Digital Circuits

ECS 371

Dr. Prapun Suksompong

prapun@siit.tu.ac.th

Lecture 11

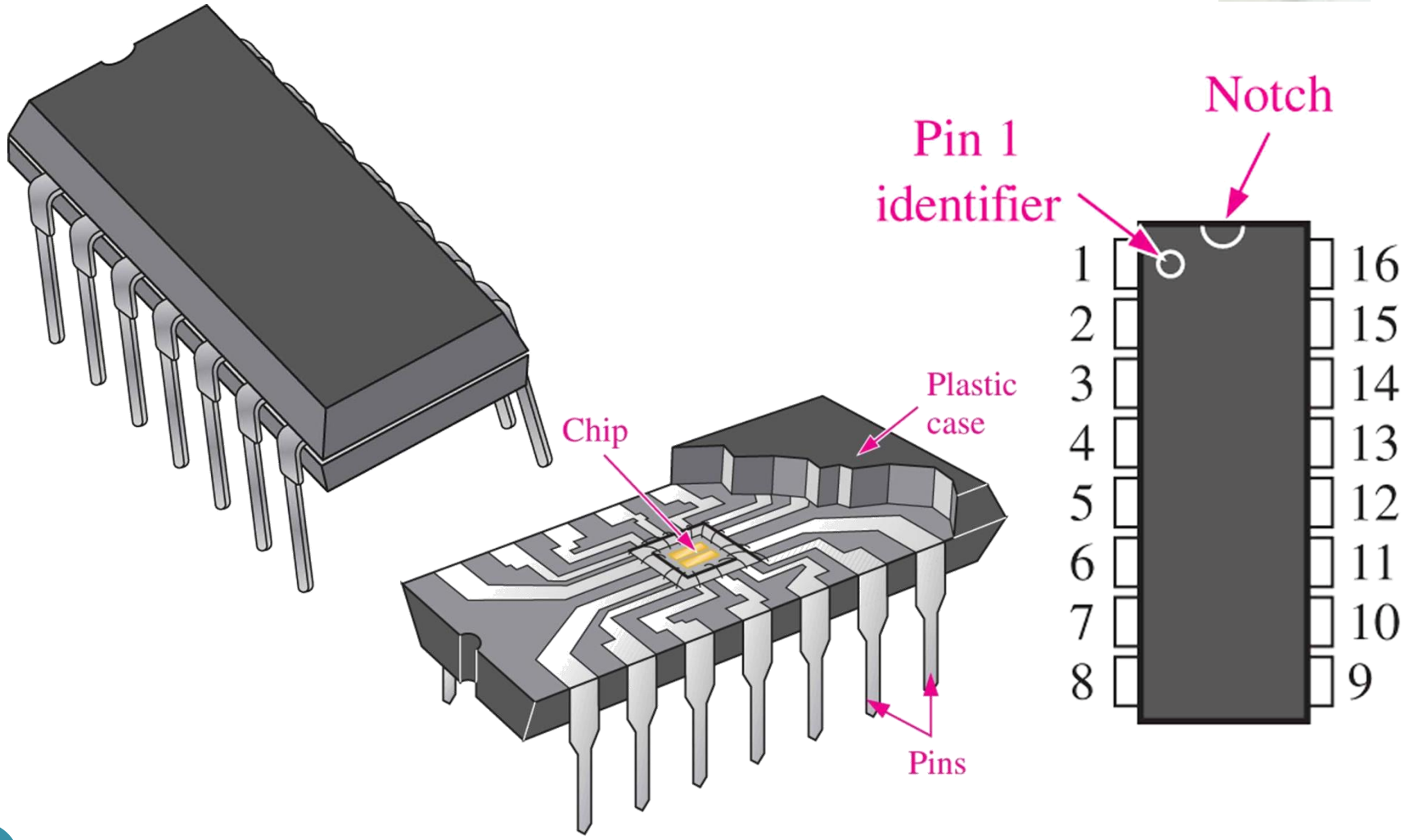
Office Hours:
BKD 3601-7
Monday 9:00-10:30, 1:30-3:30
Tuesday 10:30-11:30

ECS371.PRAPUN.COM

Announcement

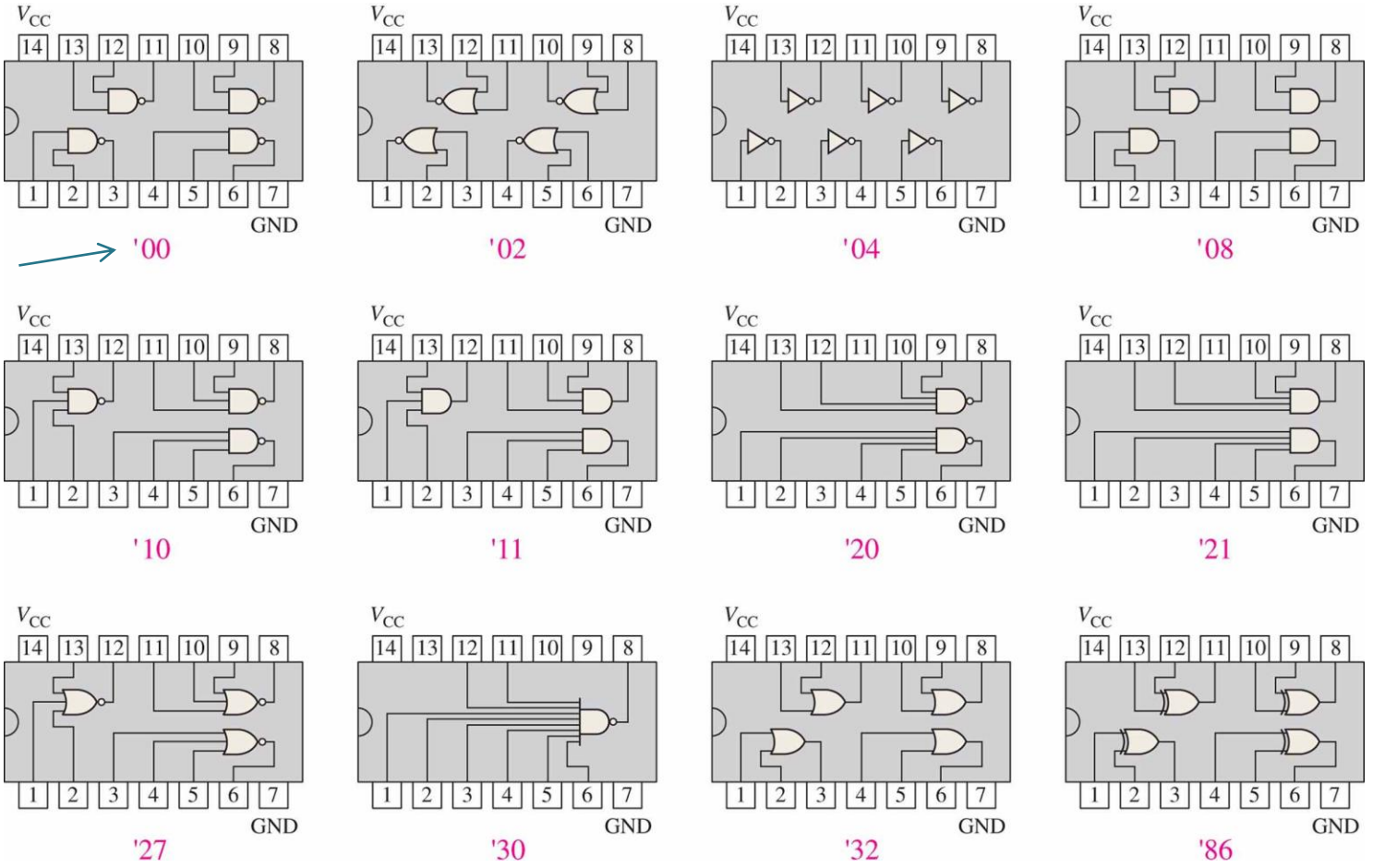
- HW4 posted on the course web site
  - Chapter 5: 4(b,c,e), 20a, 22a, 56
  - **Write down all the steps** that you have done to obtain your answers.
  - Due date: July 16, 2009 (Thursday)
- Reading Assignment
  - Chapter 6: 6-5, 6-8, 6-9

Fixed-function IC package

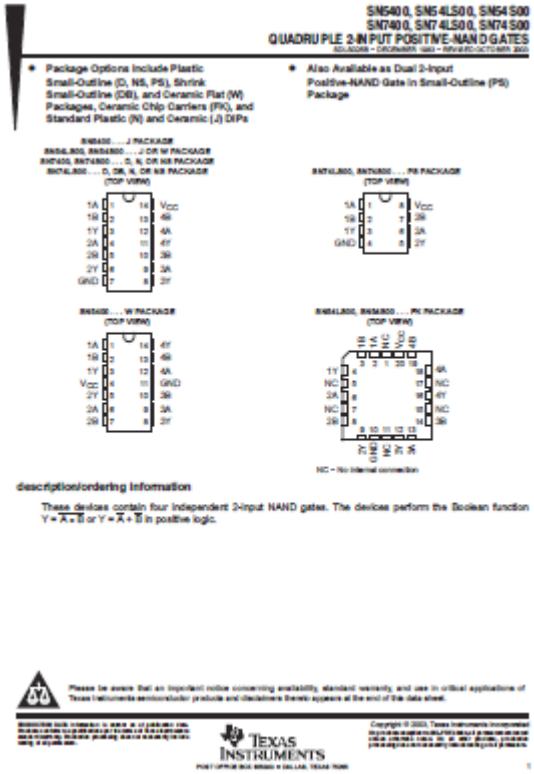


Cutaway view of DIP (Dual-In-line Pins) chip

SSI



Example of Data Sheet

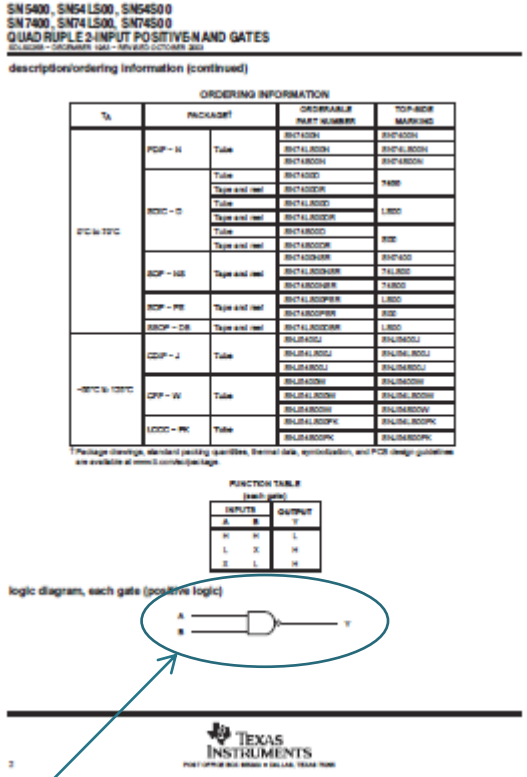


SN5400, SN54LS00, SN54S00
SN7400, SN74LS00, SN74S00
QUADRUPLE 2-INPUT POSITIVE-NAND GATES

- Package Options Include Plastic Small-Outline (D, NS, PS), Shrink Small-Outline (DB), and Ceramic Flat (W) Packages, Ceramic Chip Carriers (FK), and Standard Plastic (N) and Ceramic (J) DIPs
- Also Available as Dual 2-Input Positive-NAND Gate in Small-Outline (PS) Package

description/ordering information

These devices contain four independent 2-input NAND gates. The devices perform the Boolean function $Y = \overline{A \cdot B}$ or $Y = \overline{A} + \overline{B}$ in positive logic.



SN5400, SN54LS00, SN54S00
SN7400, SN74LS00, SN74S00
QUADRUPLE 2-INPUT POSITIVE-NAND GATES

description/ordering information (continued)

ORDERING INFORMATION

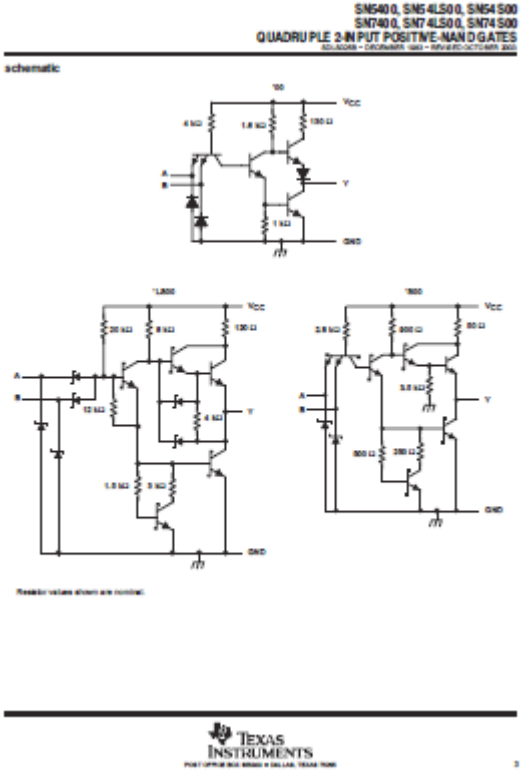
T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP – N	Tube	SN7400N	SN7400N
			SN74LS00N	SN74LS00N
			SN74S00N	SN74S00N
	SOIC – D	Tube	SN7400D	7400
		Tape and reel	SN7400DR	
		Tube	SN74LS00D	LS00
		Tape and reel	SN74LS00DR	
		Tube	SN74S00D	S00
		Tape and reel	SN74S00DR	
	SOP – NS	Tape and reel	SN7400NSR	SN7400
			SN74LS00NSR	74LS00
			SN74S00NSR	74S00
	SOP – PS	Tape and reel	SN74LS00PSR	LS00
			SN74S00PSR	S00
	SSOP – DB	Tape and reel	SN74LS00DBR	LS00
−55°C to 125°C	CDIP – J	Tube	SNJ5400J	SNJ5400J
			SNJ54LS00J	SNJ54LS00J
			SNJ54S00J	SNJ54S00J
	CFP – W	Tube	SNJ5400W	SNJ5400W
			SNJ54LS00W	SNJ54LS00W
			SNJ54S00W	SNJ54S00W
	LCCC – FK	Tube	SNJ54LS00FK	SNJ54LS00FK
			SNJ54S00FK	SNJ54S00FK

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

FUNCTION TABLE
(each gate)

INPUTS A	INPUTS B	OUTPUT Y
H	H	L
L	X	H
X	L	H

logic diagram, each gate (positive logic)

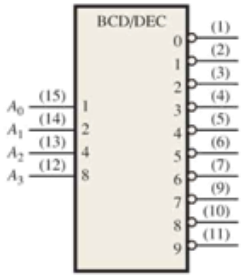


SN5400, SN54LS00, SN54S00
SN7400, SN74LS00, SN74S00
QUADRUPLE 2-INPUT POSITIVE-NAND GATES

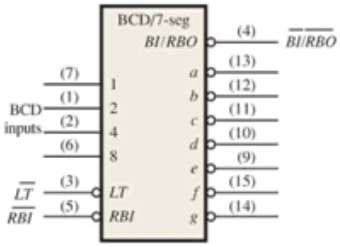
schematic

A NAND gate

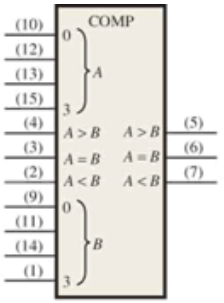
MSI



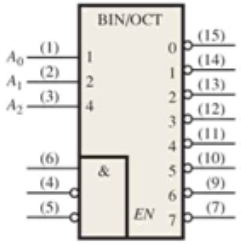
74HC42
BCD-to-decimal decoder



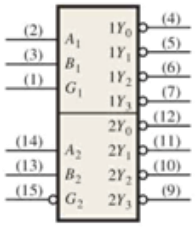
74LS47
BCD-to-7-segment decoder/driver



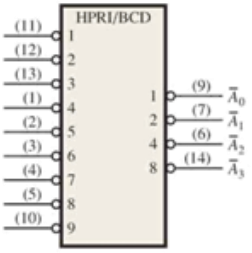
74LS85
4-bit magnitude comparator



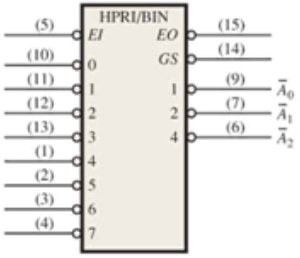
74LS138
3-line-to-8-line decoder



74LS139
Dual 2-line-to-4-line decoder

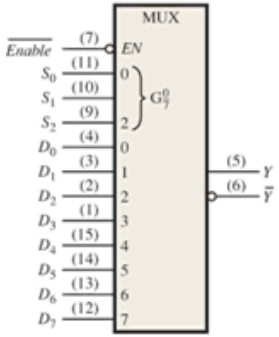


74HC147
Decimal-to-BCD priority encoder

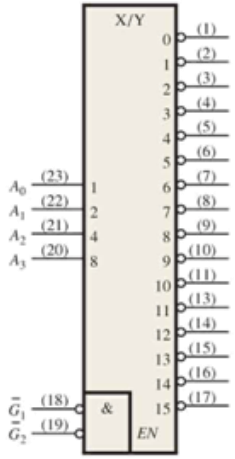


74LS148
Octal-to-binary encoder

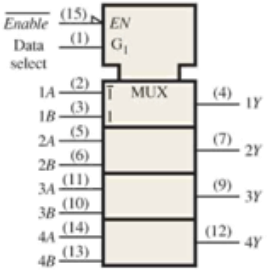
For the next couple lectures, we will study most of these 74-series MSI.



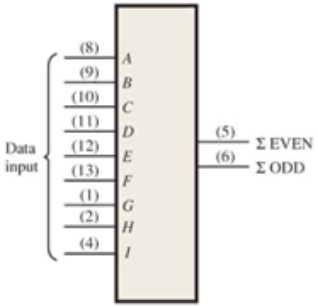
74LS151
8-input data selector/multiplexer



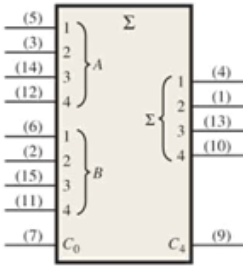
74HC154
1-of-16-line decoder



74HC157
Quad 2-input data selector/multiplexer

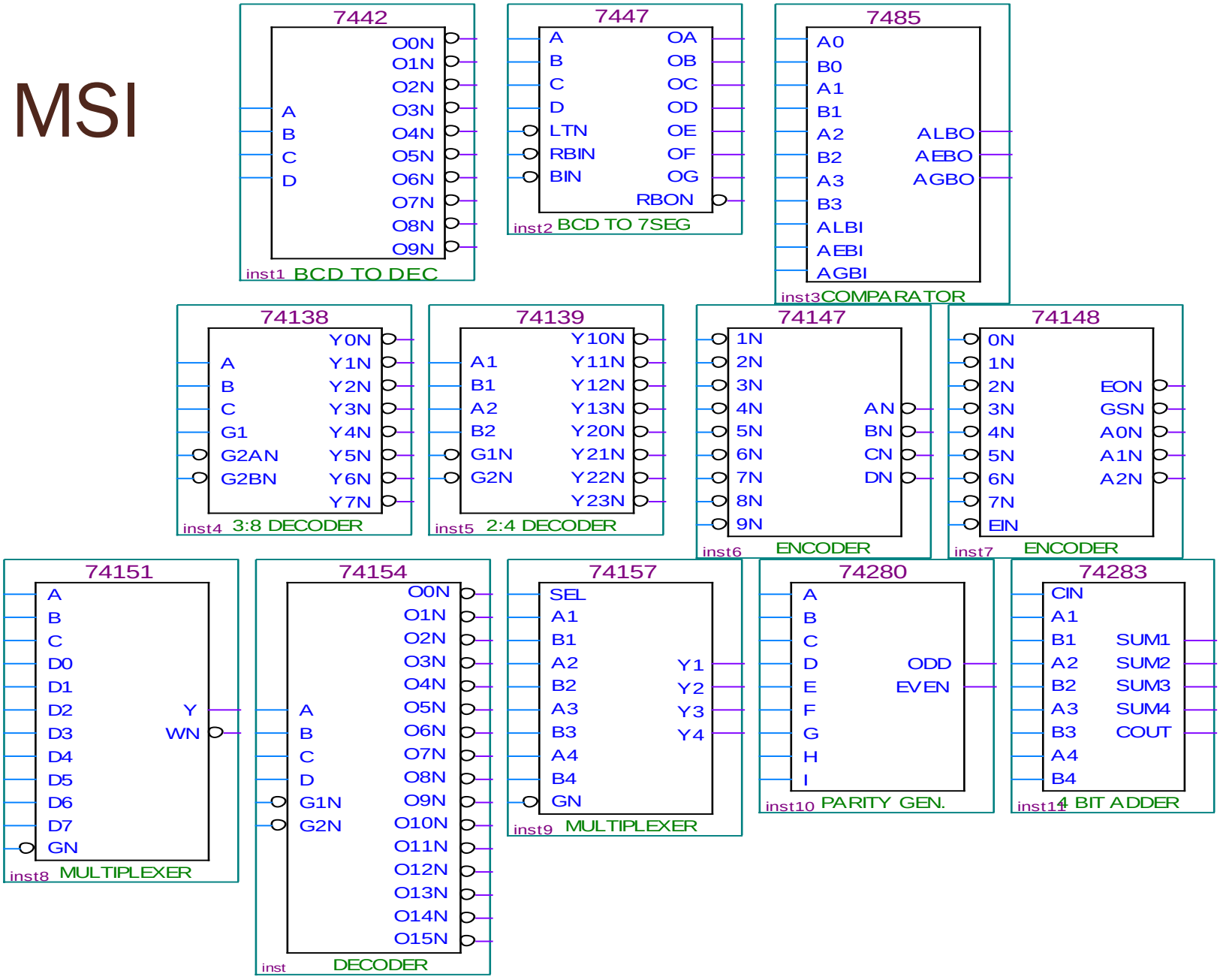


74LS280
9-bit parity generator/checker



74LS283
4-bit adder

MSI


7442
A
B
C
D
O0N
O1N
O2N
O3N
O4N
O5N
O6N
O7N
O8N
O9N
inst1 BCD TO DEC
7447
A
B
C
D
LTN
RBIN
BIN
OA
OB
OC
OD
OE
OF
OG
RBON
inst2 BCD TO 7SEG
7485
A0
B0
A1
B1
A2
B2
A3
B3
ALBI
AEBI
AGBI
ALBO
AEBO
AGBO
inst3 COMPARATOR
74138
A
B
C
G1
G2AN
G2BN
Y0N
Y1N
Y2N
Y3N
Y4N
Y5N
Y6N
Y7N
inst4 3:8 DECODER
74139
A1
B1
A2
B2
G1N
G2N
Y10N
Y11N
Y12N
Y13N
Y20N
Y21N
Y22N
Y23N
inst5 2:4 DECODER
74147
1N
2N
3N
4N
5N
6N
7N
8N
9N
AN
BN
CN
DN
inst6 ENCODER
74148
0N
1N
2N
3N
4N
5N
6N
7N
EIN
EON
GSN
A0N
A1N
A2N
inst7 ENCODER
74151
A
B
C
D0
D1
D2
D3
D4
D5
D6
D7
GN
Y
WN
inst8 MULTIPLEXER
74154
A
B
C
D
G1N
G2N
O0N
O1N
O2N
O3N
O4N
O5N
O6N
O7N
O8N
O9N
O10N
O11N
O12N
O13N
O14N
O15N
inst DECODER
74157
SEL
A1
B1
A2
B2
A3
B3
A4
B4
GN
Y1
Y2
Y3
Y4
inst9 MULTIPLEXER
74280
A
B
C
D
E
F
G
H
I
ODD
EVEN
inst10 PARITY GEN.
74283
CIN
A1
B1
A2
B2
A3
B3
A4
B4
SUM1
SUM2
SUM3
SUM4
COUT
inst11 4 BIT ADDER

Signals and Their Active Levels

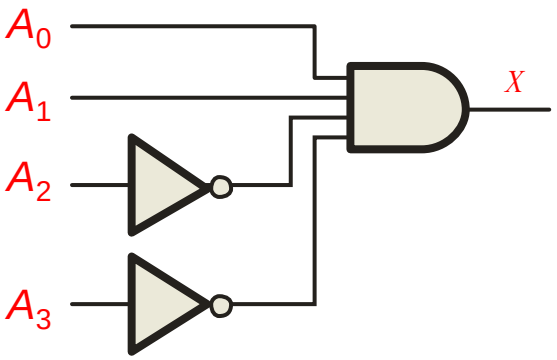
- Each input and output in a logic circuit should have a descriptive alpha-numeric label, the signal's name.
- A signal is **active high** if it performs the named action or denotes the named condition when it is HIGH (H) or 1.
- A signal is **active low** if it performs the named action or denotes the named condition when it is LOW (L) or 0.
- If not specified, assume active-high signal.
- A signal is said to be **asserted** when it is at its active level.
- A signal is said to be **negated** (or, sometimes, **deasserted**) when it is not at its active level.
- A **bus** is a collection of two or more related signal lines.

Simple Decoder

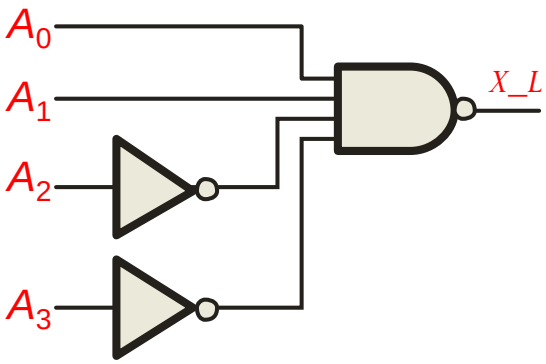
A **decoder** is a logic circuit that detects the presence of a specific combination of bits at its input.

Input				Output
A_3	A_2	A_1	A_0	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Two simple decoders that **detect the presence** of the binary code 0011 are shown below. The first has an active HIGH output; the second has an active LOW output.



Active-HIGH decoder for 0011



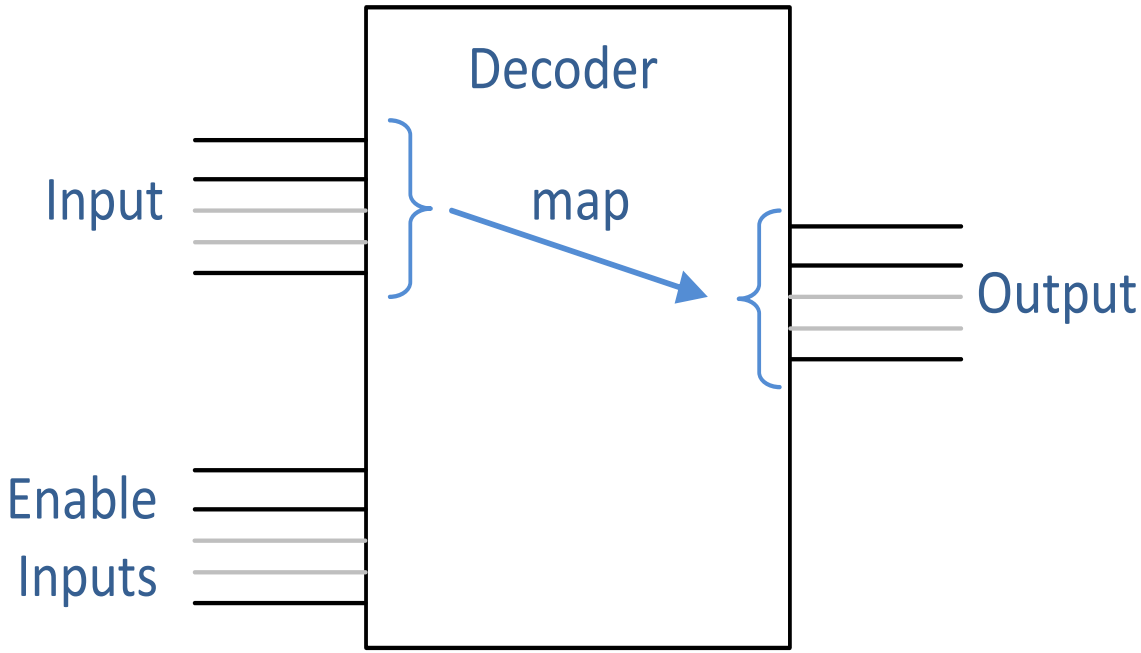
Active-LOW decoder for 0011

Input				Output
A_3	A_2	A_1	A_0	X_L
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

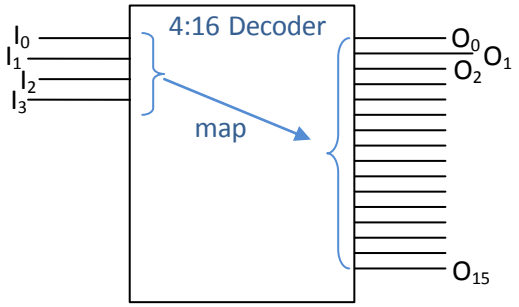
(A_0 is the LSB and A_3 is the MSB)

Binary-to-Decimal Decoder

- The input has n bits
- The output has 2^n bits. Only one bit is asserted at any time.
  - Also known as "1-out-of-m" (where $m = 2^n$)
- Zero or more EN (enable) lines



4:16 Decoder



Input				Output															
I_3	I_2	I_1	I_0	O_{15}	O_{14}	O_{13}	O_{12}	O_{11}	O_{10}	O_9	O_8	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_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	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

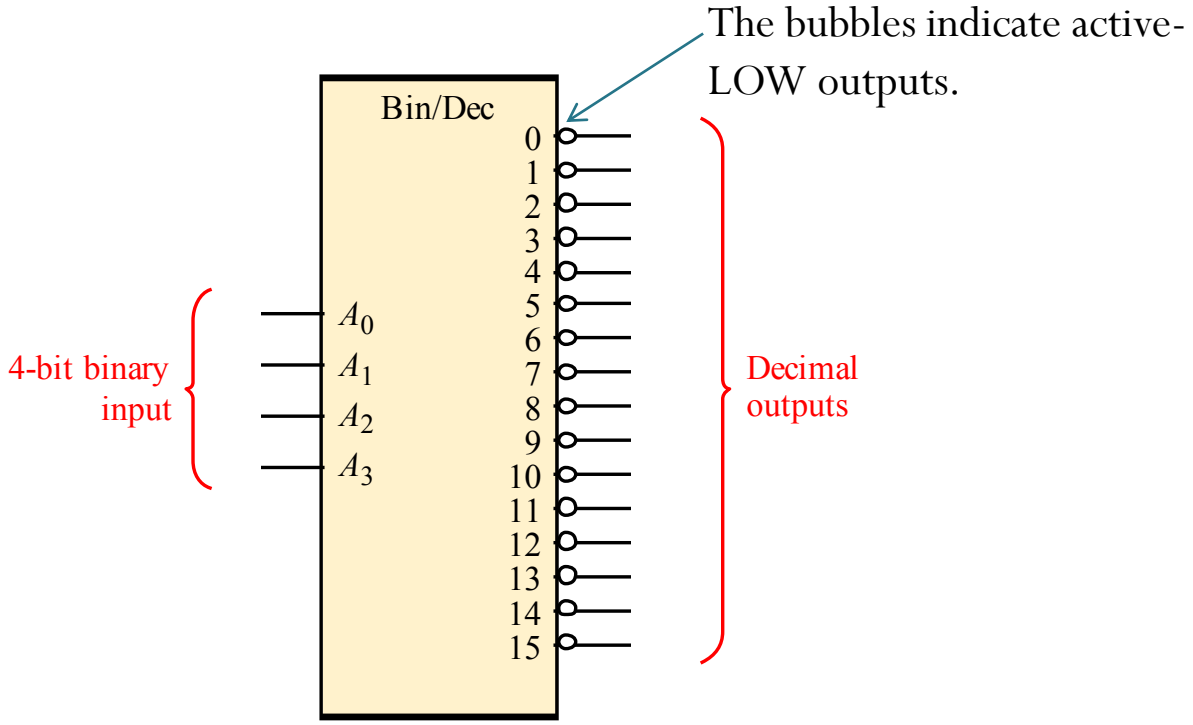
Note: All signals here are active-HIGH.

4:16 Decoder with Active-LOW outputs

Input				Output															
I_3	I_2	I_1	I_0	O_{15}_L	O_{14}_L	O_{13}_L	O_{12}_L	O_{11}_L	O_{10}_L	O_9_L	O_8_L	O_7_L	O_6_L	O_5_L	O_4_L	O_3_L	O_2_L	O_1_L	O_0_L
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Binary-to-Decimal Decoder

The binary-to-decimal decoder shown here has 16 outputs – one for each combination of binary inputs.



4:16 Decoder with EN

Input					Output														
EN	I_3	I_2	I_1	I_0	O_{15}	O_{14}	O_{13}	O_{12}	O_{11}	O_{10}	O_9	O_8	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: All signals here are active-HIGH.

4:16 Decoder

(Active-HIGH EN)
Active-LOW output

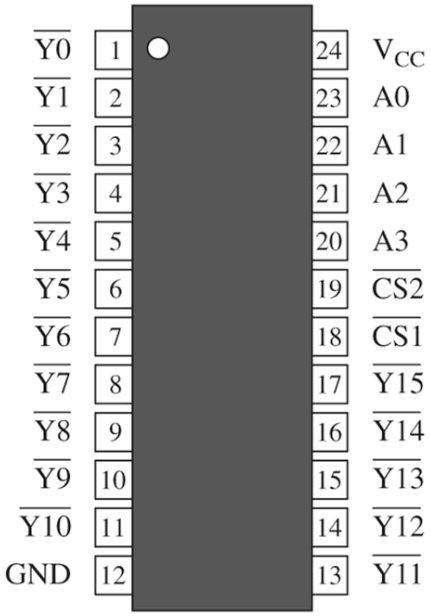
Input					Output														
EN	I_3	I_2	I_1	I_0	O_{15}_L	O_{14}_L	O_{13}_L	O_{12}_L	O_{11}_L	O_{10}_L	O_9_L	O_8_L	O_7_L	O_6_L	O_5_L	O_4_L	O_3_L	O_2_L	O_1_L	O_0_L
0	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	1	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
1	1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

4:16 Decoder

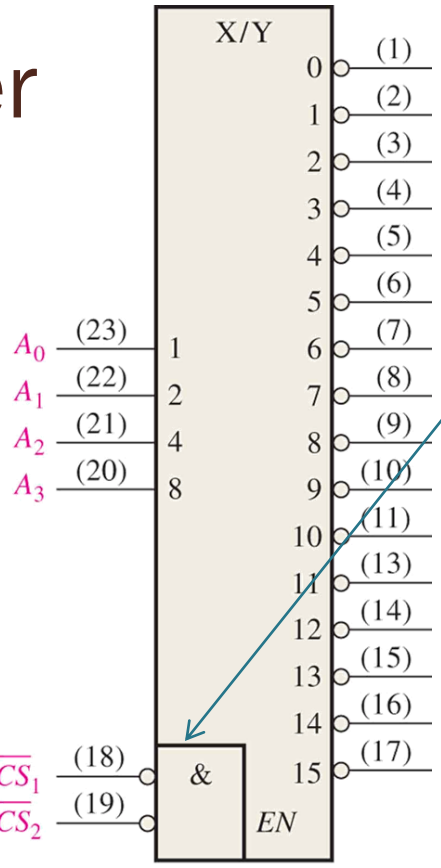
Active-LOW EN
Active-LOW output

Input					Output														
EN_L	I_3	I_2	I_1	I_0	O_{15}_L	O_{14}_L	O_{13}_L	O_{12}_L	O_{11}_L	O_{10}_L	O_9_L	O_8_L	O_7_L	O_6_L	O_5_L	O_4_L	O_3_L	O_2_L	O_1_L	O_0_L
1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0	1	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	1	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	1	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

74x154: 4:16 Decoder



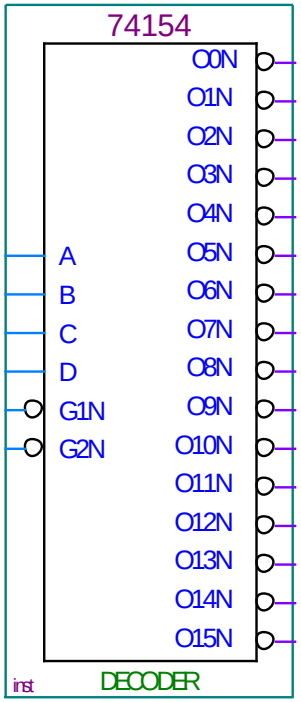
(a) Pin diagram



(b) Logic symbol

A LOW level on each chip select input is required to make the **enable** gate output (EN) HIGH.

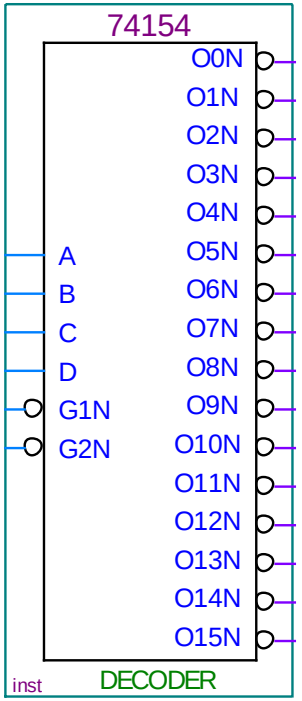
Alternative logic symbol



Include two **active-LOW** chip select (CS) lines which must be at the active level to enable the outputs. These lines can be used to expand the decoder to larger inputs.

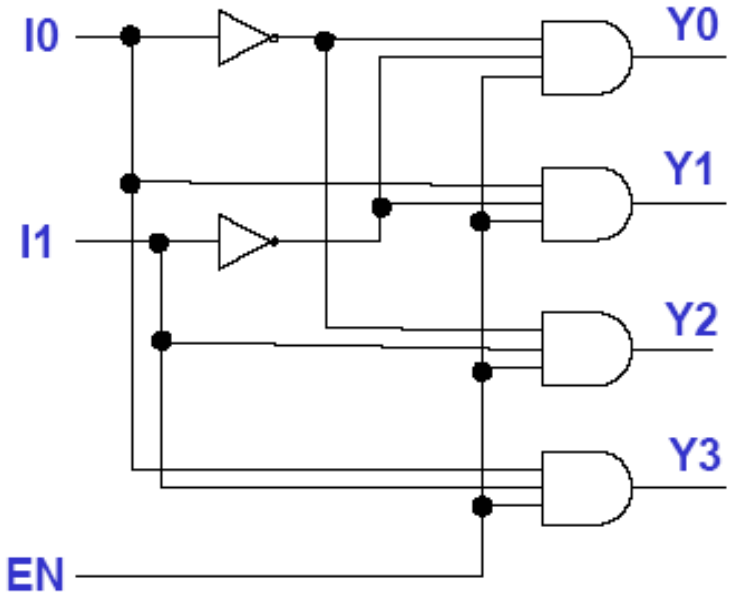
What do you need to know?

- On the exam/quiz/hw, when you see a logic symbol, you should be able to write down the following items on your own:
  - The description of its function.
  - The truth table.
  - The logic diagram.



Logic Diagram for 2:4 decoder

2-to-4 line decoder with enable input

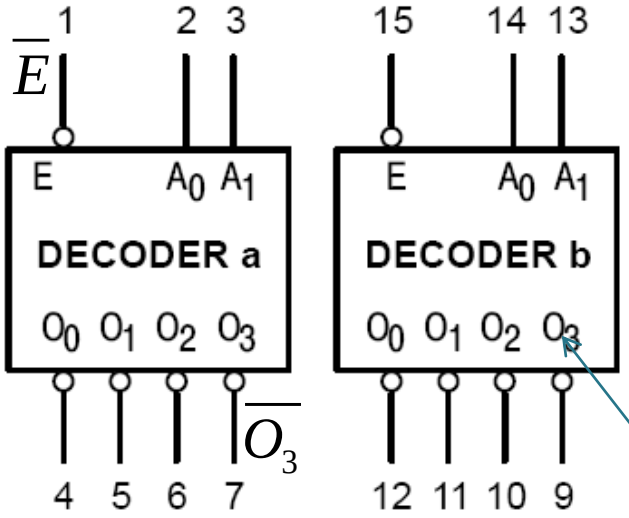


Inputs			Outputs			
EN	I1	I0	Y3	Y2	Y1	Y0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

74x139: Dual 2:4 Decoder

Most MSI decoders were originally designed with active-LOW output.

- Two independent 2:4 decoders
- The outputs and the enable (E) input are active-LOW.
- When E_L is HIGH all outputs are forced HIGH.

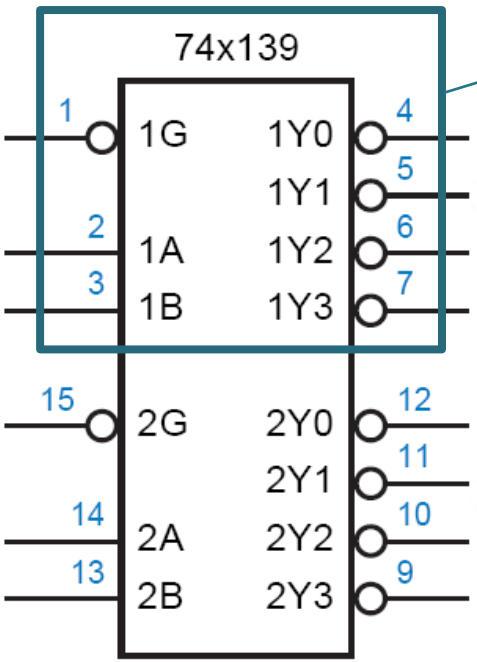


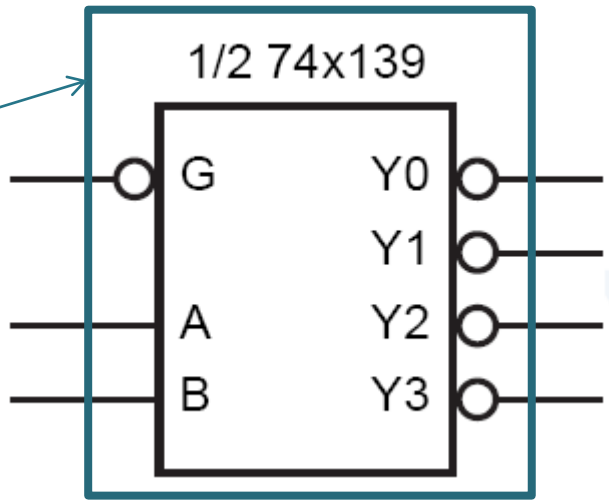
V_{CC} = PIN 16
GND = PIN 8

INPUTS			OUTPUTS			
$\overline{E}$	A_0	A_1	$\overline{O_0}$	$\overline{O_1}$	$\overline{O_2}$	$\overline{O_3}$
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	H	L	H	L	H	H
L	L	H	H	H	L	H
L	H	H	H	H	H	L

Notice that all of the signal names inside the symbol outline are active-HIGH, and that bubbles indicate active-LOW inputs and outputs.

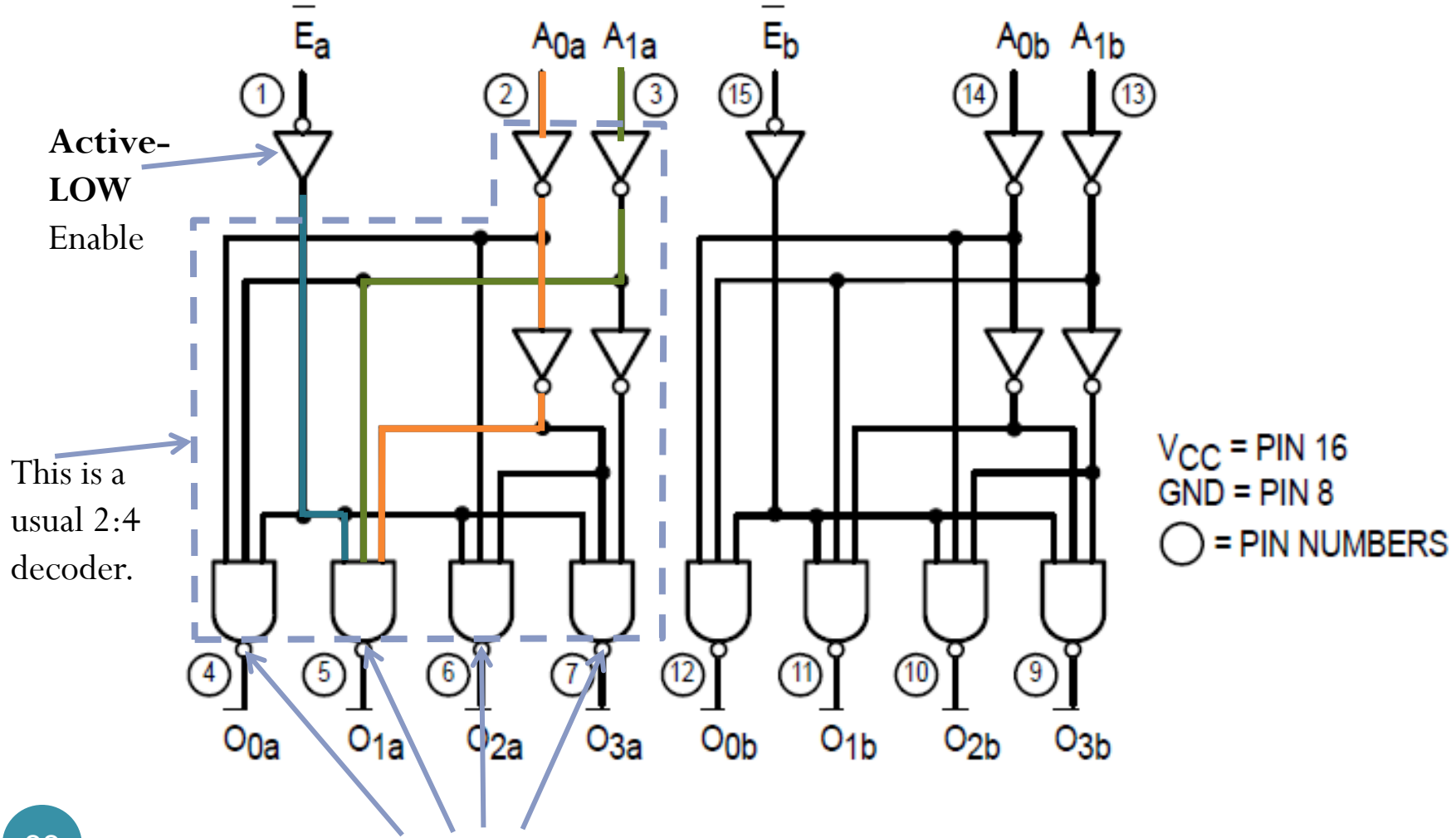
74x139





Inputs			*Outputs*			
G_L	B	A	Y3_L	Y2_L	Y1_L	Y0_L
1	x	x	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1

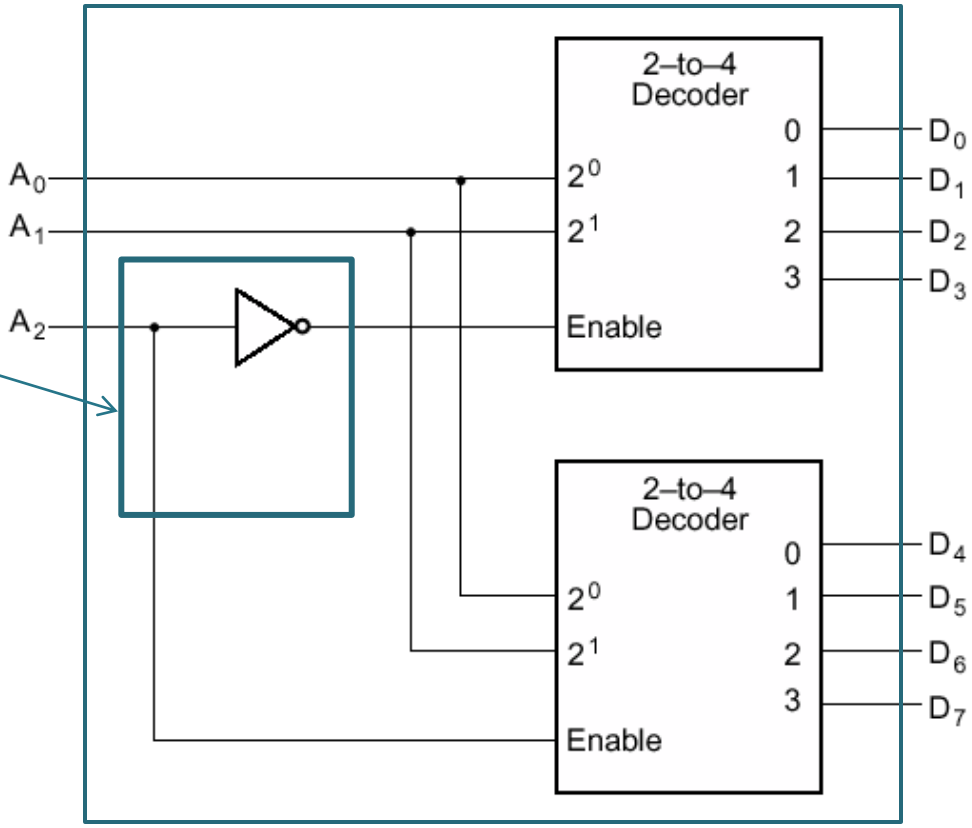
74x139: Logic diagram



Active-LOW output because NAND gates are used instead of AND gates

Example: Building a larger decoder

Construct a 3-to-8 decoder from two 2-to-4 decoders



Notice that this part is equivalent to a 1:2 decoder.

How can we add an active-HIGH enable input?

Low order bits (A_1, A_0) select within decoders. High order bit (A_2) controls which decoder is active.

Building larger decoder from smaller ones

- To construct $(k+n)$-to-2^{n+k} decoders, can use
  1. 2^n of k-to-2^k decoders with enable input and
  2. one n-to-$2n$ decoders.
- The connections are:
  - For each of the k-to-2^k decoder with enable input,
    - all have k input
    - we put in $A_0 \ldots A_{k-1}$.
  - The enable line of the r^{th} decoder is connected to D_r of the n-to-2^n decoders.
  - The inputs of the n-to-2^n decoder get A_k to A_{n+k-1}.
- Basically, each k-to-2^k decoder works on the last k bits.
- We use the first n bit, via the n-to-2^n decoder, to select which one (and only one) of the k-to-2^k decoders will be enabled.

Example

Construct a 4:16 decoder with an active-LOW enable from three 2:4 decoders.