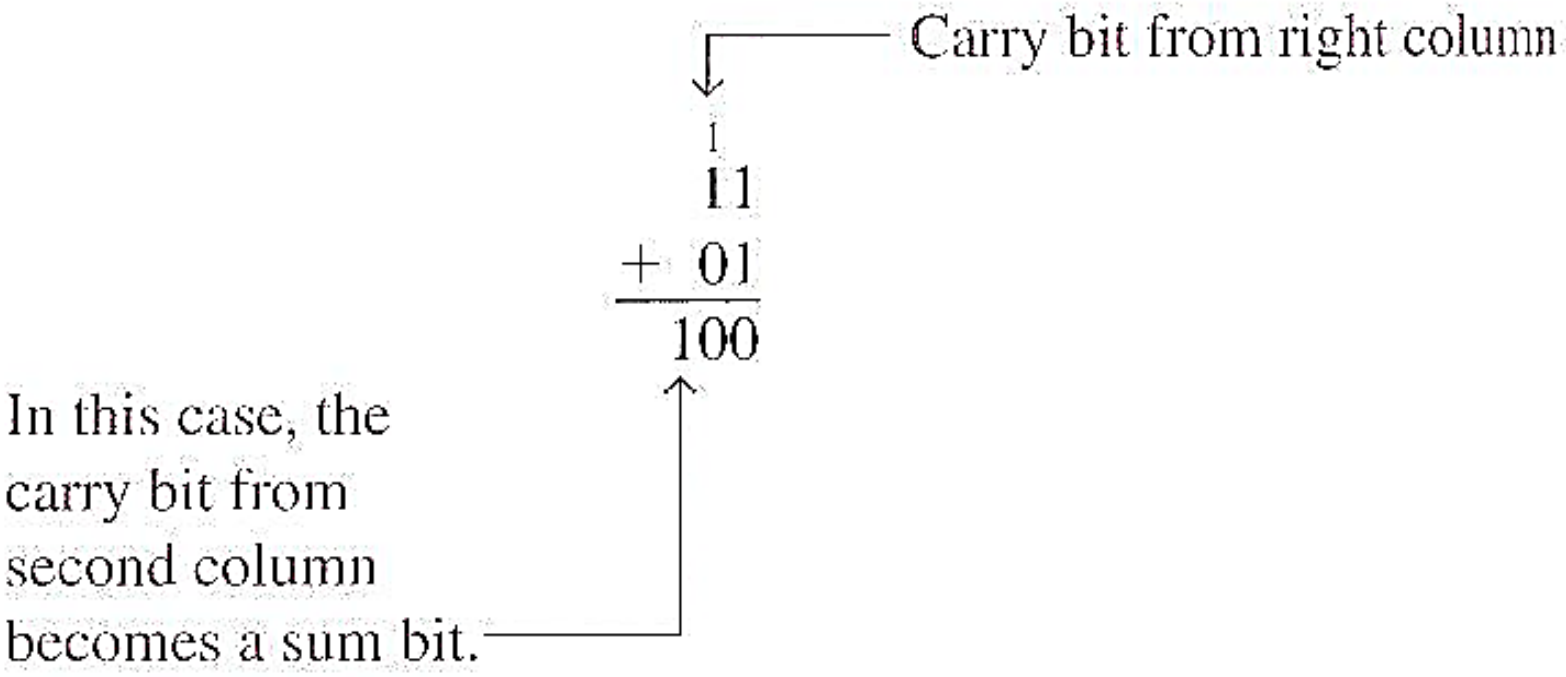
Even parity generator



Odd parity generator

Parallel Binary Adders

- Adding two multiple binary together can be done by using multiple full adders.
- For example: Two bits addition.

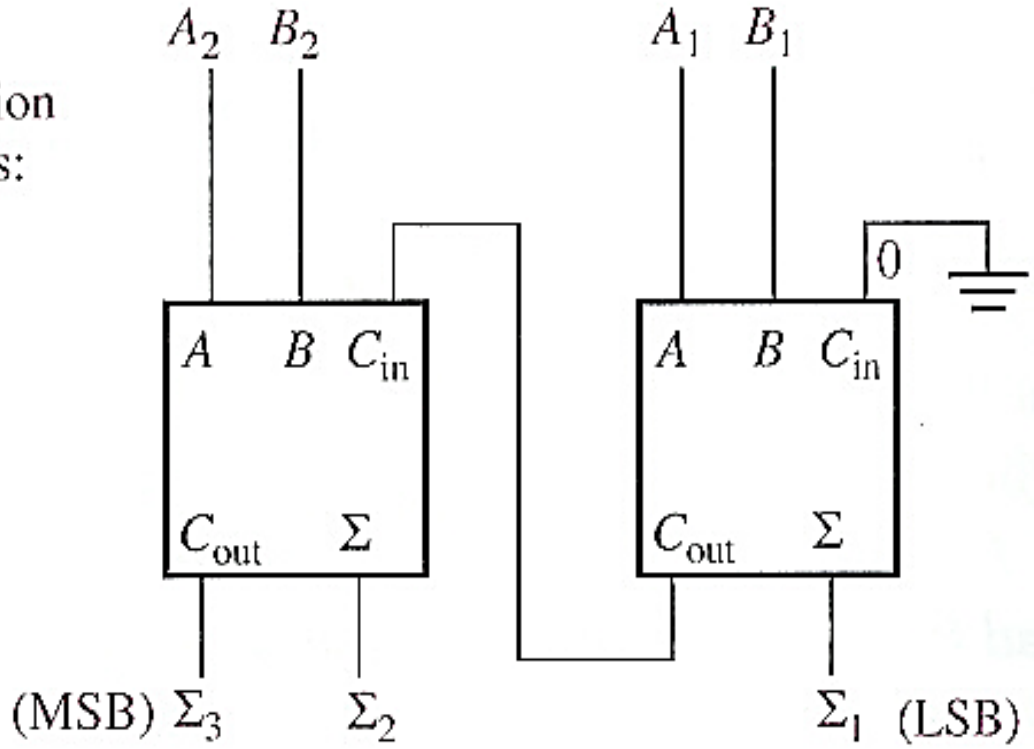


Parallel Binary Adders

Two-bits addition: Use TWO full adders:

General format, addition of two 2-bit numbers:

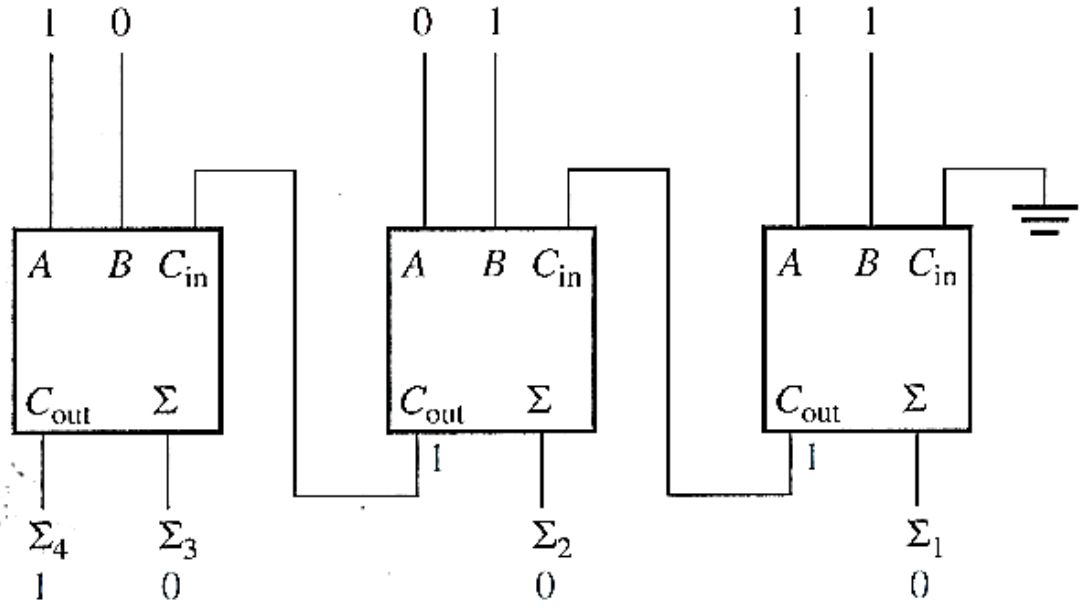
$$\begin{array}{r} A_2A_1 \\ +\ B_2B_1 \\ \hline \Sigma_3\Sigma_2\Sigma_1 \end{array}$$



Parallel Binary Adders: 3 Bits

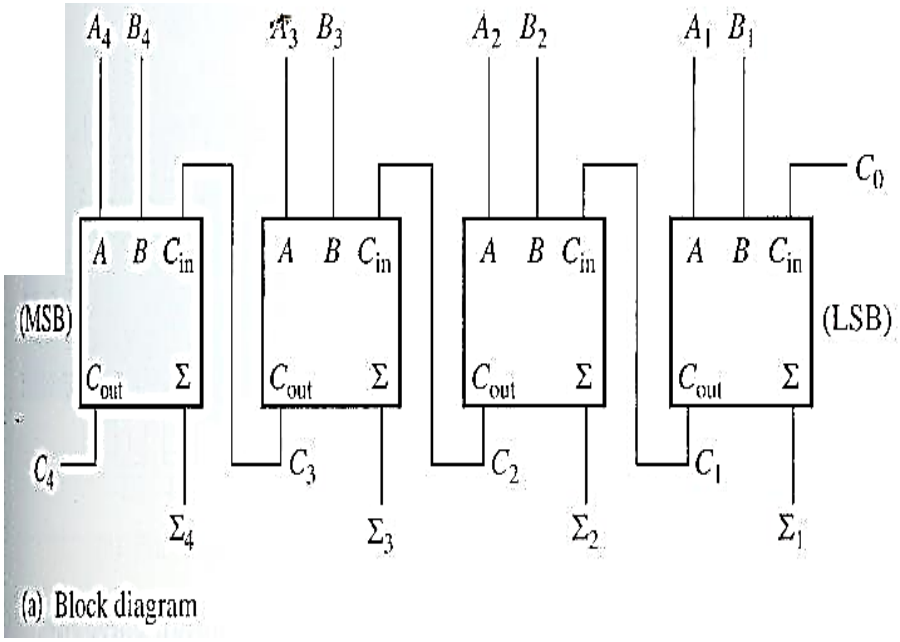
Three-bits addition: Use THREE full adders:

Determine the sum generated by the 3-bit parallel adder in Figure 6–8 and show the intermediate carries when the binary numbers 101 and 011 are being added.

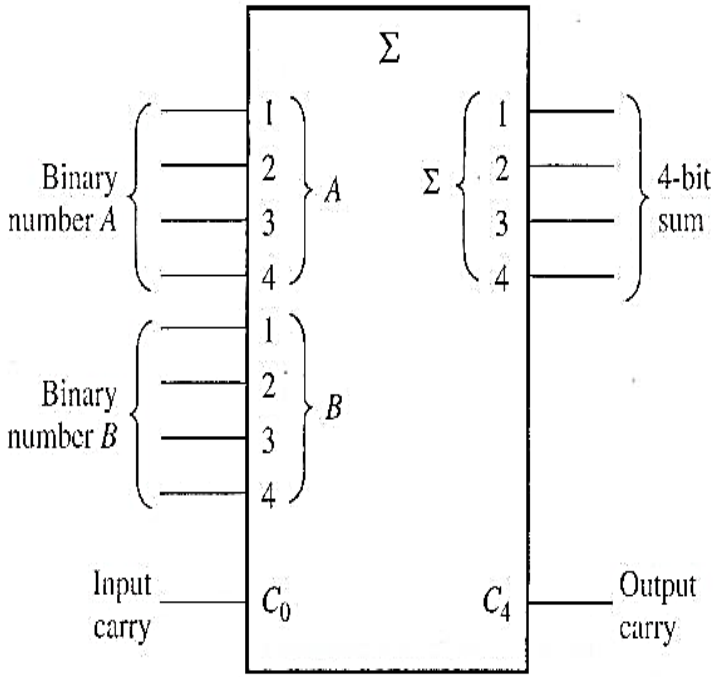


Parallel Binary Adders: 4 Bits

Four-bits addition: Use FOUR full adders ➔ IC:



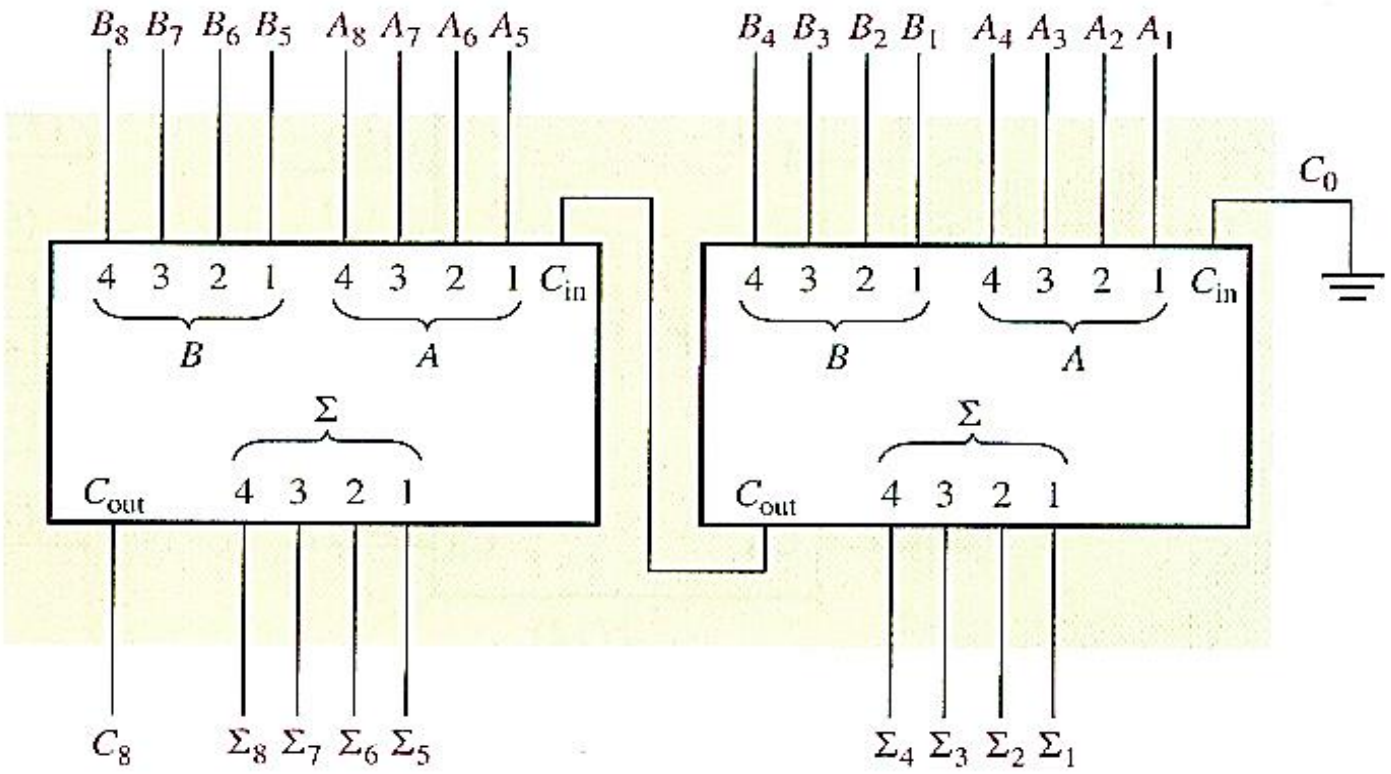
(a) Block diagram



(b) Logic symbol

Parallel Binary Adders: 8 Bits

Cascading of two 4-bit adders to form an 8-bit adder.

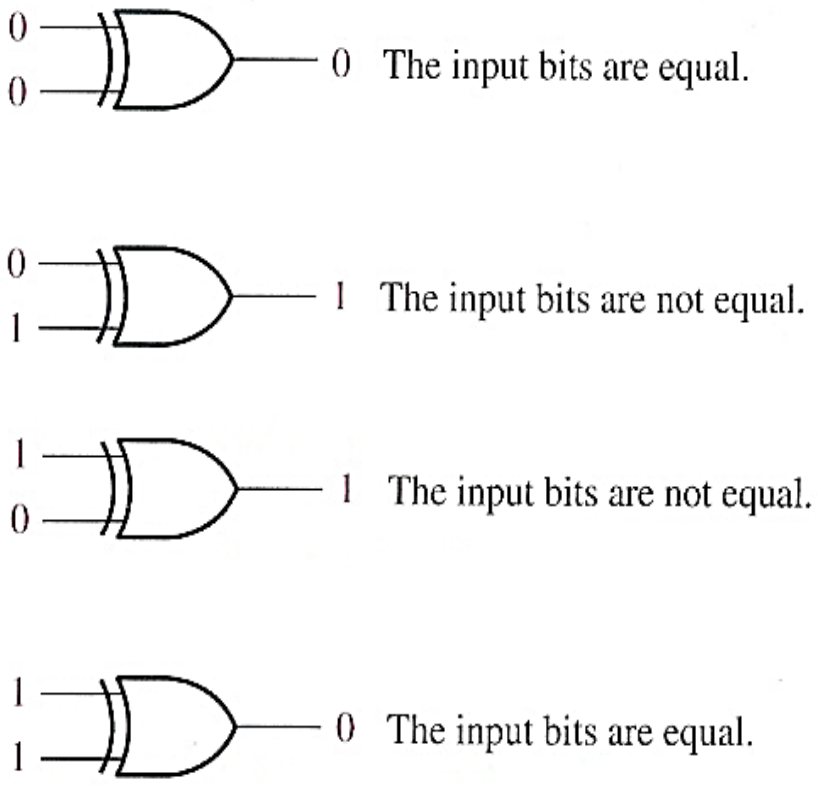


Comparators

The comparison circuit is used to compare two binary numbers if they are the same value or not.

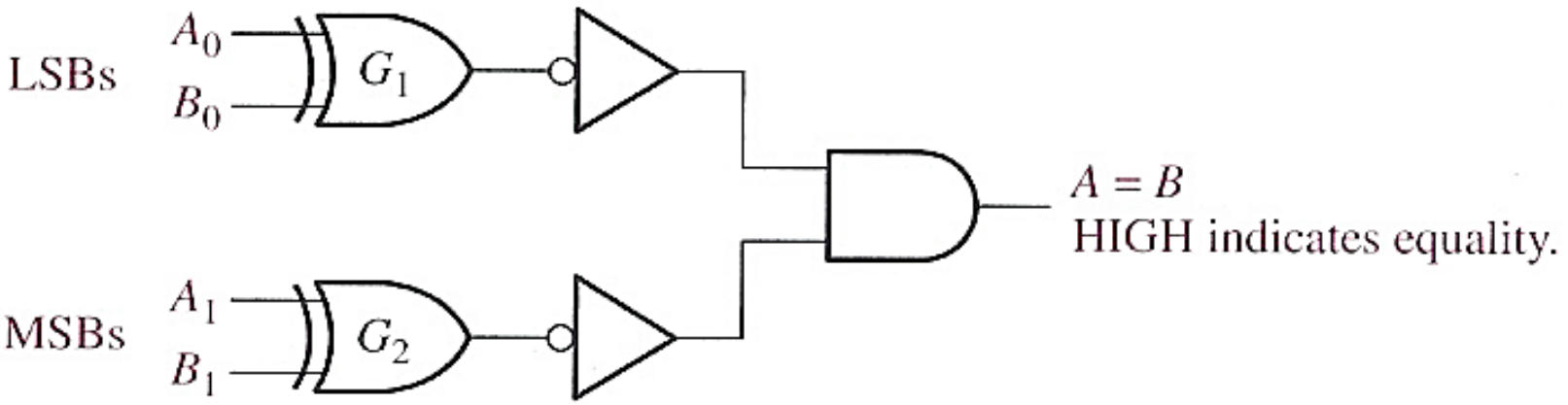
May add a circuit to determine their values which one is more or less.

1-Bit comparison



Comparators

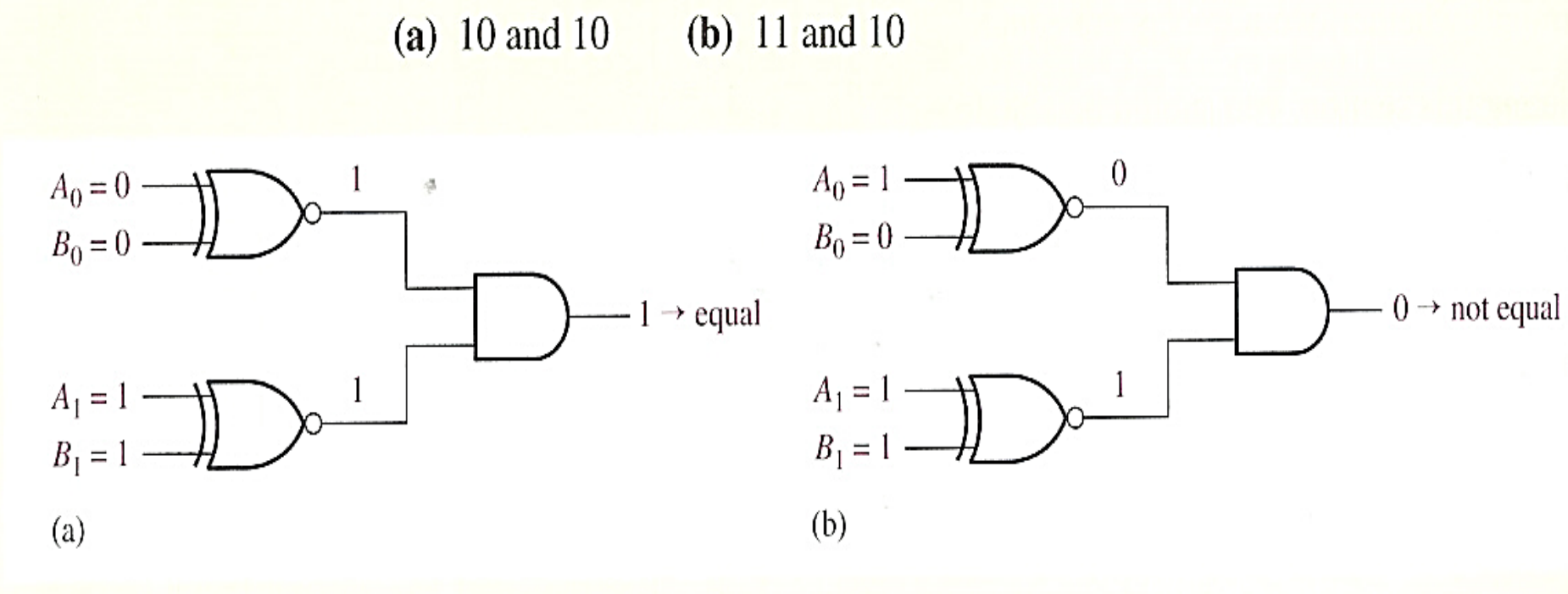
2-Bit comparison



General format: Binary number $A \rightarrow A_1A_0$
Binary number $B \rightarrow B_1B_0$

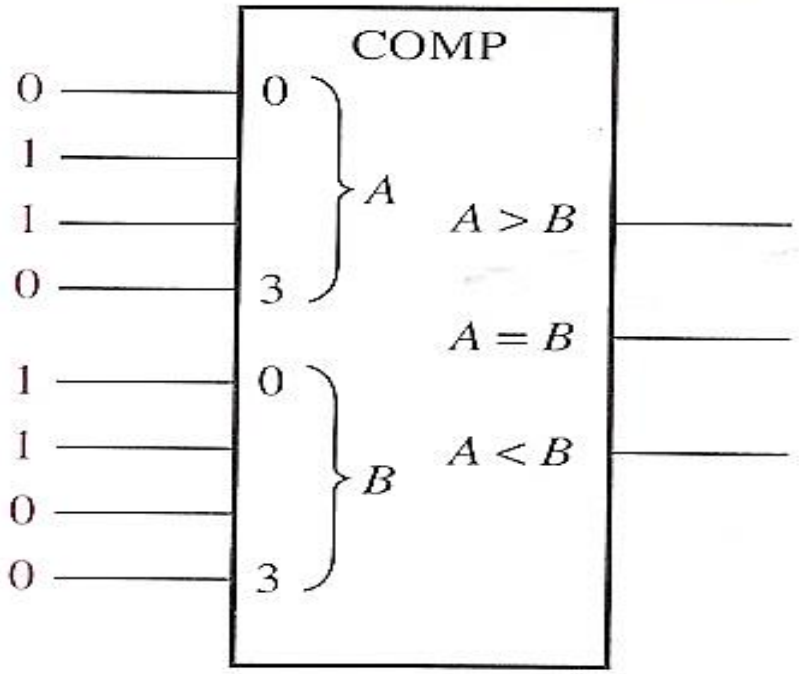
Comparators

2-Bit comparison



Comparators: 4-Bit comparison IC

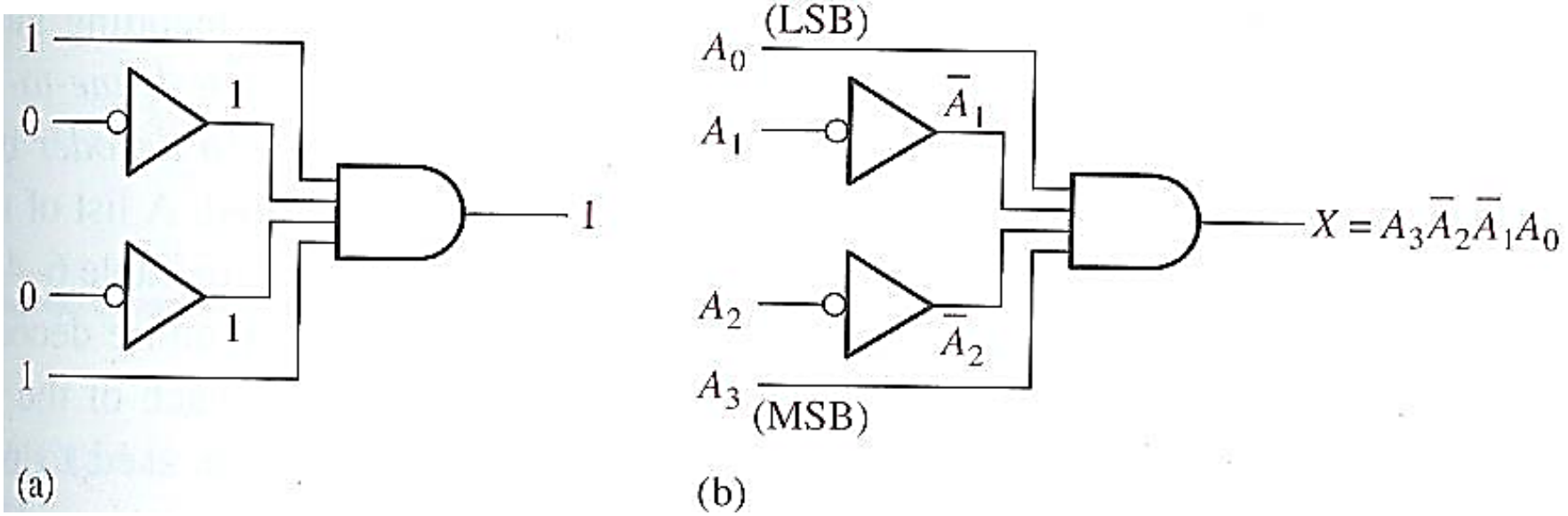
Besides bit-to-bit comparison, should it also indicate which one is greater or smaller.



Decoders

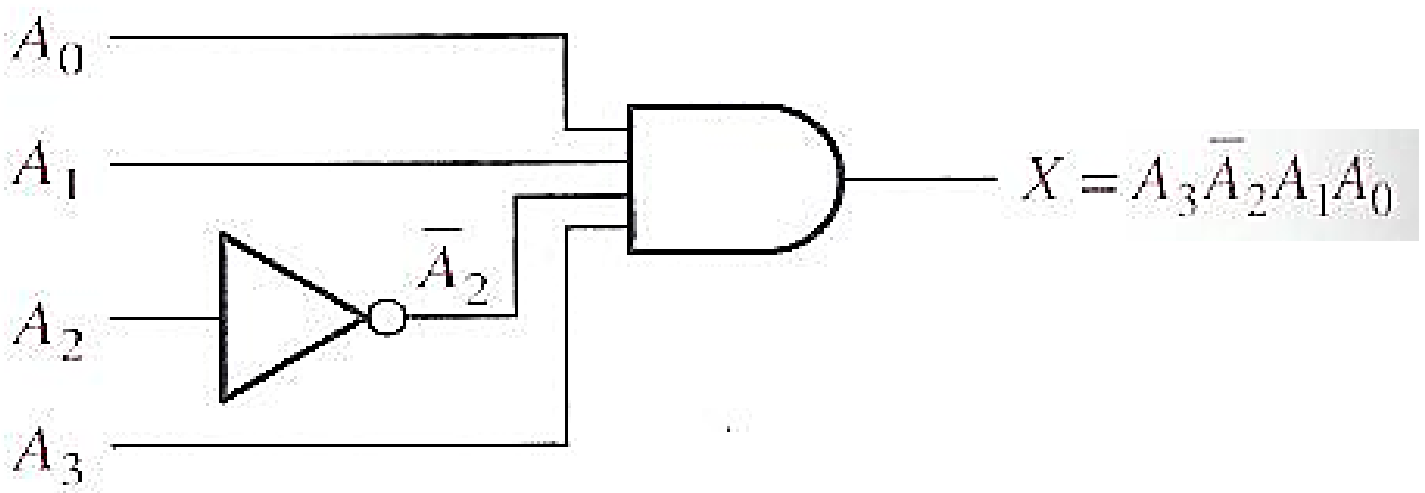
The decoder circuit is a circuit that detects the specified code number (input).

Example: The circuit to detect 1001_2.



Decoders

Example: The circuit to detect 1011_2.



BCD to Decimal Decoders

BCD to Decimal Decoder is the circuit that converts 4-bit BCD to one decimal value.

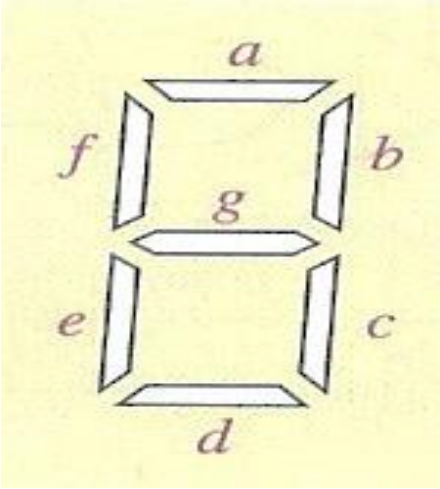
Known as 4-line-to-10-line decoder or 1-of-10-decoder.

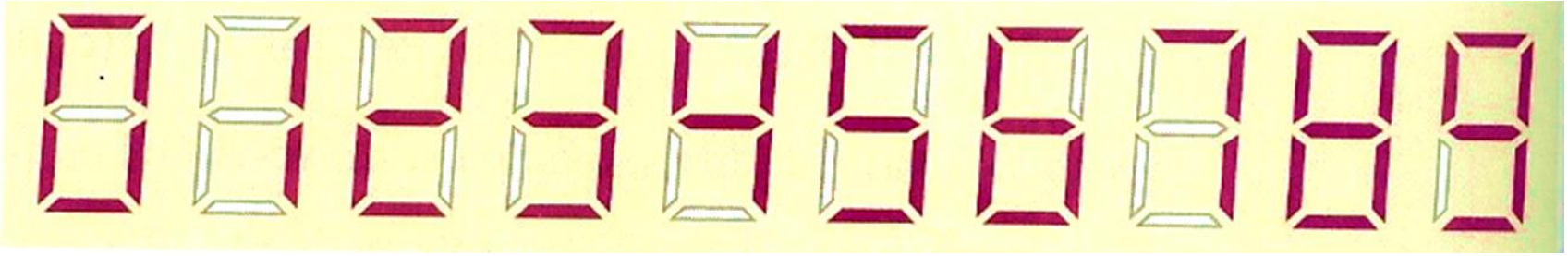
The truth table of the BCD decoder

DECIMAL DIGIT	BCD CODE A_3	A_2	A_1	A_0	DECODING FUNCTION
0	0	0	0	0	$\overline{A}_3\overline{A}_2\overline{A}_1\overline{A}_0$
1	0	0	0	1	$\overline{A}_3\overline{A}_2\overline{A}_1A_0$
2	0	0	1	0	$\overline{A}_3\overline{A}_2A_1\overline{A}_0$
3	0	0	1	1	$\overline{A}_3\overline{A}_2A_1A_0$
4	0	1	0	0	$\overline{A}_3A_2\overline{A}_1\overline{A}_0$
5	0	1	0	1	$\overline{A}_3A_2\overline{A}_1A_0$
6	0	1	1	0	$\overline{A}_3A_2A_1\overline{A}_0$
7	0	1	1	1	$\overline{A}_3A_2A_1A_0$
8	1	0	0	0	$A_3\overline{A}_2\overline{A}_1\overline{A}_0$
9	1	0	0	1	$A_3\overline{A}_2\overline{A}_1A_0$

7 segment Display

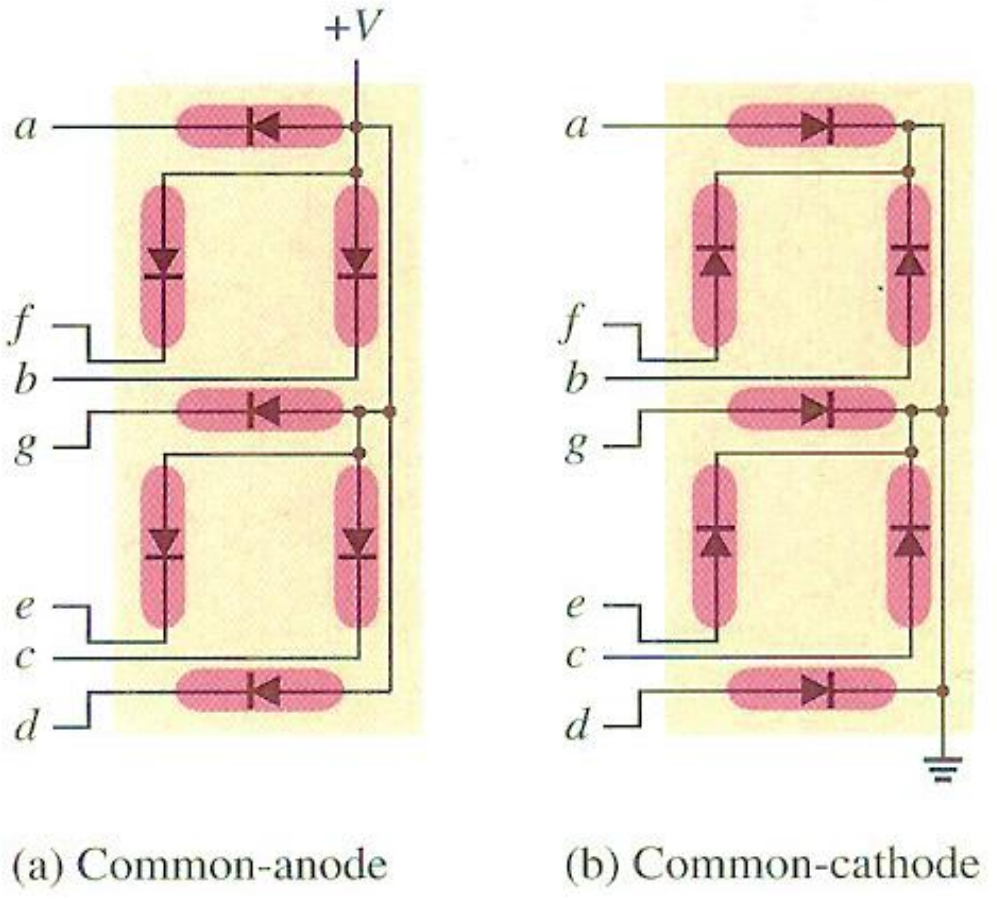
The 7 segment Display is illustrated as figure below:





7 segment Display

Two Configurations of the 7 segment Display :



(a) Common-anode (b) Common-cathode

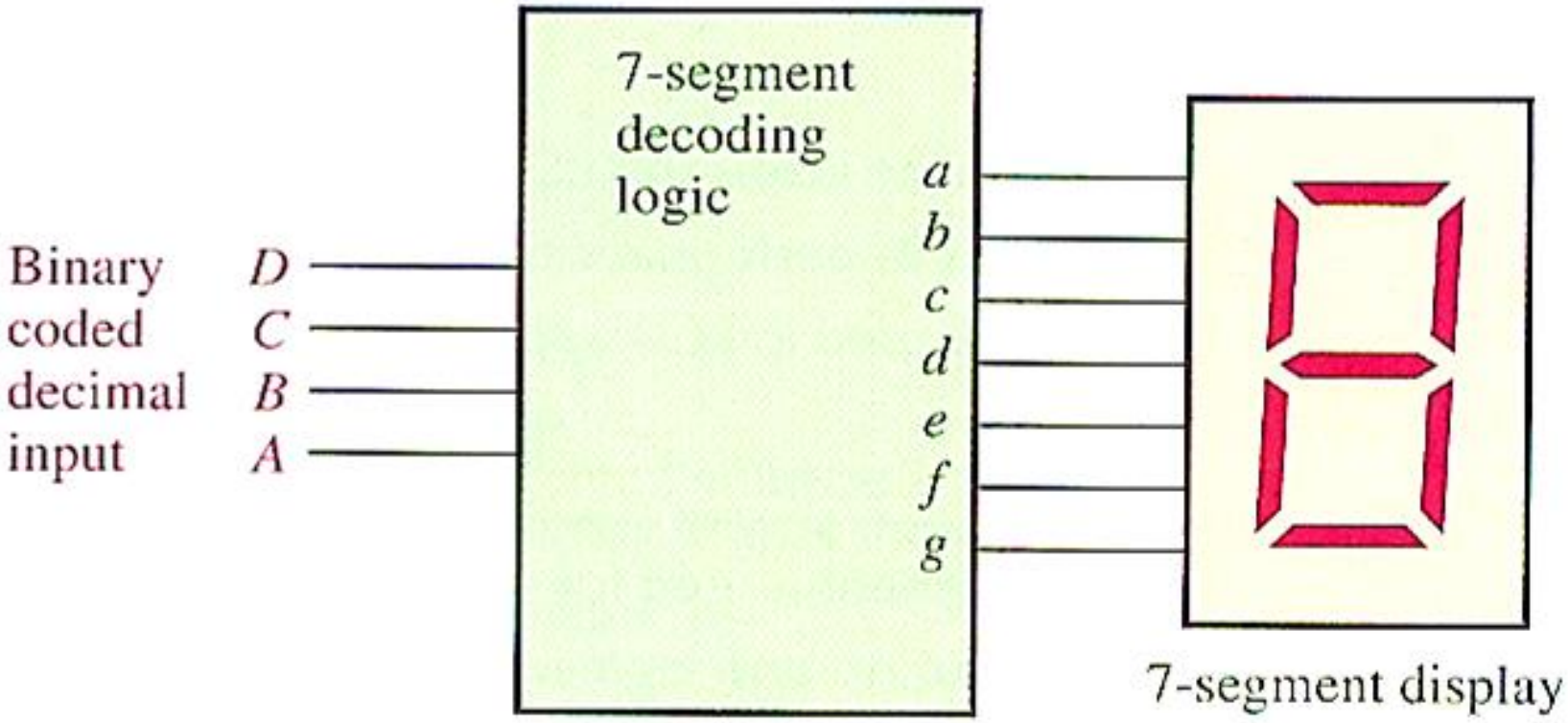
7 segment Display

The driving segments in the 7 segment Display.

DIGIT	SEGMENTS ACTIVATED
0	*a*, *b*, *c*, *d*, *e*, *f*
1	*b*, *c*
2	*a*, *b*, *d*, *e*, *g*
3	*a*, *b*, *c*, *d*, *g*
4	*b*, *c*, *f*, *g*
5	*a*, *c*, *d*, *f*, *g*
6	*a*, *c*, *d*, *e*, *f*, *g*
7	*a*, *b*, *c*
8	*a*, *b*, *c*, *d*, *e*, *f*, *g*
9	*a*, *b*, *c*, *d*, *f*, *g*

7 segment Display

Block diagram of 7-segment display.



7 segment Display

The truth table for driving the 7-segment display.

DECIMAL DIGIT	INPUTS				SEGMENT OUTPUTS						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
10	1	0	1	0	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X

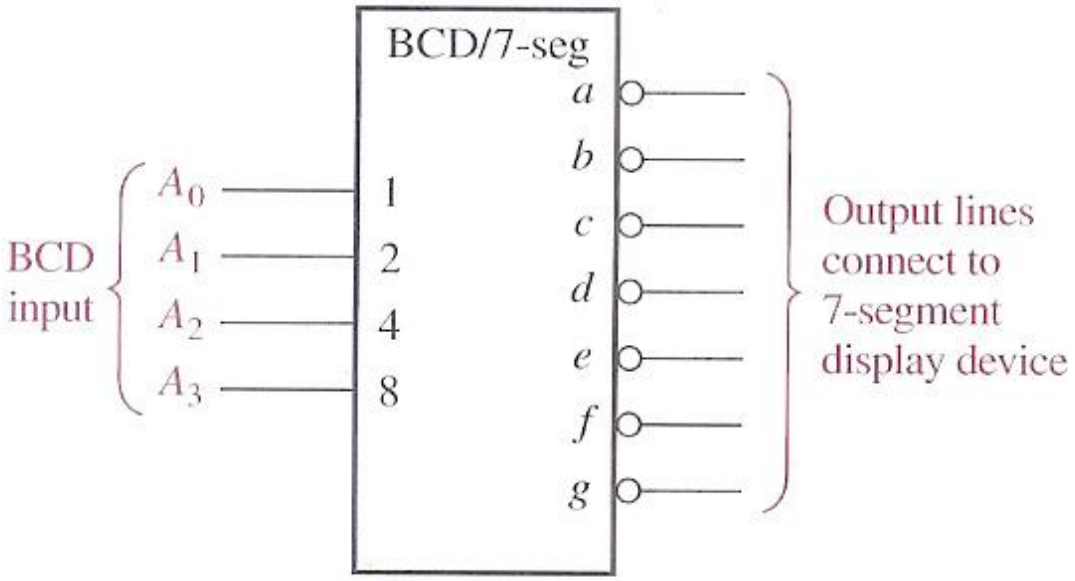
Don’t Care

BCD to 7-segment Decoder

The circuit to decode the BCD input to drive a 7 segment LED.

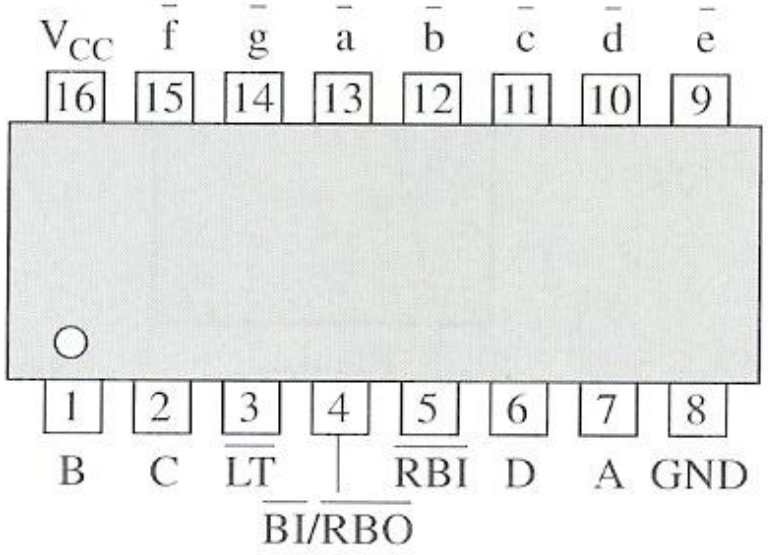
- 4 inputs for BCD.
- 7 outputs for individual LED segments.

Logical diagram of the 7-segment transponder circuit

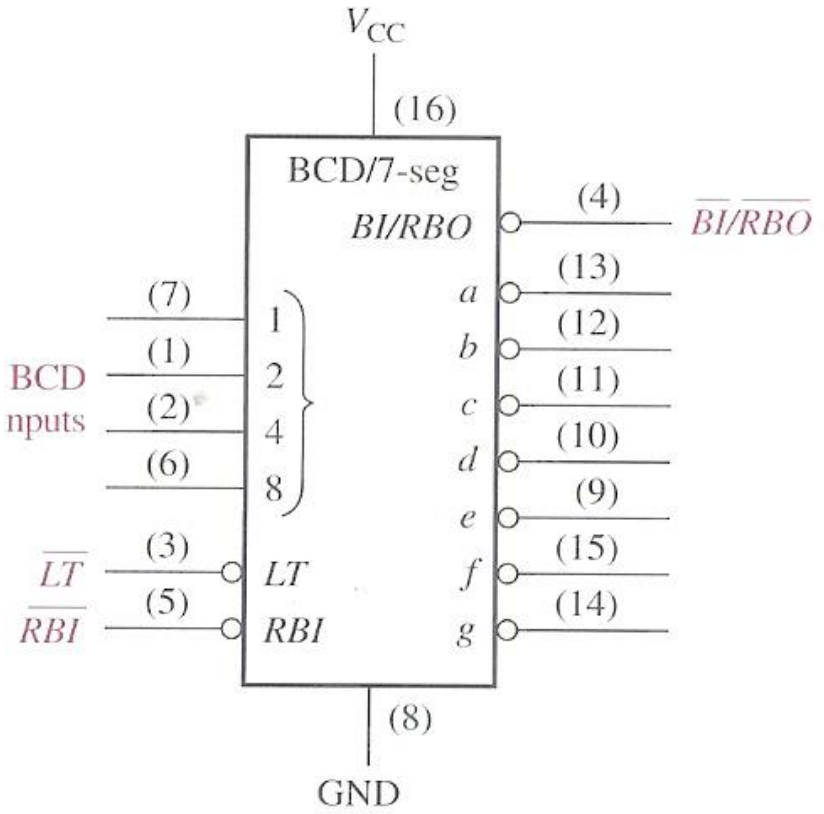


BCD to 7-segment Decoder

The commercial BCD to 7-segment Decoder IC: 74LS47.



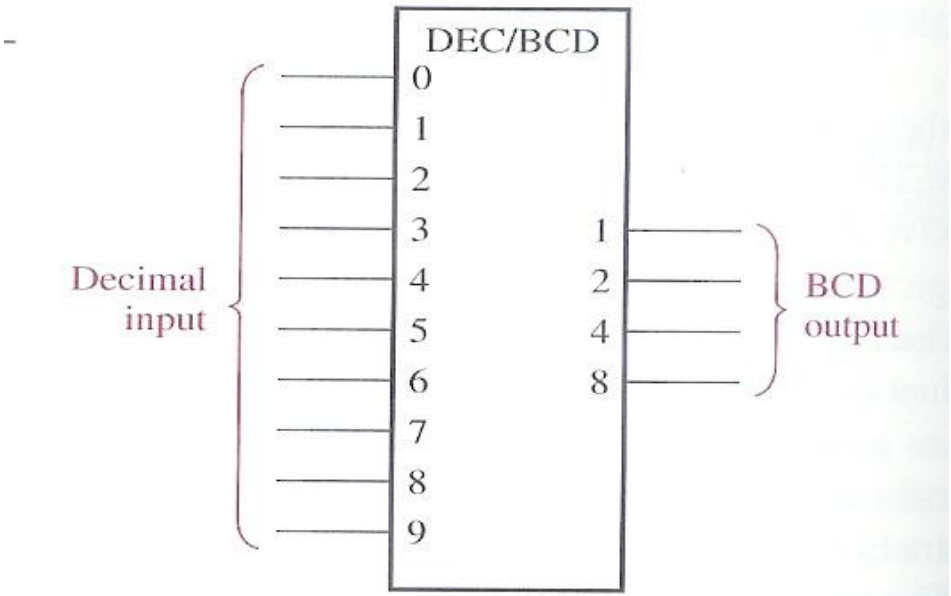
(a) Pin diagram



ENCODER

- The encoder circuit is a logical combination circuit that has reverse processes with decoder.
- Used to represent decimal digits by the binary or BCD numbers.
- The encoder circuit is used for data conversion in data transmission.

Logic symbol for a decimal-to-BCD encoder



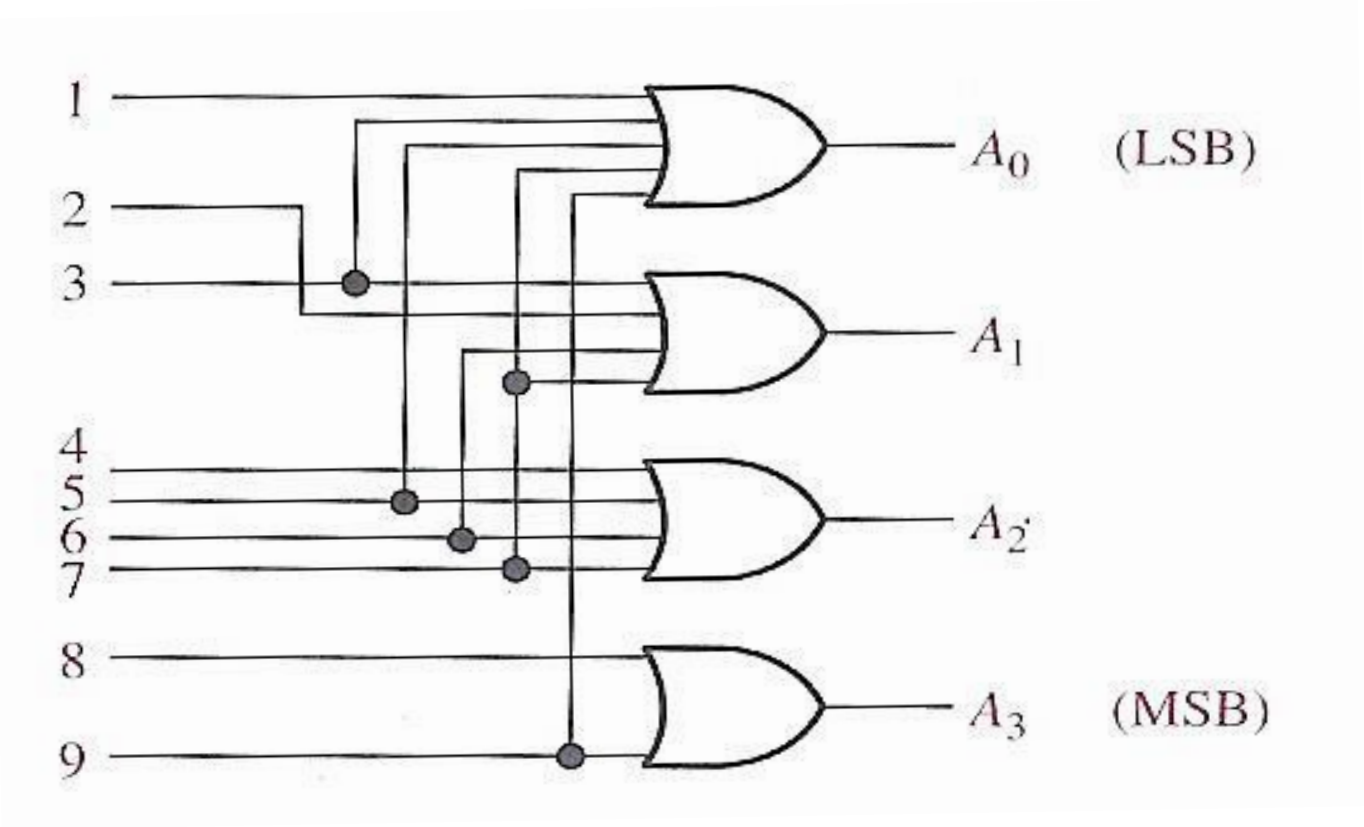
Decimal-to-BCD Encoder

The truth table of the Decimal-to-BCD encoder.

DECIMAL DIGIT	BCD CODE			
	A_3	A_2	A_1	A_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

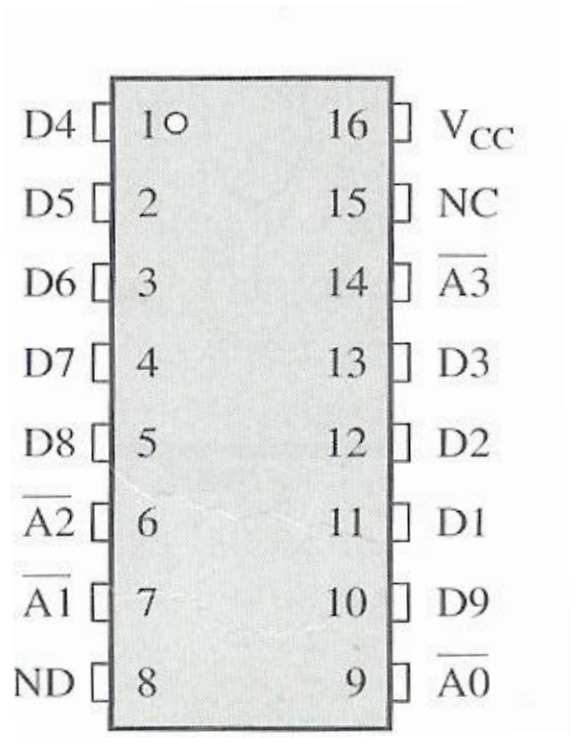
Decimal-to-BCD Encoder

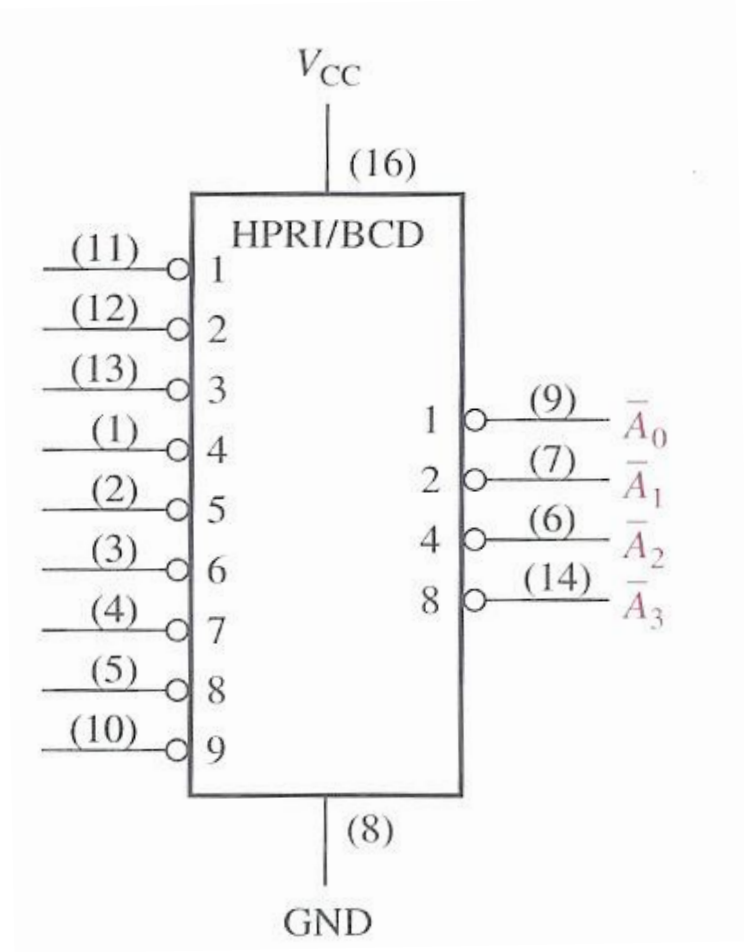
The logical circuit diagram for the Decimal-to-BCD encoder.



Decimal-to-BCD Encoder

The commercial Decimal-to-BCD Encoder IC: 74HC147





BDC-to-Binary Conversion

- A combinational logic circuits to convert from one code to another code.
- BDC-to-Binary Conversion
  - Represent a value, or weight, of each bit in the BCD number by a binary number;
  - Consider adding this weight(s) ONLY the bits that are 1's in the BCD;
  - The result of this addition is the binary equivalent of the BCD number.

BDC-to-Binary Conversion

Table for the BDC-to-Binary Conversion

	BCD BIT	BCD WEIGHT	(MSB) 64	32	16	8	4	2	(LSB) 1
Units	A_0	1	0	0	0	0	0	0	1
	A_1	2	0	0	0	0	0	1	0
	A_2	4	0	0	0	0	1	0	0
	A_3	8	0	0	0	1	0	0	0
Tens	B_0	10	0	0	0	1	0	1	0
	B_1	20	0	0	1	0	1	0	0
	B_2	40	0	1	0	1	0	0	0
	B_3	80	1	0	1	0	0	0	0

	Tens Digit				Units Digit			
Weight:	80	40	20	10	8	4	2	1
Bit designation:	B_3	B_2	B_1	B_0	A_3	A_2	A_1	A_0

BDC-to-Binary Conversion

Example: Convert BDC 87 to the binary number

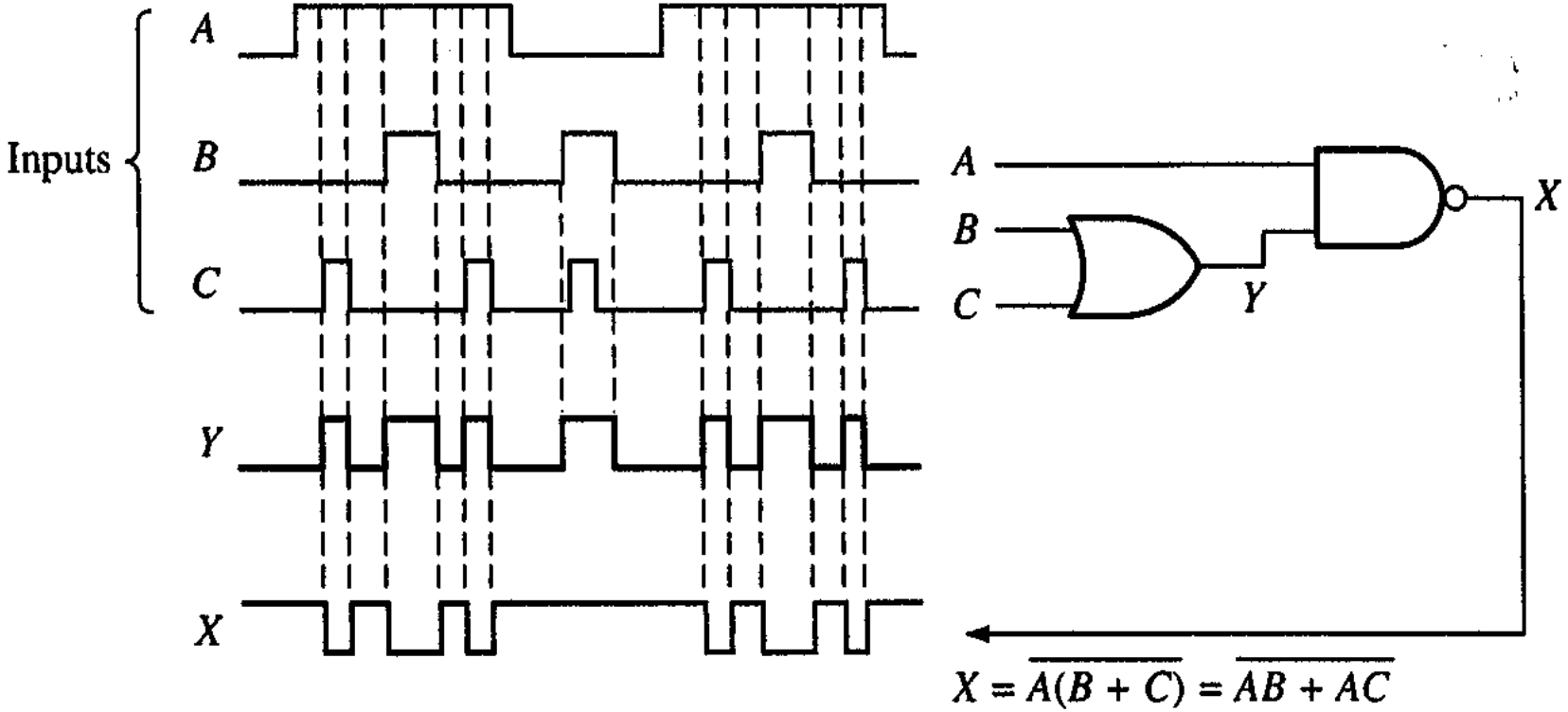
$$\underbrace{1000}_{8}\quad\underbrace{0111}_{7}$$

	Tens Digit				Units Digit			
Weight:	80	40	20	10	8	4	2	1
Bit designation:	B_3	B_2	B_1	B_0	A_3	A_2	A_1	A_0
Convert BDC:	1	0	0	0	0	1	1	1
Weights	0	1	0	1	0	1	1	1
Get binary number	0	1	0	1	0	1	1	1

Pulse Waveform/Diagram

- Pulse Waveform/Diagram is a tool to illustrate the process in digital system.
- The diagram displays any waveforms that we are interested to investigate.

Example: Draw the waveform diagram for *x* and *y* of the circuit with the inputs of *A*, *B* and *C*.

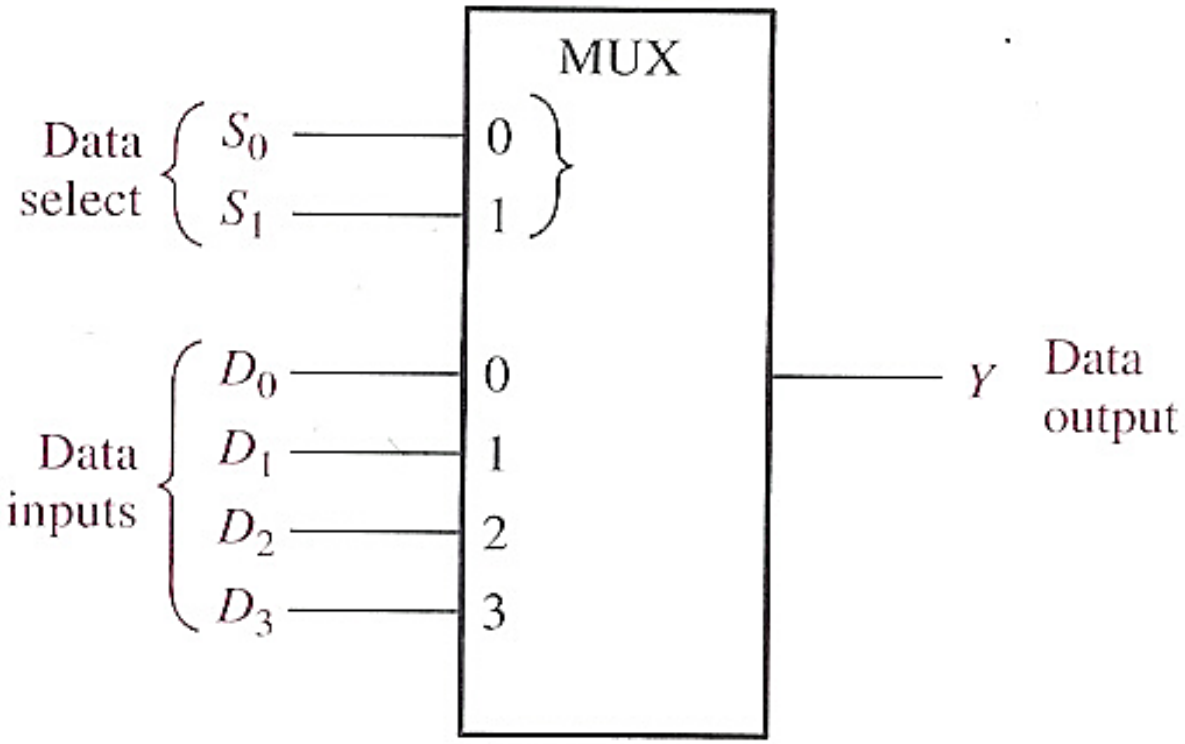


MULTIPLEXER

- A multiplexer (MUX) is a device that allows digital information from several sources to be routed onto a single line for transmission over that line to a common destination.
- The basic multiplexer has several data-input lines and a single output line. It also has data-select inputs, which permit digital data on any one of the inputs to be switched to the output line.

MULTIPLEXER

The Circuit symbol for a 4-to-1 Data Selector/Multiplexer.

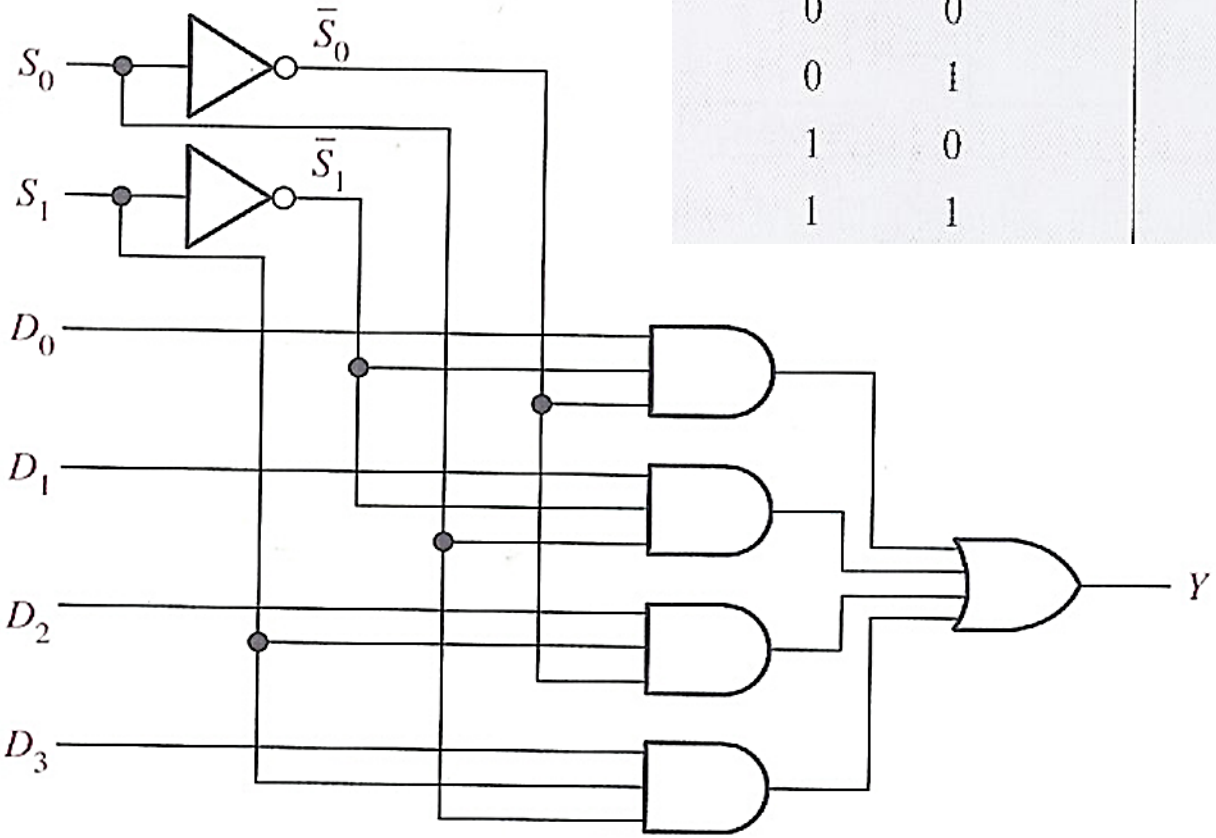


MULTIPLEXER

Data selection for 1-to-4-Multiplexer

Control codes

DATA-SELECT INPUTS S_1	S_0	INPUT SELECTED
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

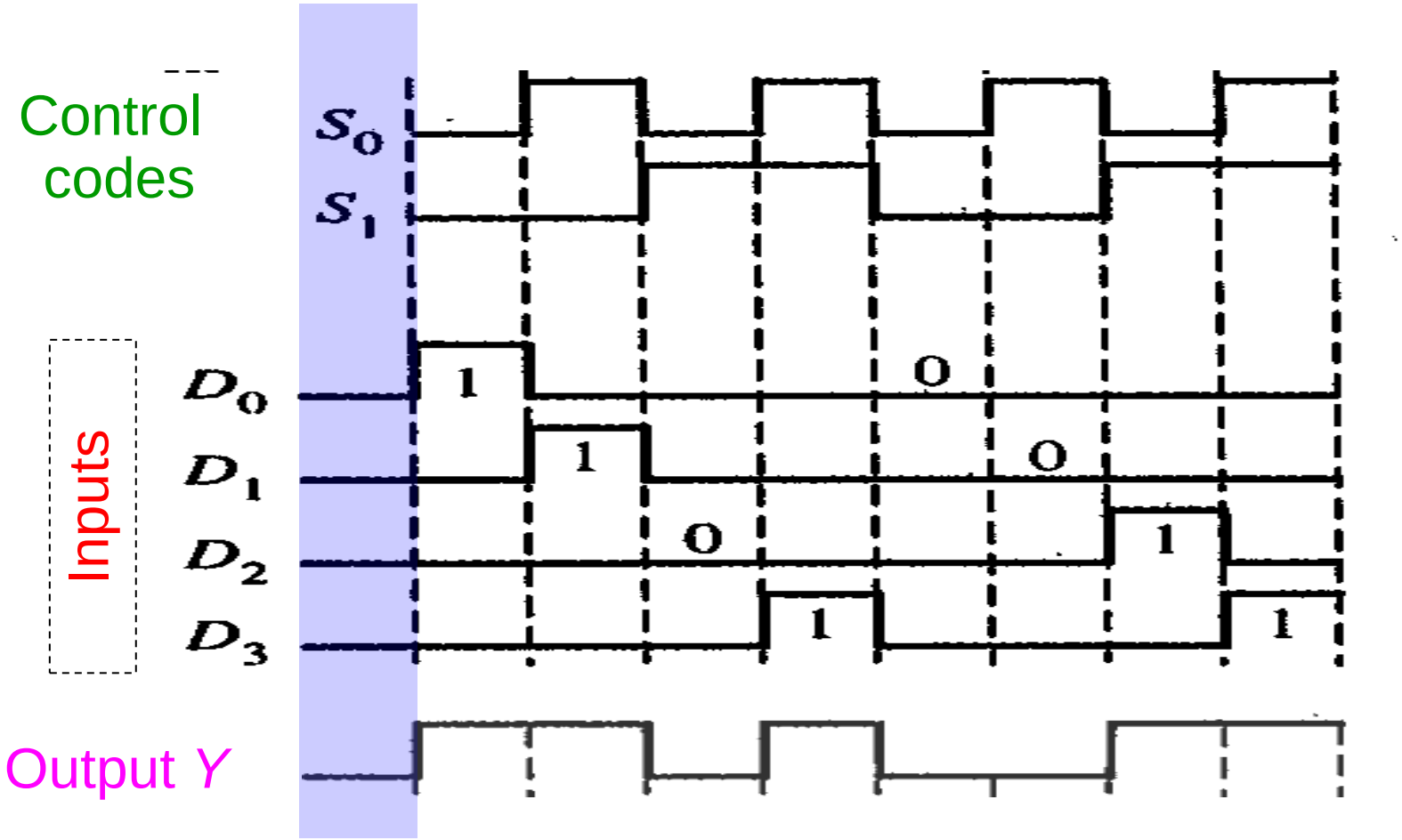


Inputs

Gates

MULTIPLEXER

The Pulse diagram for the input signals D_0 to D_3 and control signals S_0-S_1, the data output signal comparing with the output signals.



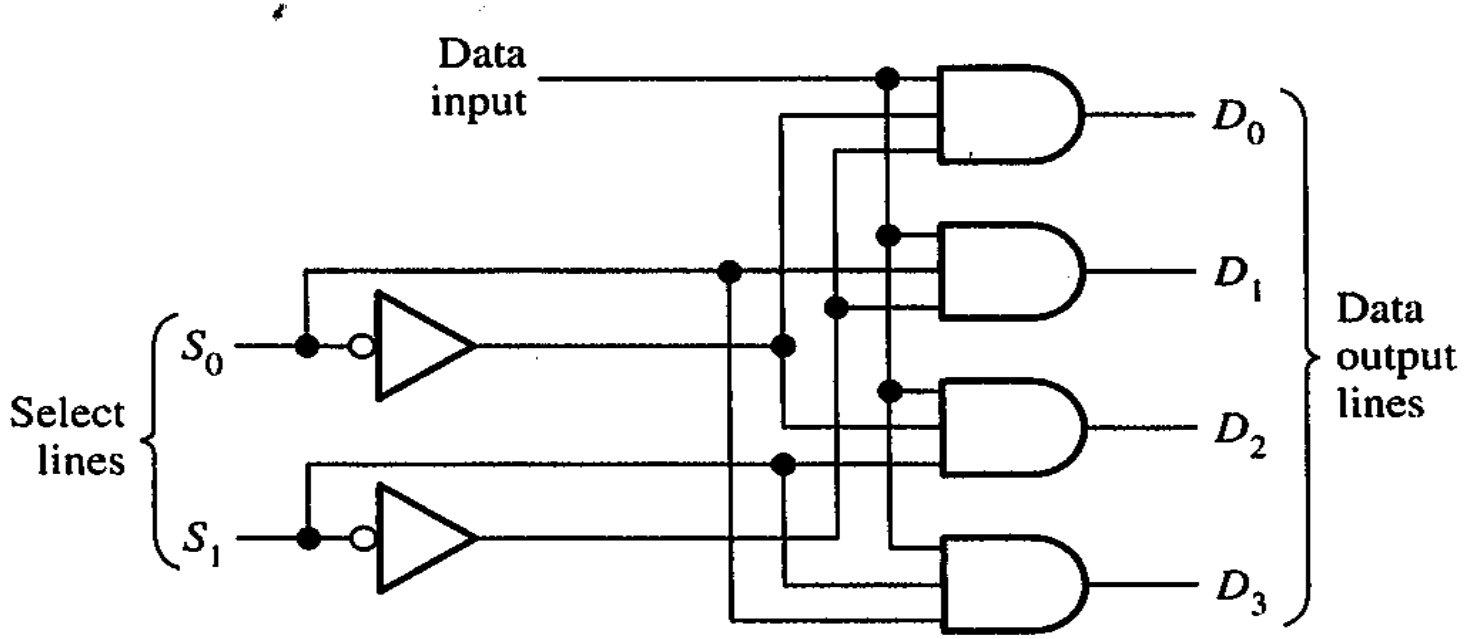
DEMULTIPLEXER

- A Demultiplexer (DEMUX) basically *reverses* the multiplexing function.
- It takes a digital information from ONE line and distributed it to a given NUMBER of output lines.
- Also known as a *data distributor.*

DEMULTIPLEXER

Schematic circuit diagram of the Demultiplexer with 4 Inputs

(1-to-4 Demultiplexer)



DEMULTIPLEXER

The Pulse diagram for the output signals D_0 to D_3 comparing with the data input signal.