DIGITAL TECHNICS I

Dr. Bálint Pődör

Óbuda University, Microelectronics and Technology Institute

9. LECTURE: FUNCTIONAL BUILDING BLOCKS I

1st year BSc course 1st (Autumn) term 2018/2019

1

FUNCTIONAL BUILDING BLOCKS I

1. Functional building blocks, an overview
2. Encoders, decoders, code converters
3. Code conversion (combinational) networks

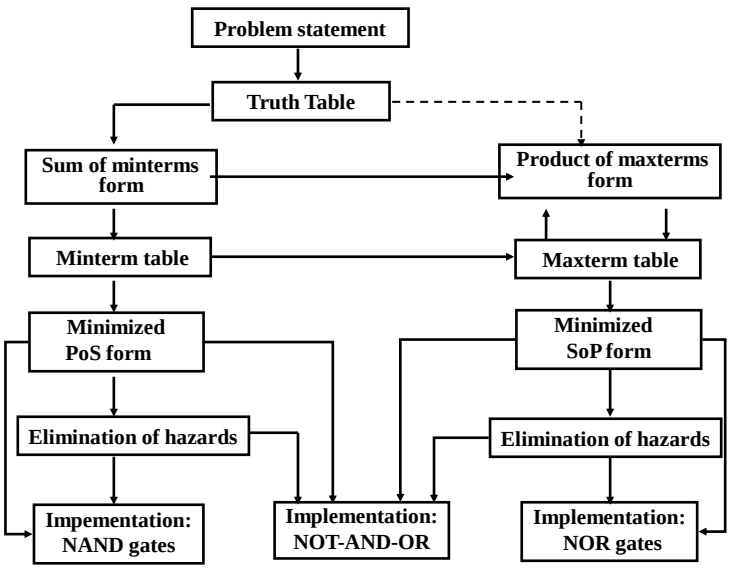
2

SYNTHESIS USING LOGIC GATES

The traditional process of logic synthesis is based on the application of logic gates.

Its more modern variant makes use of programmable logic devices too.

However in many case it is more advantageous to use a logic synthesis procedure based on the application of logic functional blocks.



Flow diagram of logic synthesis procedure using gates

DIGITAL SYNTHESIS: BUILDING BLOCKS

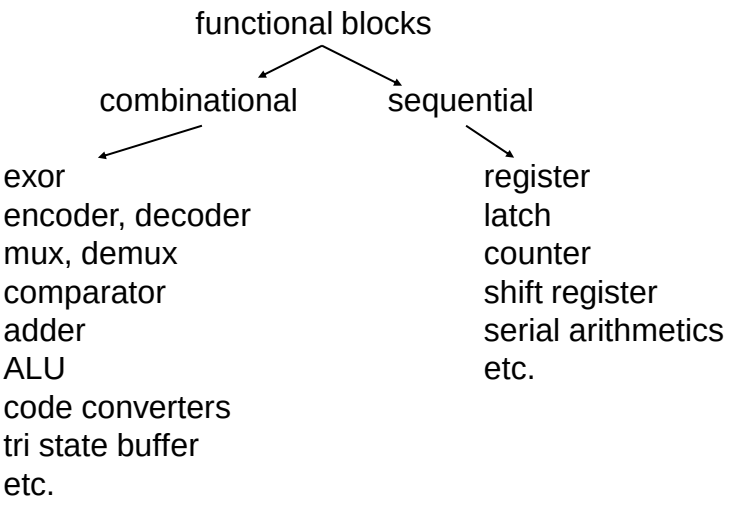
Lower level of abstraction: gates

Higher hierarchy: functional building blocks

- Encoders, decoders
- Multiplexers, demultiplexers
- Shifters
- Registers, memories
- Comparators
- Adders, etc. (binary arithmetic blocks)

Technological realization: SSI/MSI circuits

FAMILY TREE OF FUNCTIONAL BLOCKS



DIGITAL COMPONENTS

High level digital circuit designs are normally made using collections of logic gates referred to as components, rather than using individual logic gates.

Levels of integration (numbers of gates) in an integrated circuit (IC):

Small scale integration (SSI): about 10 gates.
Medium scale integration (MSI): 10 to 100 gates.
Large scale integration (LSI): 100-1,000 logic gates.
Very large scale integration (VLSI): 1,000-upward.
Ultra large scale integration (ULSI): 10,000-upward.
Giga large scale Integration (GLSI): 100, 000 upward.
Ridiculously (?) large scale integration (RLSI): 1,000,000 upward.

These levels are approximate, but the distinctions are useful in comparing the relative complexity of circuits.

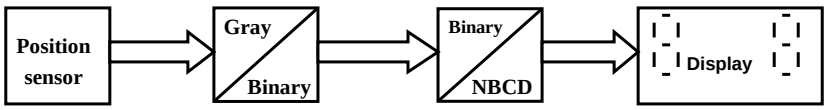
CODE CONVERSION NETWORKS, ENCODERS, DECODERS,

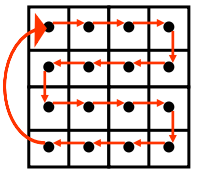
An important application of combinational networks in digital systems is as *code converters*. Much of the ”mystery” surrounding computers and other digital systems or gadgets stems from the unfamiliar language of digital circuits. Digital devices can process only 1s and 0s. For that reason, code converters are necessary to convert from the language of people to the language of the machine.

Code converters ary typically multiple input multiple output combinational circuits. They can be realized by appropropriate gate networks or using ROMs.

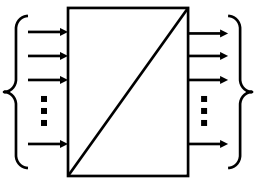
8

CODE CONVERSION EXAMPLE





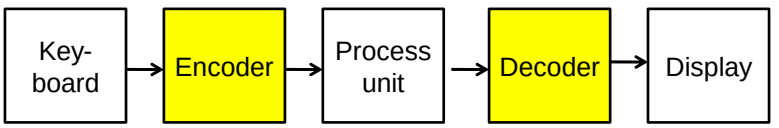
Gray
0000
0001
0011
0010
0110
0111
0101
0100
1100
1101
1111
1110
1010
1011
1001
1000



10	1
8421	8421
0000	0000
0000	0001
0000	0010
0000	0011
0000	0100
0000	0101
0000	0110
0000	0111
0000	1000
0000	1001
0001	1010
0001	1011
0001	1100
0001	1101
0001	1110
0001	1111

9

CODE CONVERTERS: POCKET CALCULATOR



Functional diagram of a pocket calculator: role of encoders and decoders

10

TYPICAL TASKS, TYPICAL NETWORKS

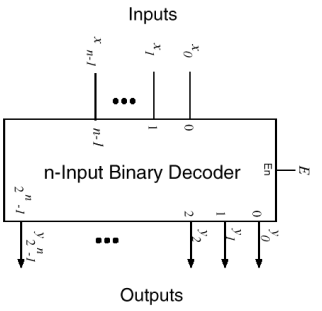
Input code	Output code
n-bit binary code word	one-out of-2^n
BCD digit	decimal digit (1-out of-10)
Excess-3 coded digit	decimal digit (1-out of-10)
BCD digit	seven-segment display
Binary number	BCD coded dumber
BCD coded number	binary number

11

DECODERS

A combinational circuit that converts binary information from *n* coded inputs to a maximum 2^n coded outputs
→ *n-to-* 2^n decoder, more generally *n-to-m* decoder, $m \leq 2^n$

Enable input: it must be on (active) for the decoder to function, otherwise its outputs assume a single ”disabled” output code word



Examples: BCD-to-7-segment decoder, where *n=4* and *m=10*

At any time only one output can be active (true), all others are false.

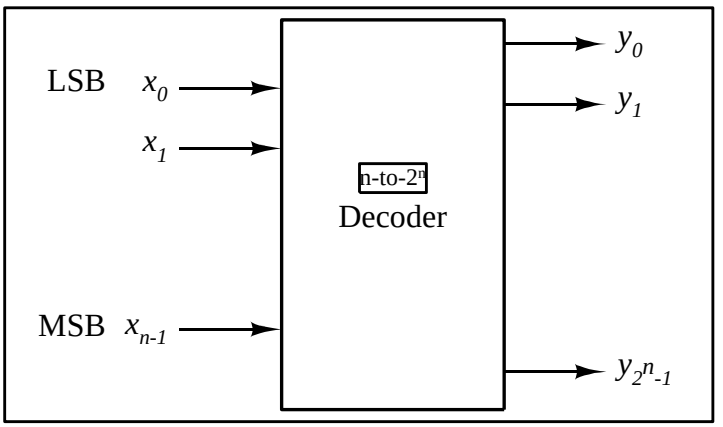
DECODER

A decoder asserts one out of n output lines, depending on the value of an m-bit binary input data. In general, an m-to-n decoder has m input lines, A_{m-1}, …, A_0, and n output lines, Y_{n-1}, …, Y_0, where $n = 2^m$. In addition, it has an enable line, E, for enabling the decoder. When the decoder is disabled with E set to 0, all the output lines are de-asserted. When the decoder is enabled, then the output line whose index is equal to the value of the input binary data is asserted.

For example, for a 3-to-8 decoder, if the input address is 101, then the output line Y_5 is asserted (set to 1 for active-high) while the rest of the output lines are de-asserted (set to 0 for active-high).

13

N-TO-2^N LINE DECODER



At any time only one output is active

14

DECODER

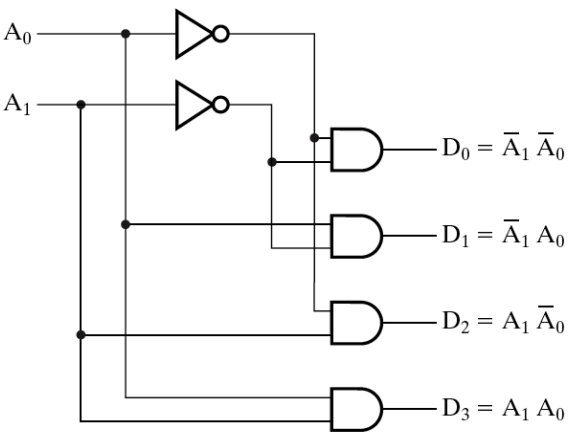
A decoder is used in a system having multiple components, and we want only one component to be selected or enabled at any one time. For example, in a large memory system with multiple memory chips, only one memory chip is enabled at a time. One output line from the decoder is connected to the enable input on each memory chip. Thus, an address presented to the decoder will enable that corresponding memory chip.

15

2-TO-4 LINE DECODER

A_1	A_0	D_0	D_1	D_2	D_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

(a)

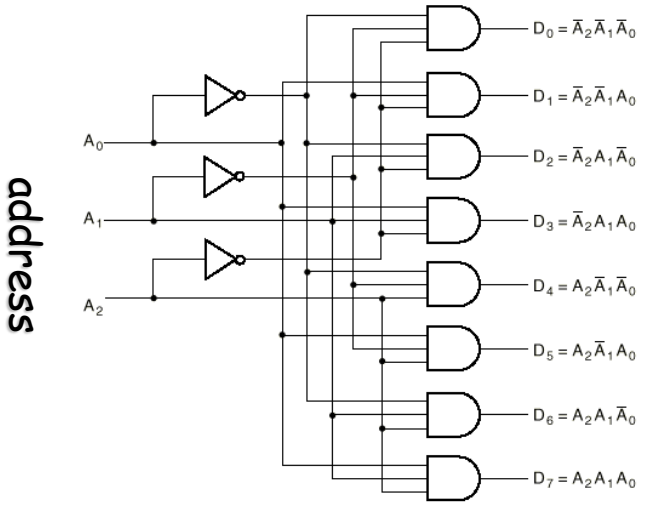


(b)

2-to-4 line decoder, gate level logic diagram

16

3-TO-8 LINE DECODER



3-to-8 line decoder, gate level logic diagram

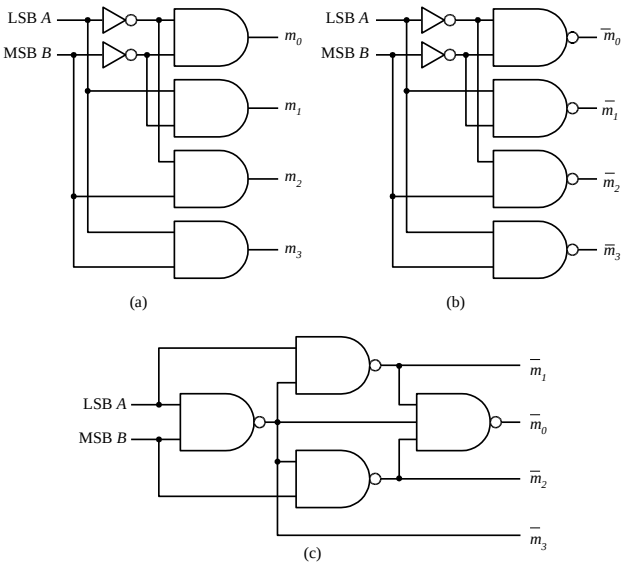
17

3-TO-8 DECODER (CONT.)

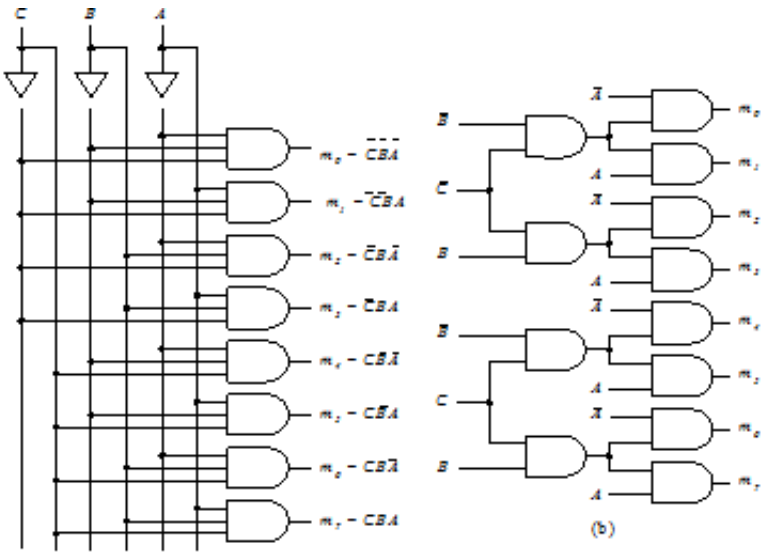
- Three inputs, A_0, A_1, A_2, are decoded into eight outputs, D_0 through D_7
- Each output D_i represents one of the minterms of the 3 input variables.
- $D_i = 1$ when the binary number $A_2A_1A_0 = I$
- Shorthand: $D_i = m_i$
- The output variables are *mutually exclusive*; exactly one output has the value 1 at any time, and the other seven are 0.

18

DECODER REALIZATIONS

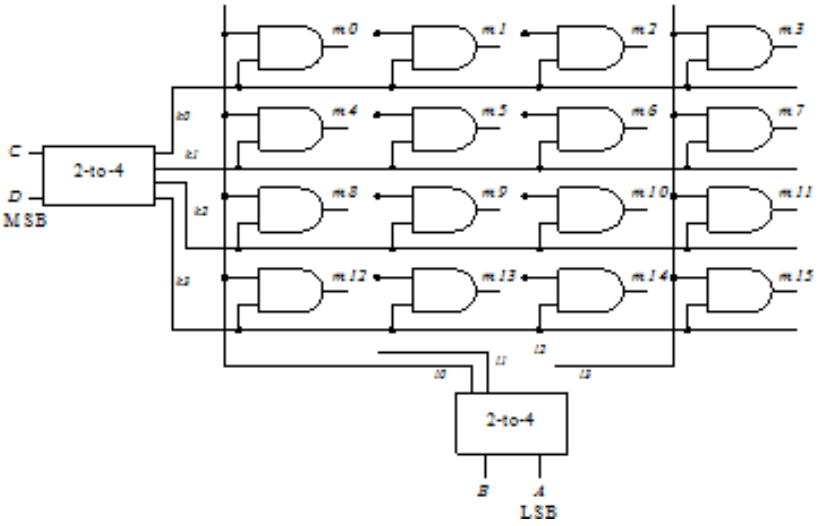


DECODER GATE LEVEL DIAGRAMS



8-line decoders, gate-level logic diagrams

DECODER GATE LEVEL DIAGRAMS

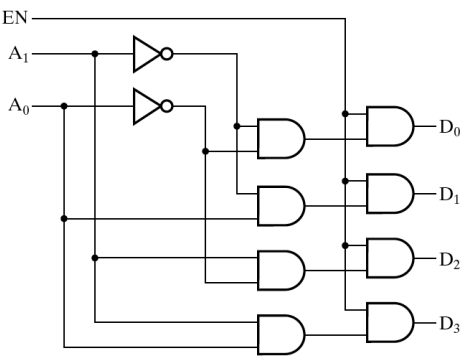


16-line decoder, gate-level logic diagram

DECODER WITH ENABLE: 2-TO-4

EN	A_1	A_0	D_0	D_1	D_2	D_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

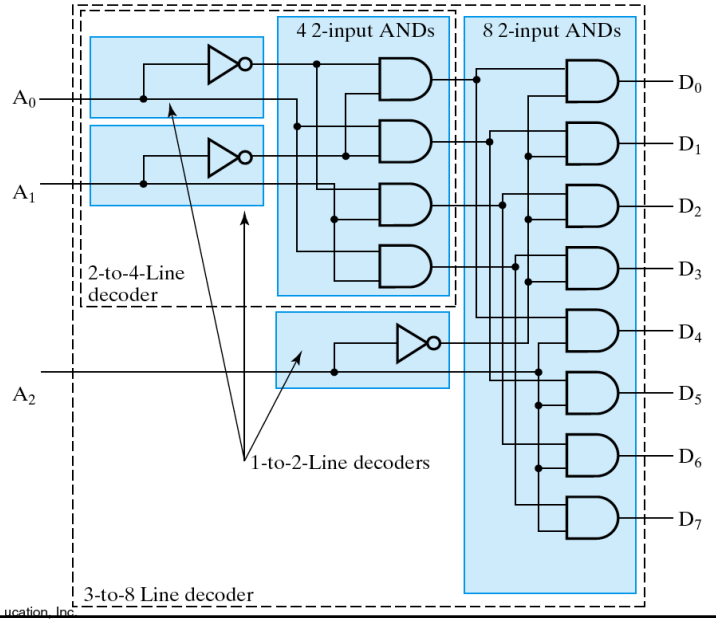
(a)



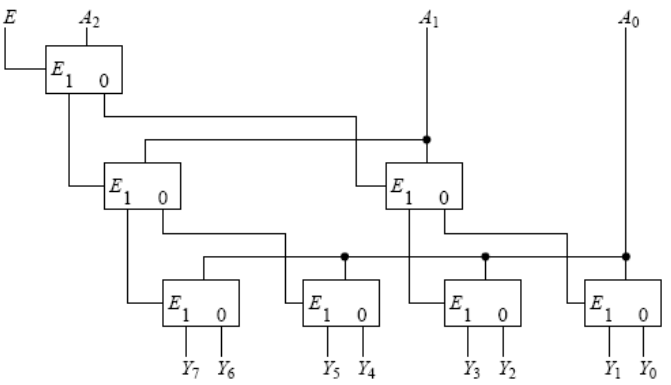
(b)

Additional gate level: time delay

DECODER EXPANSION: GATE LEVEL



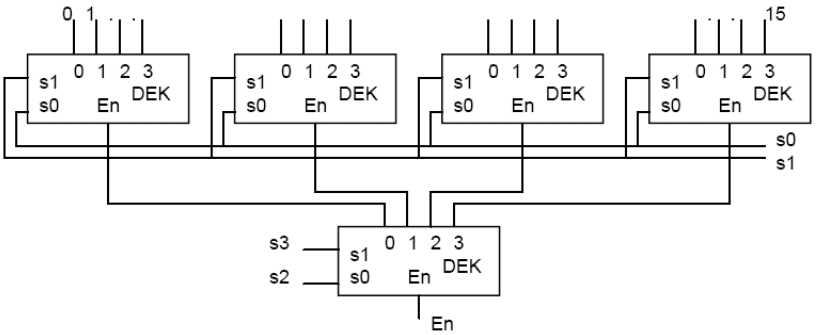
DECODER EXPANSION



A 3-to-8 decoder implemented with seven 1-to-2 decoders. Note the tree structure and the utilization of ENABLE inputs in logic role.

24

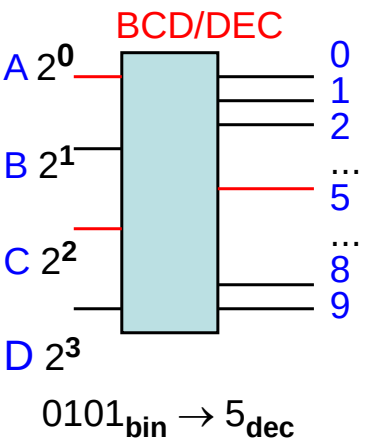
DECODER EXPANSION: MODULAR LEVEL



Decoder expansion (e.g. from 2-to-4 to 4-to-16) using the enable inputs. Note the tree structure.

Usual MSI implementations: 2-to-4, 3-to-8, 4-to-16 decoders.

APPLICATIONS: BCD(BIN)/DECIMAL DECODER/CONVERTER



$0101_{bin} \rightarrow 5_{dec}$

A common application: BCD/DEC decoder.

Number of inputs: 4, number of outputs 10.

Active output either H level (1) or L level (1) depending on the implementation, or module type.

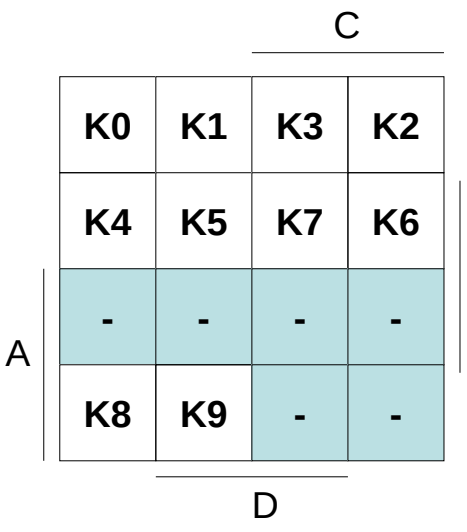
26

BCD/DECIMAL DECODER: TRUTH TABLE

A B C D	0	1	2	3	4	5	6	7	8	9
0 0 0 0	1	0	0	0	0	0	0	0	0	0
0 0 0 1	0	1	0	0	0	0	0	0	0	0
.....										
1 0 1 0	0	0	0	0	0	0	0	0	0	1
1 0 1 1	X	X	X	X	X	X	X	X	X	X
.....										
1 1 1 1	X	X	X	X	X	X	X	X	X	X

27

BCD/DECIMAL DECODER: KARNAUGH MAPS



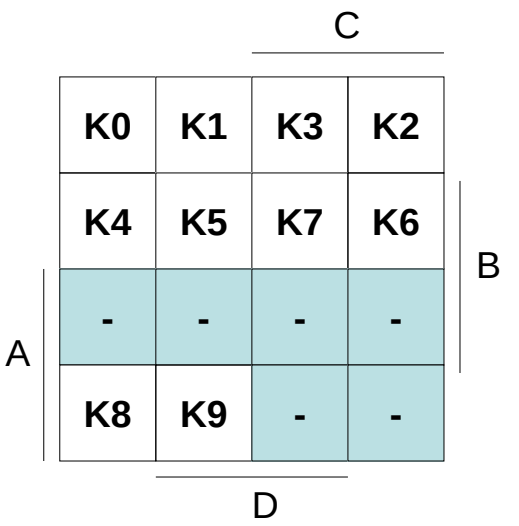
K0 and K1 cannot be simplified.

Outputs K3 ... K7 can be simplified by looping with one don't care minterm.

K8 and K9 can be looped with three don't care terms.

28

RESULT OF MINIMIZATION



$K0 = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$

$K1 = \overline{A}\,\overline{B}\,\overline{C}\,D$

$K2 = \overline{B}\,C\,\overline{D}$

.....

$K8 = A\,\overline{D}$

$K9 = A\,D$

29

BCD/DEC DECODER

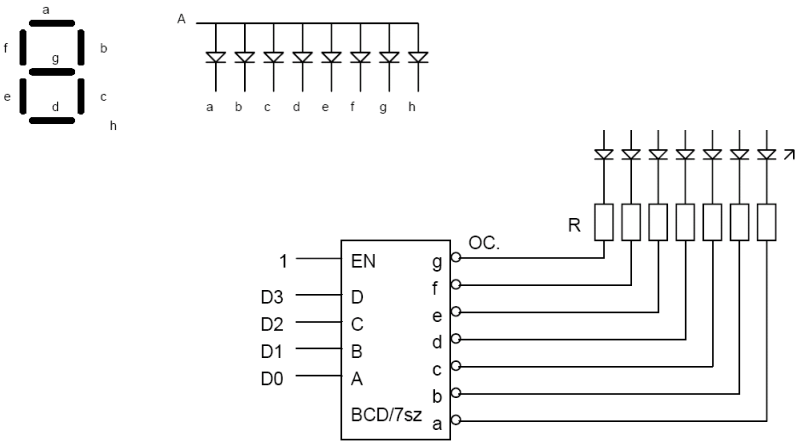
In MSI implementation it is not necessary to economize with gates, so the not used combinations can be used to realize auxiliary circuit functions.

E.g. the outputs can be disabled for the illegal input combinations. However other prescriptions can also be applied. The main thing is that it is possible to define the outputs for the illegal combinations too.

Practical examples: ROM- based BCD-to-binary, binary-to-BCD decoder, 74184, 74185, 6 (5+1) bit, 1st bit bypassed, cascadable.

30

BCD-TO-SEVEN SEGMENT DECODER

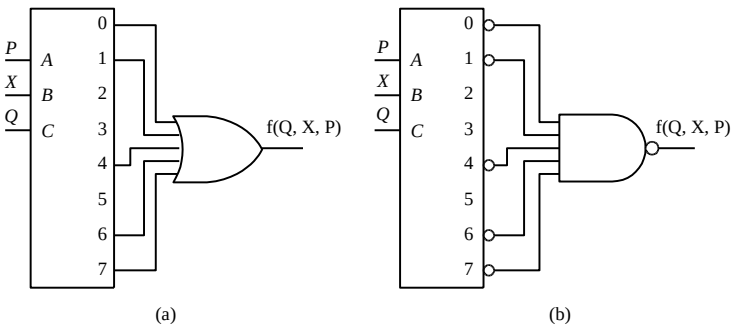


MSI BCD-to-seven segment decoders 7446, ..47, etc.

31

DECODER APPLICATION: IMPLEMENTING BOOLEAN FUNCTIONS USING DECODERS

Any combinational circuit can be constructed using decoders and OR gates! The decoder generates the required minterms and an external OR gate is used to produce the sum of minterms



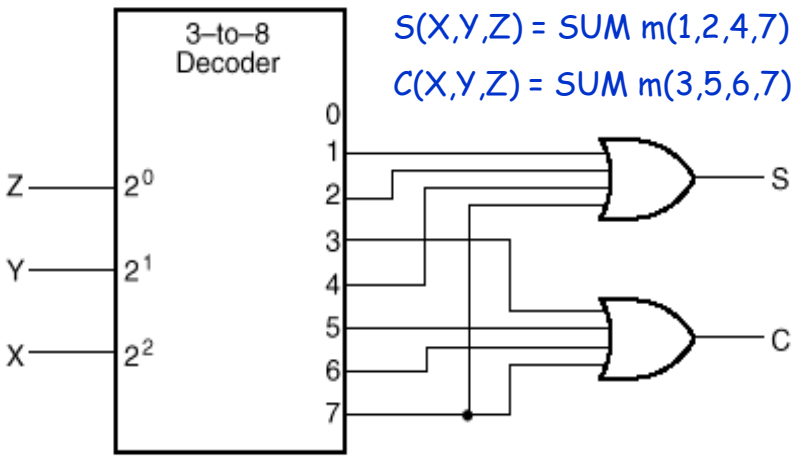
Realize $f(Q,X,P) = \sum m(0,1,4,6,7) = \prod M(2,3,5)$

32

DECODER APPLICATION: IMPLEMENTING BOOLEAN FUNCTIONS USING DECODERS

- Any combinational circuit can be constructed using decoders and OR gates!
- A decoder can be conveniently used to implement a given Boolean function. The decoder generates the required minterms and an external OR gate is used to produce the sum of minterms.
- Example: Implement a full adder circuit with a decoder and two OR gates.
- Recall full adder equations, and let X, Y, and Z be the inputs:
  - $S(X,Y,Z) = X+Y+Z = \Sigma m(1,2,4,7)$
  - $C(X,Y,Z) = \Sigma m(3, 5, 6, 7)$.
- Since there are 3 inputs and a total of 8 minterms, we need a 3-to-8 decoder.

IMPLEMENTING THE FULL ADDER USING A 3-to-8 LINE DECODER

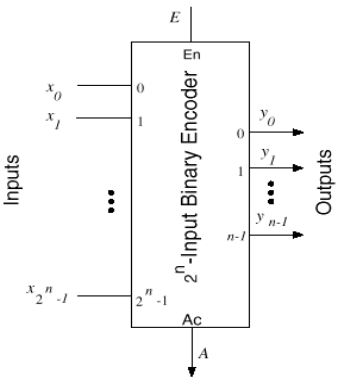


34

ENCODERS

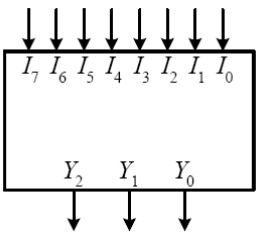
An encoder is a digital circuit that performs the inverse operation of a decoder. An encoder has 2^n input lines and n output lines.

The output lines generate the binary equivalent of the input line whose value is 1.



ENCODERS

An encoder is almost like the inverse of a decoder where it encodes a 2^n-bit input data into an n-bit code. The encoder has 2^n input lines and n output lines, as shown here for n = 3.



The operation of the encoder is such that exactly one of the input lines should have a 1 while the remaining input lines should have 0's. The output is the binary value of the index of the input line that has the 1. For example, when input I_3 is a 1, the three output bits Y_2, Y_1, and Y_0, are set to 011, which is the binary number for the index 3.

Entries having multiple 1's in the truth table inputs are ignored, since we are assuming that only one input line can be a 1.

ENCODERS

Encoders are used to reduce the number of bits needed to represent some given data either in data storage or in data transmission. Encoders are also used in a system with 2^n input devices, each of which may need to request for service. One input line is connected to one input device. The input device requesting for service will assert the input line that is connected to it. The corresponding n-bit output value will indicate to the system which of the 2^n devices is requesting for service. For example, if device 5 requests for service, it will assert the I_5 input line. The system will know that device 5 is requesting for service, since the output will be 101 = 5. However, this only works correctly if it is guaranteed that only one of the 2^n devices will request for service at any one time.

37

ENCODER EXAMPLE

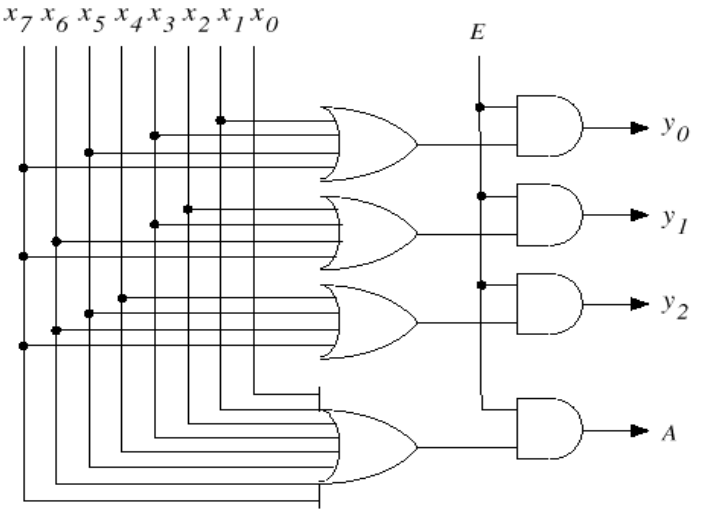
Example: 8-to-3 (octal-to-binary) encoder

Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

$$A_0 = D_1 + D_3 + D_5 + D_7$$
$$A_1 = D_2 + D_3 + D_6 + D_7$$
$$A_2 = D_4 + D_5 + D_6 + D_7$$

38

ENCODER EXAMPLE: 8-INPUT BINARY ENCODER WITH ENABLE



39

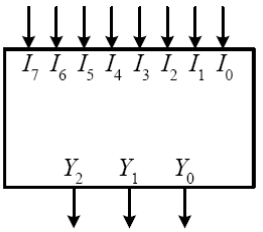
ENCODER DESIGN ISSUES

There are two ambiguities associated with the design of a simple encoder:

Only one input can be active at any given time. If two inputs are active simultaneously, the output produces an undefined combination (for example, if D_3 and D_6 are 1 simultaneously, the output of the encoder will be 111).

An output with all 0's can be generated when all the inputs are 0's,or when D_0 is equal to 1.

PRIORITY ENCODER



If two or more devices request for service at the same time, then the output will be incorrect.
For example, if devices 1 and 4 of the 8-to-3 encoder request for service at the same time, then the output will also be 101, because I_4 will assert the Y_2 signal, and I_1 will assert the Y_0 signal. To resolve this problem, a priority is assigned to each of the input lines so that when multiple requests are made, the encoder outputs the index value of the input line with the highest priority. This modified encoder is known as a *priority encoder*.

41

PRIORITY ENCODERS

Solves the ambiguities mentioned above.

Multiple asserted inputs are allowed; one has priority over all others.

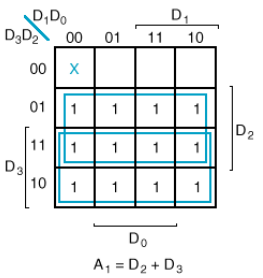
Separate indication of no asserted inputs.

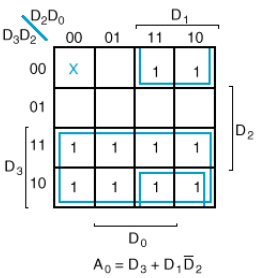
EXAMPLE: 4-TO-2 PRIORITY ENCODER

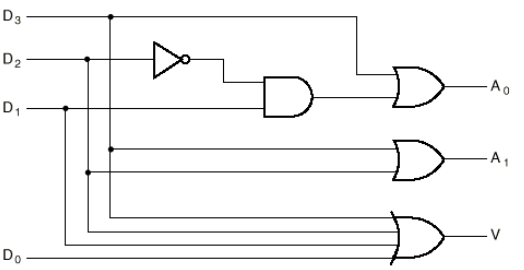
Inputs				Outputs		
D_3	D_2	D_1	D_0	A_1	A_0	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

The operation of the priority encoder is such that:
If two or more inputs are equal to 1 at the same time, the input in the highest-numbered position will take precedence.
A *valid output indicator*, designated by V, is set to 1 only when one or more inputs are equal to 1.
$V = D_3 + D_2 + D_1 + D_0$ by inspection.

4-TO-2 PRIORITY ENCODER: KARNAUGH MAP AND LOGIC DIAGRAM







8-TO-3 PRIORITY ENCODER

A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	Z_0	Z_1	Z_2	NR
0	0	0	0	0	0	0	0	X	X	X	1
X	X	X	X	X	X	X	1	1	1	1	0
X	X	X	X	X	X	1	0	1	1	0	0
X	X	X	X	X	1	0	0	1	0	1	0
X	X	X	X	1	0	0	0	1	0	0	0
X	X	X	1	0	0	0	0	0	1	1	0
X	X	1	0	0	0	0	0	0	1	0	0
X	1	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0

Implementation: MSI

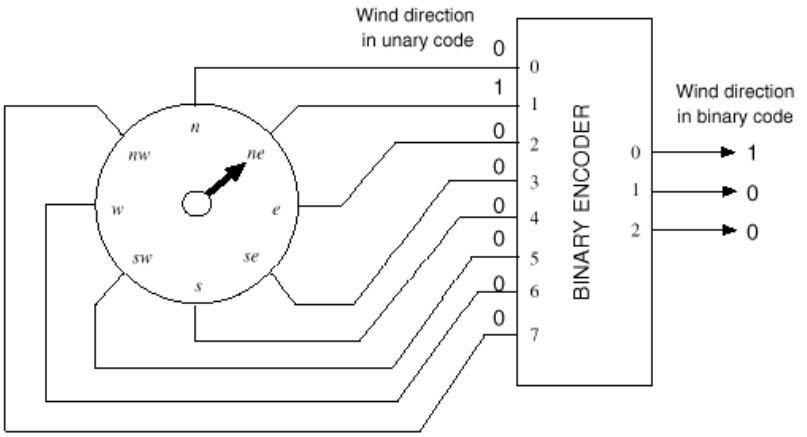
DECODERS, ENCODERS: A SUMMARY

Decoders map an n-bit signal to one of 2^n signals.

Encoders map one of 2^n signals to an n-bit signal.
Some encoders can only have one input line active.
Priority encoders can have several.
A priority encoder is a practical form of an encoder. The encoders available in IC form are all priority encoders.

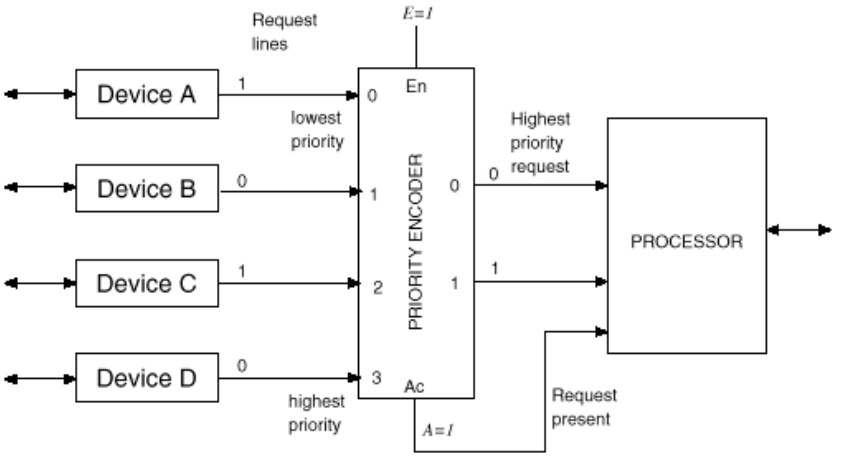
Most MSI modules have additional control I/O lines.
Many eight-bit MSI devices can be combined for wider words.

USES OF ENCODERS



Wind direction encoder

USES OF PRIORITY ENCODERS



Resolving interrupt requests using a priority encoder

CODE CONVERSION

Common examples

Binary-to-BCD
BCD-to-binary
Binary-to-Gray
Gray-to-binary
BCD-Excess3
Excess3-to-BCD
BCD-to-Aiken(4221)
Aiken-to-BCD
BCD(5421)-to-BCD
BCD-to-BCD(5421)

49

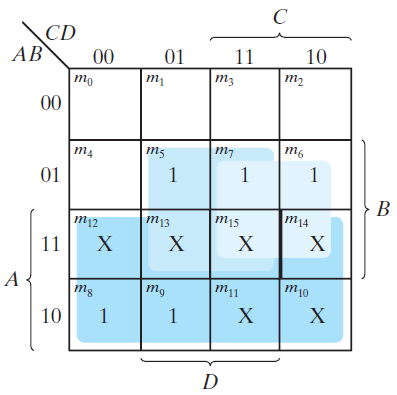
EXAMPLE: BCD-TO-EXCESS-3 CONVERSION

Truth Table for Code-Conversion Example

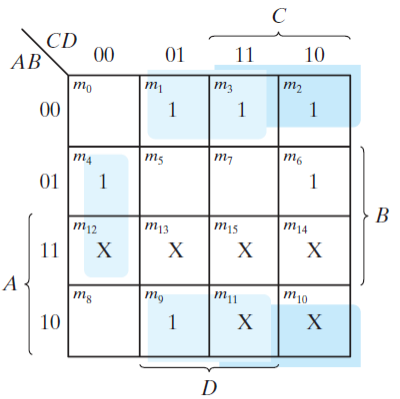
Input BCD				Output Excess-3 Code			
A	*B*	*C*	*D*	*w*	*x*	*y*	*z*
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

50

BCD-TO-EXCESS-3 CONVERSION



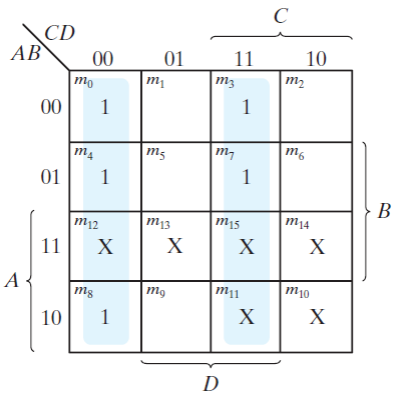
$w = A + BC + BD$



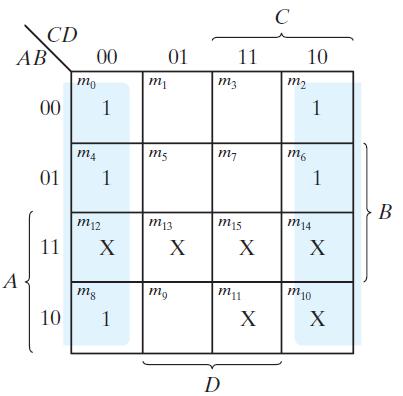
$x = B'C + B'D + BC'D'$

51

BCD-TO-EXCESS-3 CONVERSION



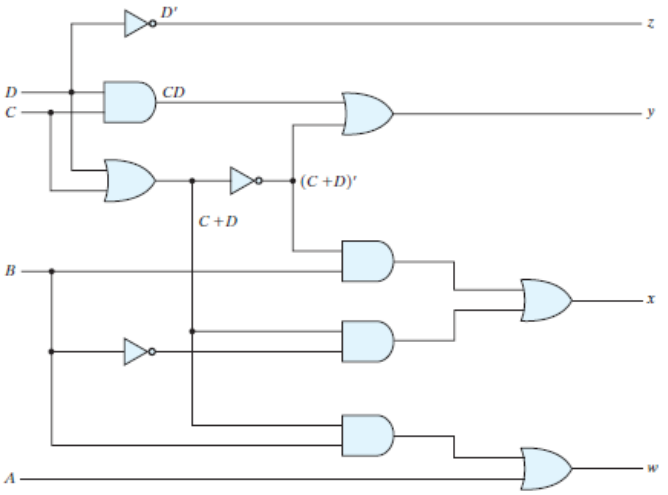
$y = CD + C'D'$



$z = D'$

52

BCD-TO-EXCESS-3 CONVERSION



A possible implementation

53

DECODERS AND ENCODERS IN PRACTICE

Dual 2-to-4 line decoder/demultiplexer **74HC/HCT139**

FEATURES

- Demultiplexing capability
- Two independent 2-to-4 decoders
- Multifunction capability
- Active LOW mutually exclusive outputs
- Output capability: standard
- I_{CC} category: MSI

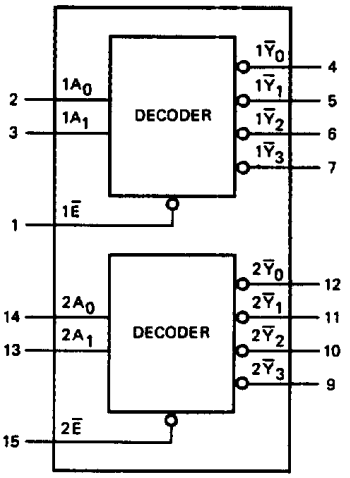
GENERAL DESCRIPTION

The 74HC/HCT139 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). It is specified in compliance with JEDEC standard no. 7A.

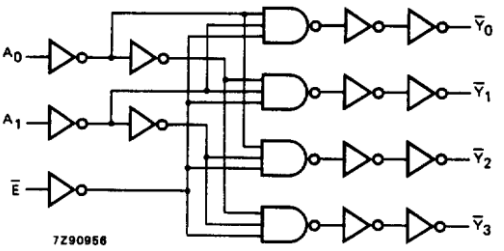
The 74HC/HCT139 are high-speed, dual 2-to-4 line decoder/multiplexers. This device has two independent decoders, each accepting two binary weighted inputs (nA_0 and nA_1) and providing four mutually exclusive active LOW outputs ($n\overline{Y}_0$ to $n\overline{Y}3$). Each decoder has an active LOW enable input ($n\overline{E}$).
When $n\overline{E}$ is HIGH, every output is forced HIGH. The enable can be used as the data input for a 1-to-4 demultiplexer application.
The "139" is identical to the HEF4556 of the HE4000B family.

54

DECODERS AND ENCODERS IN PRACTICE



Functional diagram (74HC/HCT139)



Logic diagram (one decoder)

Dual 2-to-4 line decoder

55

DECODERS AND ENCODERS IN PRACTICE

SN54147, SN54148, SN54LS147, SN54LS148
SN74147, SN74148 (TIM9907), SN74LS147, SN74LS148
10-LINE TO 4-LINE AND 8-LINE TO 3-LINE PRIORITY ENCODERS

SDLS053B – OCTOBER 1976 – REVISED MAY 2004

description/ordering information

These TTL encoders feature priority decoding of the inputs to ensure that only the highest-order data line is encoded. The '147 and 'LS147 devices encode nine data lines to four-line (8-4-2-1) BCD. The implied decimal zero condition requires no input condition, as zero is encoded when all nine data lines are at a high logic level. The '148 and 'LS148 devices encode eight data lines to three-line (4-2-1) binary (octal). Cascading circuitry (enable input EI and enable output EO) has been provided to allow octal expansion without the need for external circuitry. For all types, data inputs and outputs are active at the low logic level. All inputs are buffered to represent one normalized Series 54/74 or 54/74LS load, respectively.

56

DECODERS AND ENCODERS IN PRACTICE

FUNCTION TABLE – '147, 'LS147

INPUTS									OUTPUTS			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

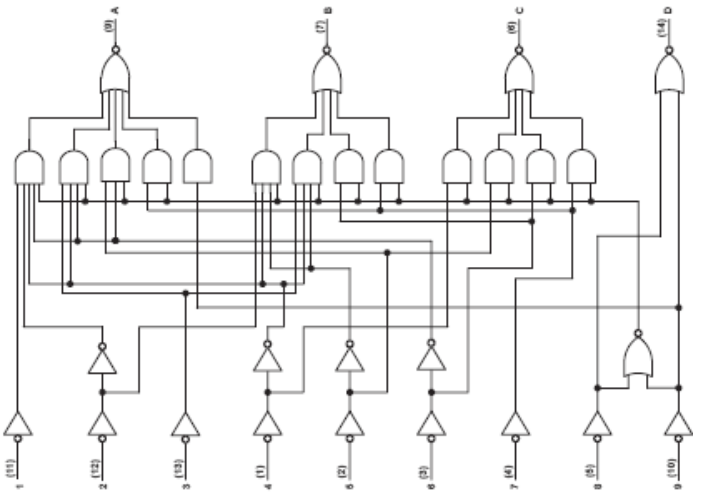
H = high logic level, L = low logic level, X = irrelevant

57

DECODERS AND ENCODERS IN PRACTICE

SN54147, SN54148, SN54LS147, SN54LS148
SN74147, SN74148 (TIM8907), SN74LS147, SN74LS148
10-LINE TO 4-LINE AND 8-LINE TO 3-LINE PRIORITY ENCODERS

'147, 'LS147 logic diagram (positive logic)



58

DECODERS AND ENCODERS IN PRACTICE

BCD to decimal decoder (1-of-10) **74HC/HCT42**

FEATURES

- Mutually exclusive outputs
- 1-of-8 demultiplexing capability
- Outputs disabled for input codes above nine
- Output capability: standard
- I_{CC} category: MSI

GENERAL DESCRIPTION

The 74HC/HCT42 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT42 decoders accept four active HIGH BCD inputs and provide 10 mutually exclusive active LOW outputs. The active LOW outputs facilitate addressing other MSI circuits with active LOW input enables.

The logic design of the "42" ensures that all outputs are HIGH when binary codes greater than nine are applied to the inputs.

The most significant input (A_3) produces an useful inhibit function when the "42" is used as a 1-of-8 decoder. The A_3 input can also be used as the data input in an 8-output demultiplexer application.

59

DECODERS AND ENCODERS IN PRACTICE

The 74HC/HCT139 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). It is specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT139 are high-speed, dual 2-to-4 line decoder/multiplexers. This device has two independent decoders, each accepting two binary weighted inputs (nA0 and nA1) and providing four mutually exclusive active LOW outputs (nY0 to nY3). Each unit has an active LOW enable input (nE). When nE is HIGH, every output is forced HIGH. The enable can be used as the data input for a 1-to-4 demultiplexer application.

The "139" is identical to the HEF4556 of the HE4000B family.

60

DECODERS AND ENCODERS IN PRACTICE

BCD to decimal decoder (1-of-10) 74HC/HCT42

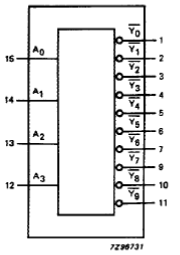


Fig.4 Functional diagram.

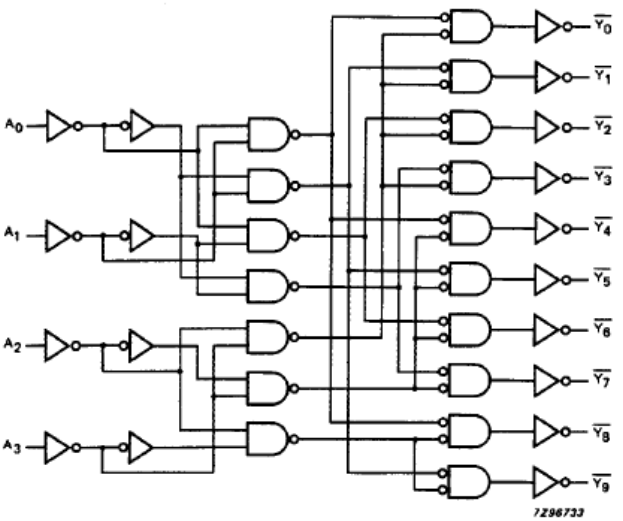
FUNCTION TABLE

INPUTS				OUTPUTS									
A_3	A_2	A_1	A_0	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$	$\overline{Y}_8$	$\overline{Y}_9$
L	L	L	L	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	H	H	L	H	H	H	H	H	H	H
L	L	H	H	H	H	H	L	H	H	H	H	H	H
L	H	L	L	H	H	H	H	L	H	H	H	H	H
L	H	L	H	H	H	H	H	H	L	H	H	H	H
L	H	H	L	H	H	H	H	H	H	L	H	H	H
L	H	H	H	H	H	H	H	H	H	H	L	H	H
H	L	L	L	H	H	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L
H	L	H	L	H	H	H	H	H	H	H	H	H	H
H	L	H	H	H	H	H	H	H	H	H	H	H	H
H	H	L	L	H	H	H	H	H	H	H	H	H	H
H	H	L	H	H	H	H	H	H	H	H	H	H	H
H	H	H	L	H	H	H	H	H	H	H	H	H	H
H	H	H	H	H	H	H	H	H	H	H	H	H	H

Note

1. H = HIGH voltage level
   L = LOW voltage level

74HC/HCT42 LOGIC DIAGRAM



BCD to decimal decoder (1-of-10)

62