电子技术
Introduction to Electronics

By Bao Qilian

鲍其莲

2018/10/23 1

---

4 Encoders

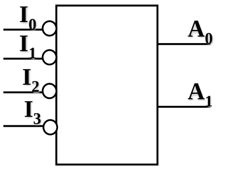
- 4-2 Encoders
- Decimal-to-BCD
- 8-3 Priority Encoders
- Expansion

Principle of Encoders

- Encoder performs a "reverse" decoder function. i.e. converts a decimal digit to a binary code.

4-2 Encoder

$\overline{I_0}$	$\overline{I_1}$	$\overline{I_2}$	$\overline{I_3}$	A_1	A_0
0	1	1	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	0	1	1



$$\begin{cases} A_0 = I_0 I_1 \overline{I_2} I_3 + I_0 I_1 I_2 \overline{I_3} = \overline{\overline{I_0} I_1 I_2 I_3 + I_0 I_1 \overline{I_2} I_3} \\ A1 = I_0 I_1 \overline{I_2} I_3 + I_0 I_1 I_2 \overline{I_3} = \overline{\overline{I_0} I_1 I_2 I_3 + I_0 \overline{I_1} I_2 I_3} \end{cases}$$

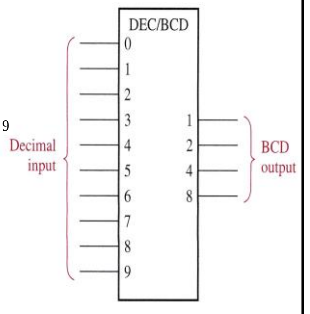
Decimal-to-BCD Encoder

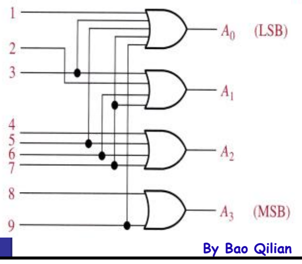
- Convert decimal number to BCD coder.

X_0

X_9	X_8	X_7	X_6	X_5	X_4	X_3	X_2	X_1	A_3	A_2	A_1	A_0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	1

$$A_3 = X_8 + X_9$$
$$A_2 = X_4 + X_5 + X_6 + X_7$$
$$A_1 = X_2 + X_3 + X_6 + X_7$$
$$A_0 = X_1 + X_3 + X_5 + X_7 + X_9$$





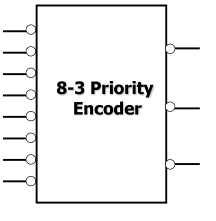
Note:

This kind of encoder works only when one input is active (HIGH) at a time, otherwise it will result in incorrect outputs.

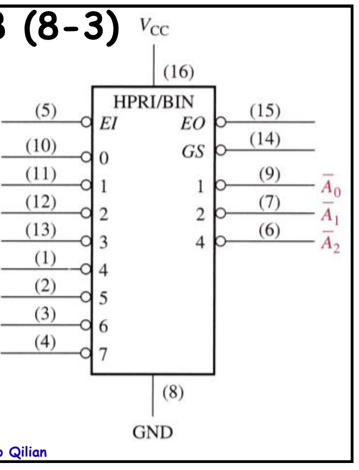
Decimal-to-BCD Priority Encoder
（优先编码器）

The priority function means that the encoder will produce a BCD output corresponding to the *highest-order* decimal digit input that is active and ignore any other lower-order active inputs.

8-3 Priority Encoder



74F148 (8-3)

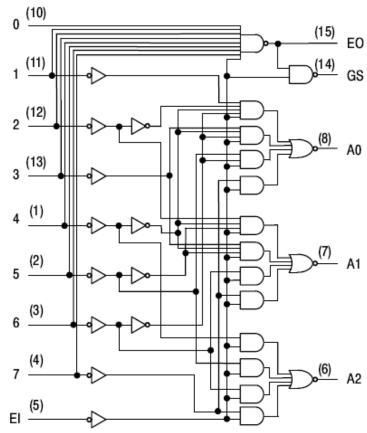


FUNCTION TABLE

INPUTS									OUTPUTS				
$\overline{EI}$	$\overline{I0}$	$\overline{I1}$	$\overline{I2}$	$\overline{I3}$	$\overline{I4}$	$\overline{I5}$	$\overline{I6}$	$\overline{I7}$	$\overline{GS}$	$\overline{A0}$	$\overline{A1}$	$\overline{A2}$	$\overline{EO}$
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	H	L	L	H
L	X	X	X	X	X	L	H	H	L	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	L	L	L	H	H
L	X	X	L	H	H	H	H	H	L	H	L	H	H
L	X	L	H	H	H	H	H	H	L	L	H	H	H
L	L	H	H	H	H	H	H	H	L	H	H	H	H

H = High voltage level
L = Low voltage level
X = Don't care

Logic diagram



2018/10/23

By

$$\overline{A_0} = \overline{7} + 76\overline{5} + 7654\overline{3} + 765432\overline{1} = \overline{7} + 6\overline{5} + 64\overline{3} + 642\overline{1}$$

$$\overline{A_1} = \overline{7} + 7\overline{6} + 7654\overline{3} + 76543\overline{2} = \overline{7} + \overline{6} + 54\overline{3} + 54\overline{2}$$

$$\overline{A_2} = \overline{7} + 7\overline{6} + 76\overline{5} + 765\overline{4} = \overline{7} + \overline{6} + \overline{5} + \overline{4}$$

$$\overline{EO} = 76543210 \cdot \overline{EI}$$

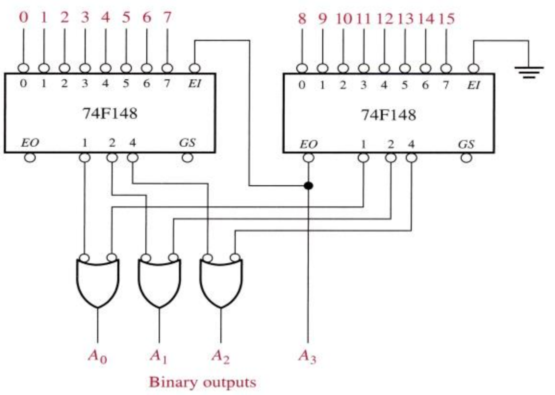
$$\overline{GS} = EO \cdot \overline{EI}$$

2018/10/23 14

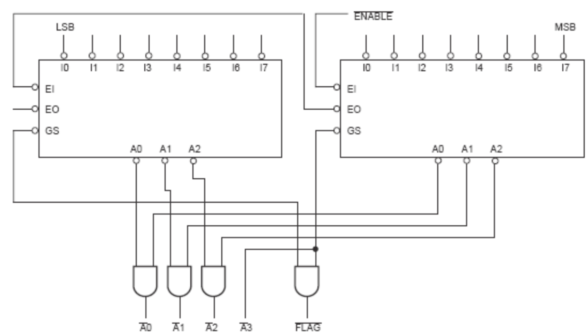
By Bao Qilian

Encoder

(I)



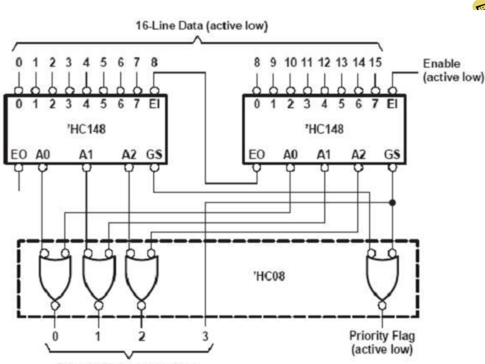
(II)



2018/10/23 16

By Bao Qilian

Priority Encoder for 16 Bits (Active Low Out)

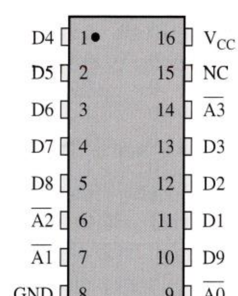


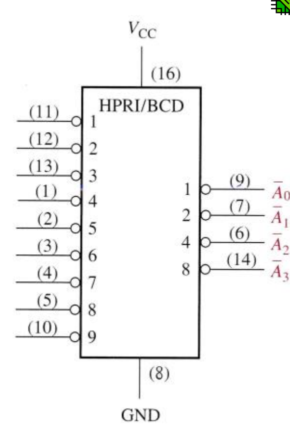
2018/10

By Bao Qilian

74LS147 (10-4)

(1-9, 0 for none)

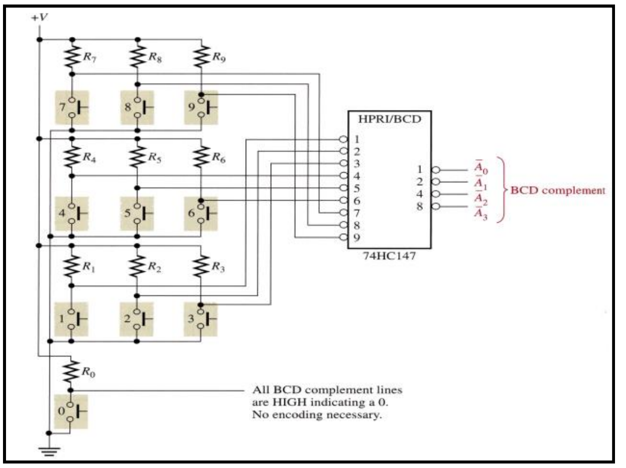




Q

Application example

Simplified keyboard encoder.



5 Coder Converter

- Convert one coder to another

BCD-Binary Conversion

- Steps

1. Determine the value or weight of each bit in the BCD number.
2. Add all the weights that the corresponding bits are 1s.
3. The sum is the equivalent binary number of BCD number.

Example: BCD00100111(27)→Binary number

1. The weights of each bit as follows:

BCD numbers	0	0	1	0	0	1	1	1
	↑	↑	↑	↑	↑	↑	↑	↑
weights	80	40	20	10	8	4	2	1
	↑	↑	↑	↑	↑	↑	↑	↑
Binary weights	1010000	0101000	0010100	0001010	0001000	0000100	0000010	0000001

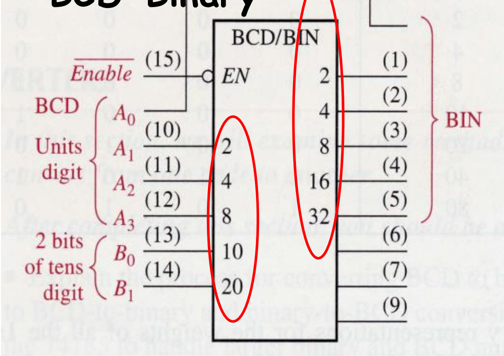
$99<127=2^7$

2. Sum of the weights:

80 40 20 10 8 4 2 1
0 0 1 0 0 1 1 1

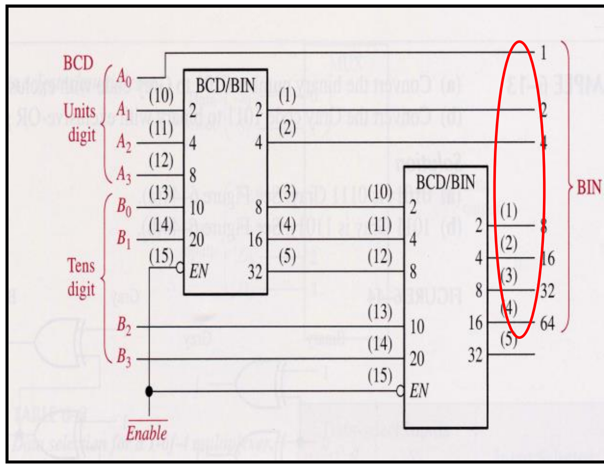
0000001	1
0000010	2
0000100	4
+ 0010100	20
0011011	Binary number for decimal 27

BCD-Binary

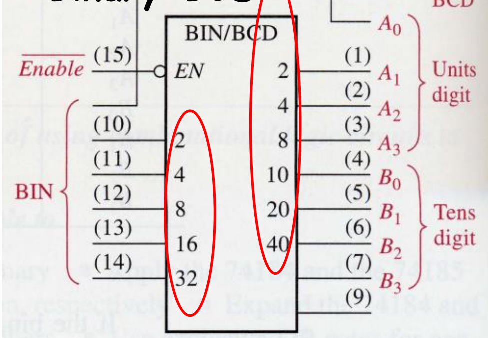


(a) The 74184 6-bit BCD-to-binary converter

2018/10/23

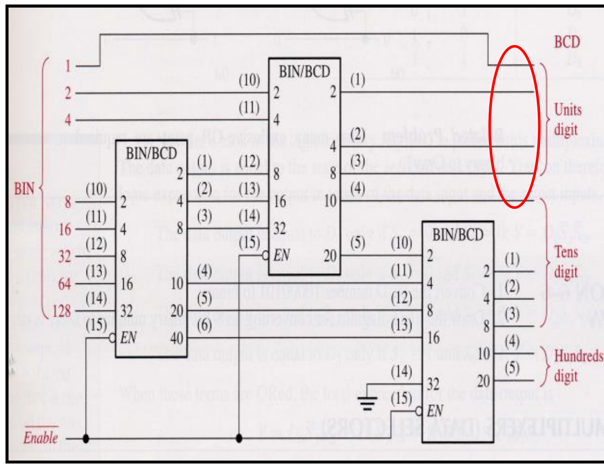


Binary-BCD



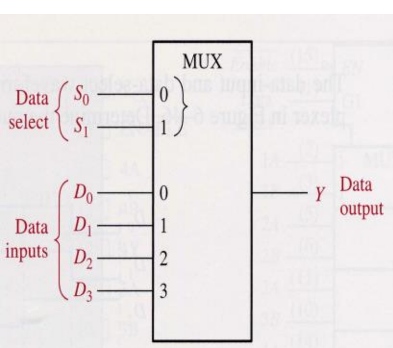
(b) The 74185 6-bit binary-to-BCD converter

2018/10/23



6 Data Selectors (Multiplexers)

数据选择器/多路开关

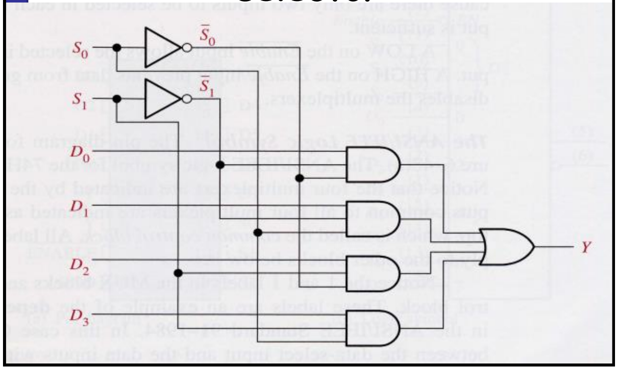


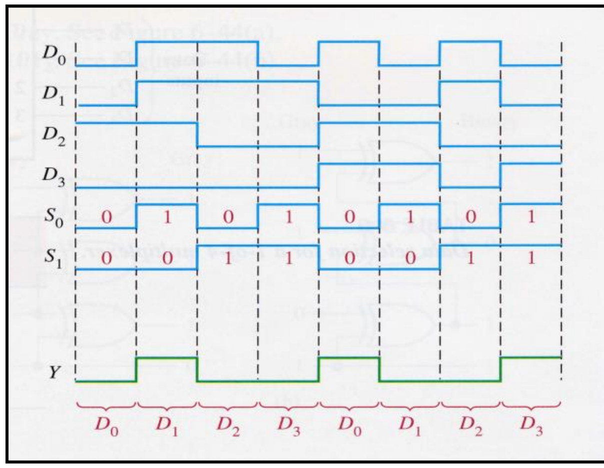
Data-select Inputs S_1	S_0	Input Selected
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$$Y = D_0\overline{S}_1\overline{S}_0 + D_1\overline{S}_1S_0 + D_2S_1\overline{S}_0 + D_3S_1S_0$$

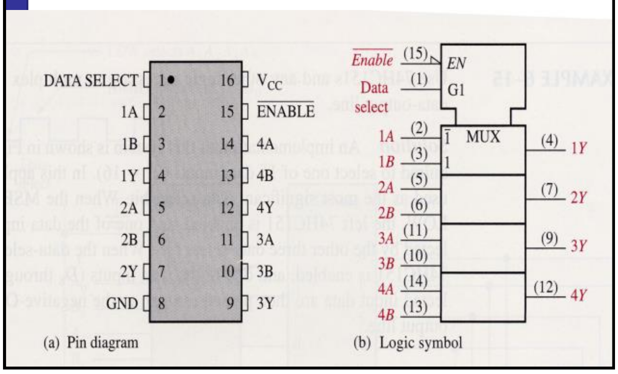
By Bao Qilian

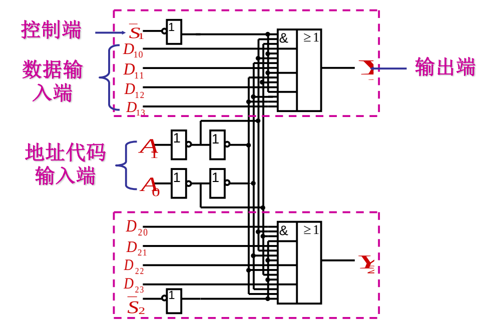
Logic Diagram for 4-bit data selector





74HC157A



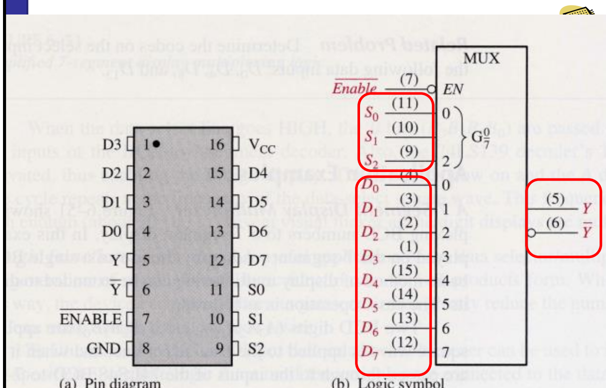


双4选1数据选择器74LS153

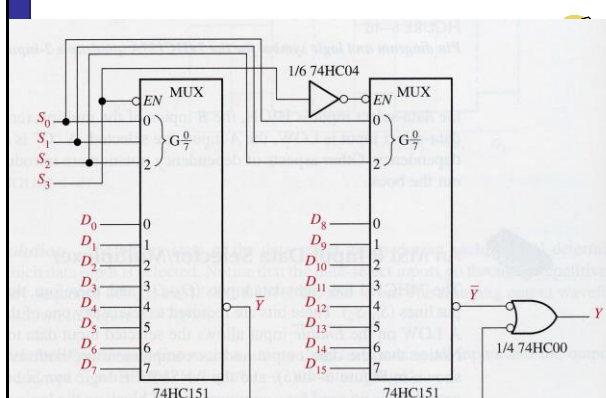
2018/10/23 34

By Bao Qilian

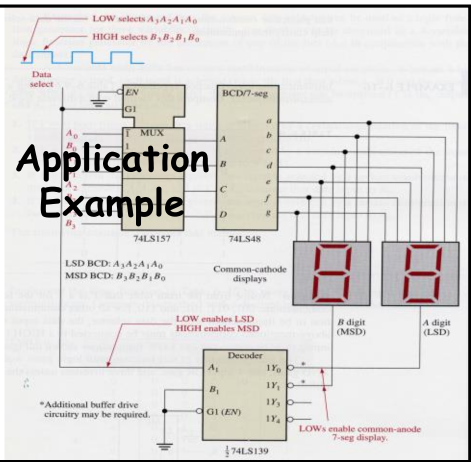
74HC151（8选1）



Expansion:16-inputs multiplexer



Application Example



2018/10/23

Example: Implement the function by using 74HC151.

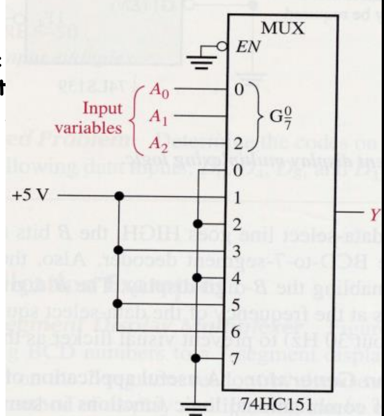
Inputs			Output
A_2	A_1	A_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$$Y = \bar{A}_2\bar{A}_1A_0 + \bar{A}_2A_1A_0 + A_2\bar{A}_1A_0 + A_2A_1\bar{A}_0 = \sum(1,3,5,6)$$

2018/10/23

By Bao Qilian

Logic Generat data



2018/10/23

By

74150（16选1）

Function Tables

54150/74150

Inputs					Outputs
Select				Strobe	W
D	C	B	A	S	
X	X	X	X	H	H
L	L	L	L	L	$\overline{E0}$
L	L	L	H	L	$\overline{E1}$
L	L	H	L	L	$\overline{E2}$
L	L	H	H	L	$\overline{E3}$
L	H	L	L	L	$\overline{E4}$
L	H	L	H	L	$\overline{E5}$
L	H	H	L	L	$\overline{E6}$
L	H	H	H	L	$\overline{E7}$
H	L	L	L	L	$\overline{E8}$
H	L	L	H	L	$\overline{E9}$
H	L	H	L	L	$\overline{E10}$
H	L	H	H	L	$\overline{E11}$
H	H	L	L	L	$\overline{E12}$
H	H	L	H	L	$\overline{E13}$
H	H	H	L	L	$\overline{E14}$
H	H	H	H	L	$\overline{E15}$

H = High Level, L = Low Level, X = Don't Care

$\overline{E0}$, $\overline{E1}$ . . . $\overline{E15}$ = the complement of the level of the respective E input

74151（8选1）

54151A/75151A

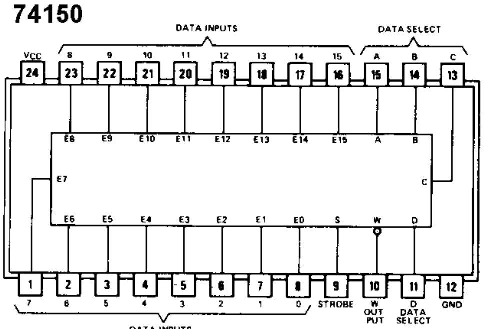
Inputs				Outputs	
Select			Strobe	Y	W
C	B	A	S		
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

H = High Level, L = Low Level, X = Don't Care

D0, D1 . . . D7 = the level of the respective D input

74150（16选1）

74150



By Bao Qilian

Decimal Digit	Inputs				Output
	A_3	A_2	A_1	A_0	Y
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1

2018/10/23

B

Exercise: implement the function by 74150.

Exercise:

- Find the dimension-reduced(降维) karnagh map of Y=F(A_0,A_1,A_2,A_3), assume that the values could be 0,1 and A_0.

Decimal Digit	A_3	A_2	A_1	A_0	Output Y
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1

Y=A_0

Y=1

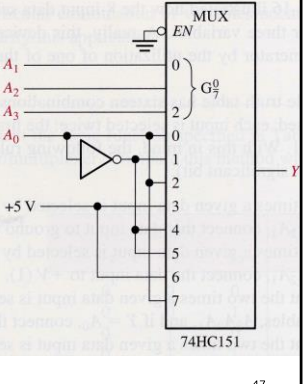
Example 6-17

$$Y=\overline{A_3}\,\overline{A_2}\,\overline{A_1}A_0+\overline{A_3}\,\overline{A_2}A_1\overline{A_0}+\overline{A_3}A_2\overline{A_1}A_0+\overline{A_3}A_2A_1\overline{A_0}+\overline{A_3}A_2A_1A_0$$
$$+A_3\overline{A_2}\,\overline{A_1}\,\overline{A_0}+A_3\overline{A_2}A_1\overline{A_0}+A_3A_2\overline{A_1}\,\overline{A_0}+A_3A_2\overline{A_1}A_0+A_3A_2A_1A_0$$
$$=(\overline{A_3}\,\overline{A_2}\,\overline{A_1})A_0+(\overline{A_3}\,\overline{A_2}A_1)\overline{A_0}+(\overline{A_3}A_2\overline{A_1})A_0+(\overline{A_3}A_2A_1)1$$
$$+(A_3\overline{A_2}\,\overline{A_1})\overline{A_0}+(A_3\overline{A_2}A_1)\overline{A_0}+(A_3A_2\overline{A_1})1+(A_3A_2A_1)A_0$$

$$Y=(\overline{S}_2\overline{S}_1\overline{S}_0)D_0+(\overline{S}_2\overline{S}_1S_0)D_1+(\overline{S}_2S_1\overline{S}_0)D_2+(\overline{S}_2S_1S_0)D_3$$
$$+(S_2\overline{S}_1\overline{S}_0)D_4+(S_2\overline{S}_1S_0)D_5+(S_2S_1\overline{S}_0)D_6+(S_2S_1S_0)D_7$$

$$S_2=A_3, S_1=A_2, S_0=A_1$$
$$D_0=D_2=D_7=A_0, D_1=D_4=D_5=\overline{A_0}, D_3=D_6=1$$

A_3 \ A_2A_1	00	01	11	10
0	A_0	$\overline{A_0}$	1	A_0
1	$\overline{A_0}$	$\overline{A_0}$	A_0	1



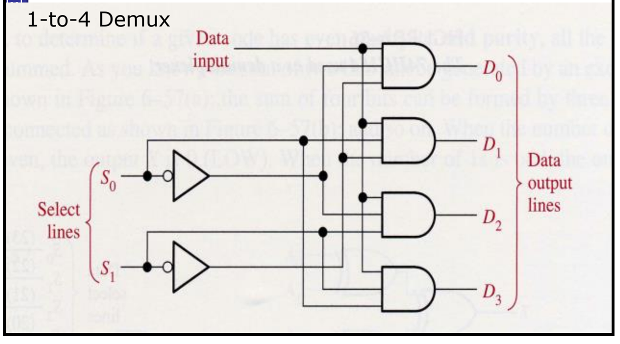
Exercise:

Please design a circuit with 74LS151 and NAND gates to implement the following functions:

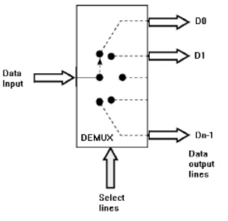
$$(1)Y(A,B,C)=\sum(1,3,5,6)$$
$$(2)Y(A,B,C,D)=\sum(1,3,4,5,6)$$

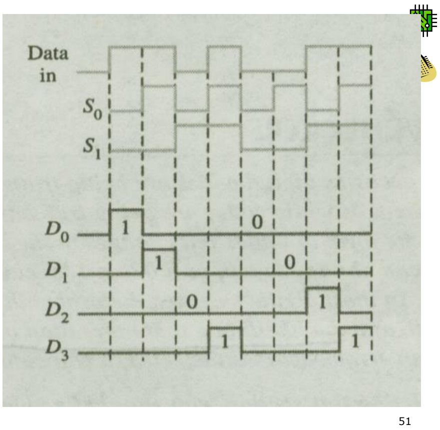
7 Demultiplexer(多路分配器)

1-to-4 Demux

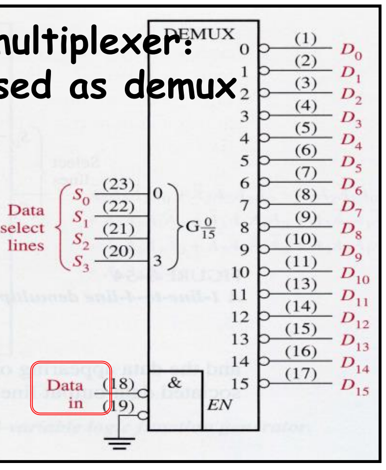


- Demux = demultiplexer = data distributor
- A demux basically reverses the multiplexing function.
- It takes data from one line and distributes them to a given number of output lines.





4-16 Demultiplexer: Decoder used as demux



---

8 Parity Generators/checkers

- To determine if a given code has *even parity* or *odd parity*.
- Basic Principle:
  - Parity bit:

    The sum of an *even number* of 1s is always 0, and the sum of *odd number* of 1s is always 1.

Even/Odd Parity

- **The basic idea is to attach a control bit (PARITY BIT) to a group of information bits to make the total number of 1 bits even (EVEN PARITY) or odd (ODD PARITY).**
- **Parity checking can detect single bit errors.**
- **Example:**

Information bits = 11010100 -> Even parity bit = 0

-> Bits transmitted = 110101000

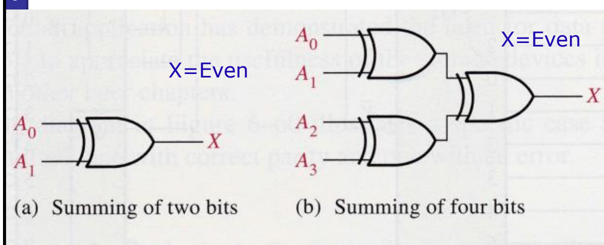
Single bit error occurs

Bits received = 100101000 -> Fail

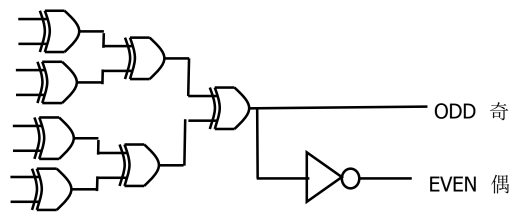
Double bit error occurs

Bits received = 000101000 -> Pass

Sum of 2-bit by XOR

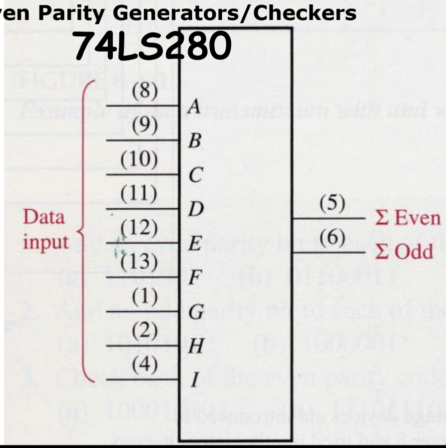


Parity checker

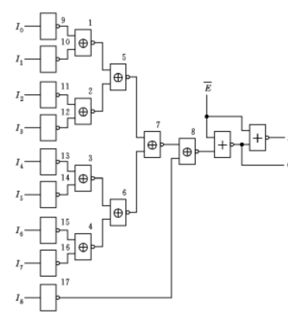


9-Bit Odd/Even Parity Generators/Checkers

74LS280



Nine-bit parity checker



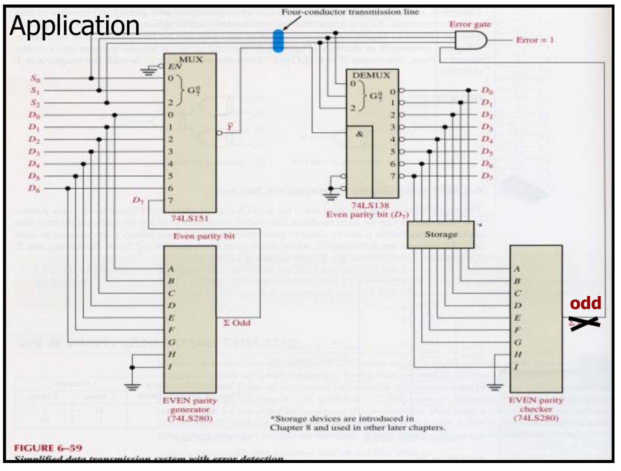
Function Table

Number of Inputs *A* Thru *I* That Are HIGH	Outputs	
	Σ Even	Σ Odd
0, 2, 4, 6, 8	H	L
1, 3, 5, 7, 9	L	H

Generator / Checker

- *Parity Checker*: the number of inputs should always be even/odd, when the output is different (odd/even), an error occurs.
- *Parity Generator*:
  - For *even* parity generator, take odd output as the parity bit;
  - For *odd* parity generator, take even output as the parity bit.

Application



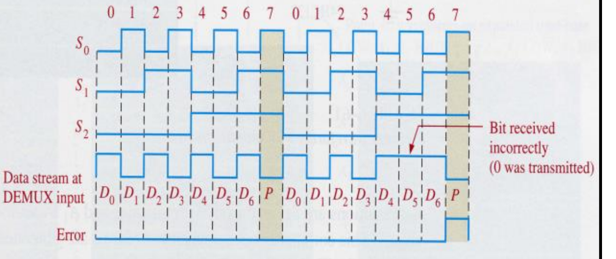


FIGURE 6–60
Example of data transmission with and without error for the system in Figure 6–59.

summary

- MSI combinational logic circuits
  - Adders(加法器)
  - Comparator（比较器）
  - Decoders（译码器）
  - Encoders（编码器）
  - Code converters（码转换器）
  - Multiplexer (data selectors)（数据选择器）
  - De-multiplexers（多路复用器）
  - Parity generators/checkers（奇偶校验器）
- Analyze and design of combinational logic circuits

Chapter 6 Functions of Combinational Logic

Problems：

22， 24 27(c,d), 28,30,

补充题

［题 3.7］ 某医院有一、二、三、四号病室 4 间，每室设有呼叫按钮，同时在护士值班室内对应地装有一号、二号、三号、四号 4 个指示灯。

现要求当一号病室的按钮按下时，无论其他病室的按钮是否按下，只有一号灯亮。当一号病室的按钮没有按下而二号病室的按钮按下时，无论三、四号病室的按钮是否按下，只有二号灯亮。当一、二号病室的按钮都未按下而三号病室的按钮按下时，无论四号病室的按钮是否按下，只有三号灯亮。只有在一、二、三号病室的按钮均未按下而按下四号病室的按钮时，四号灯才亮。试用优先编码器 74LS148 和门电路设计满足上述控制要求的逻辑电路，给出控制四个指示灯状态的高、低电平信号。

［题 3.12］ 用 3 线－8 线译码器 74LS138 和门电路设计 1 位二进制全减器电路。输入为被减数、减数和来自低位的借位；输出为两数之差和向高位的借位信号。