

General Information

1

Data Transmission and Control Circuits

2

Display Drivers

3

Peripheral Drivers/Power Actuators

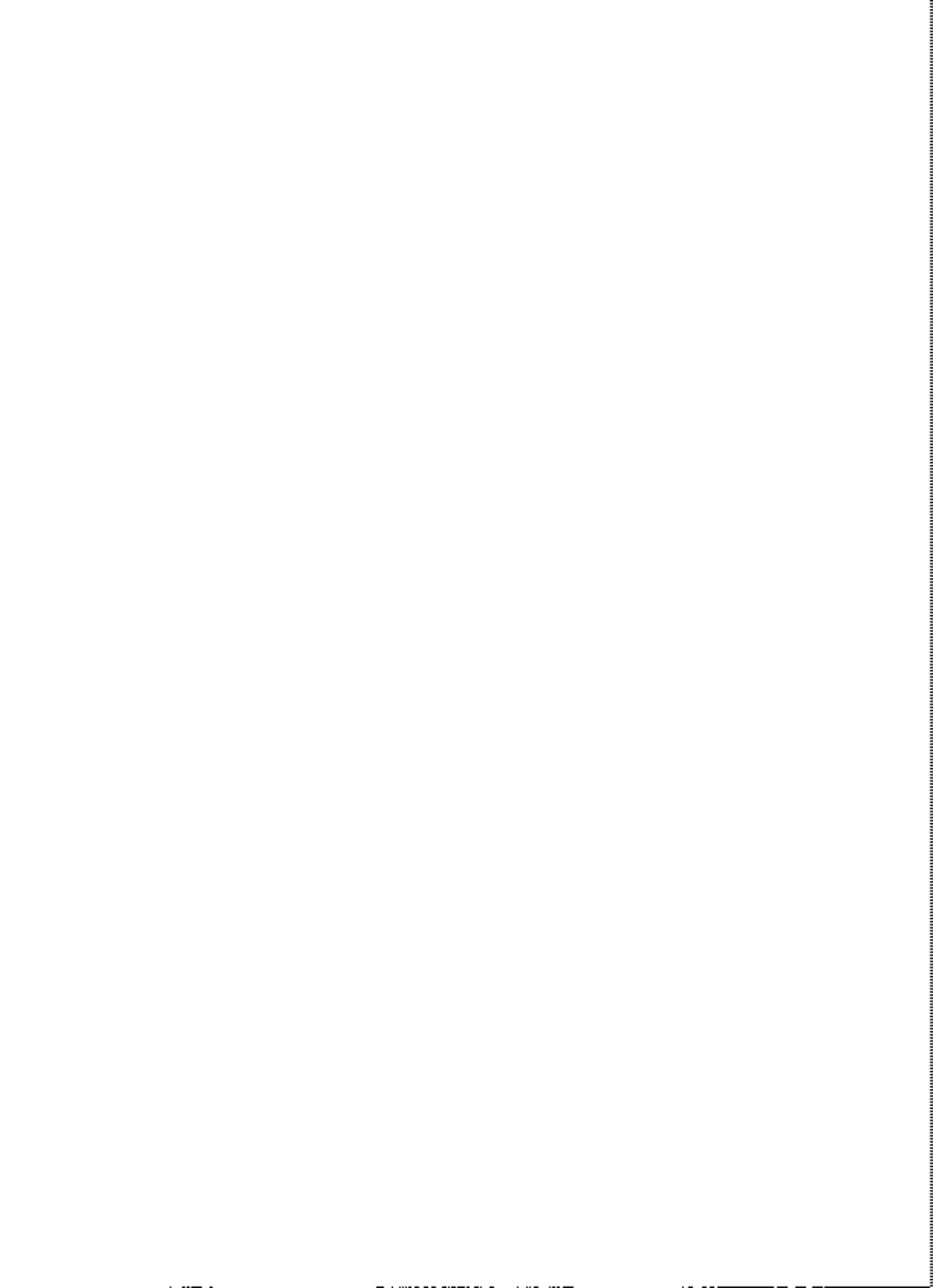
4

Mechanical Data

5

Explanation of Logic Symbols

6



Explanation of Logic Symbols

by F.A. Mann

Contents

<i>Section</i>		<i>Page</i>
1	Introduction	6-5
2	Symbol Composition	6-5
3	Qualifying Symbols	6-6
3.1	General Qualifying Symbols	6-6
3.2	Qualifying Symbols for Inputs and Outputs	6-6
3.3	Symbols Inside the Outline	6-7
3.4	Combinations of Outlines and Internal Connections	6-8
4	Dependency Notation	6-9
4.1	General Explanation	6-9
4.2	G, AND	6-10
4.3	Conventions for the Application of Dependency Notation in General	6-11
4.4	V, OR	6-11
4.5	N, Negate (Exclusive-OR)	6-12
4.6	Z, Interconnection	6-12
4.7	X, Transmission	6-13
4.8	C, Control	6-13
4.9	EN, Enable	6-14
4.10	M, Mode	6-14
5	Bistable Elements	6-16
6	Examples of Actual Device Symbols	6-17
6.1	SN75437A Quadruple Peripheral Driver	6-17
6.2	SN75128 8-Channel Line Receiver	6-17
6.3	SN75122 Triple Line Receivers	6-18
6.4	SN75113 Differential Line Drivers with Split 3-State Outputs	6-18
6.5	SN75163B Octal General-Purpose Interface Bus Transceiver	6-19
6.6	SN75161B Octal IEEE Std 488 Interface Bus Transceiver	6-20
6.7	SN75500E AC Plasma Display Driver with CMOS-Compatible Inputs	6-21
6.8	SN75551 Electroluminescent Row Driver with CMOS-Compatible Inputs	6-22

List of Tables

<i>Table</i>		<i>Page</i>
1	General Qualifying Symbols	6-6
2	Qualifying Symbols for Inputs and Outputs	6-7
3	Symbols Inside the Outline	6-8
4	Symbols for Internal Connections	6-9
5	Summary of Dependency Notation	6-16

List of Illustrations

<i>Figure</i>		<i>Page</i>
1	Symbol Composition	6-5
2	Common-Control Block	6-8
3	G Dependency Between Inputs	6-10
4	G Dependency Between Outputs and Inputs	6-10
5	G Dependency with a Dynamic Input	6-10
6	ORed Affecting Inputs	6-11
7	V (OR) Dependency	6-11
8	N (Negate/Exclusive-OR) Dependency	6-12
9	Z (Interconnection) Dependency	6-12
10	X (Transmission) Dependency	6-13
11	Analog Data Selector (Multiplexer/Demultiplexer)	6-13
12	C (Control) Dependency	6-14
13	EN (Enable) Dependency	6-14
14	M (Mode) Dependency Affecting Inputs	6-15
15	Type of Output Determined by Mode	6-15
16	Latches and Flip-Flops	6-16

If you have questions on this Explanation of Logic Symbols, please contact:

F.A. Mann, MS 3684
 Texas Instruments Incorporated
 P.O. Box 655303
 Dallas, Texas 75265
 Telephone (214) 997-2489

IEEE Standards may be purchased from:

Institute of Electrical and Electronics Engineers, Inc.
 IEEE Standards Office
 345 East 47th Street
 New York, N.Y. 10017

International Electrotechnical Commission (IEC) publications may be purchased from:

American National Standards Institute, Inc.
 1430 Broadway
 New York, N.Y. 10018

1 Introduction

The International Electrotechnical Commission (IEC) has been developing a very powerful symbolic language that can show the relationship of each input of a digital logic circuit to each output without showing explicitly the internal logic. At the heart of the system is dependency notation, which will be explained in Section 4.

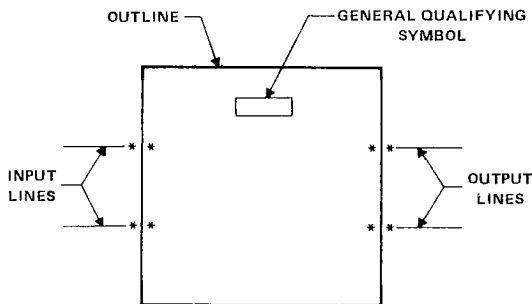
The system was introduced in the USA in a rudimentary form in IEEE/ANSI Standard Y32.14-1973. Lacking at that time a complete development of dependency notation, it offered little more than a substitution of rectangular shapes for the familiar distinctive shapes for representing the basic functions of AND, OR, negation, etc. This is no longer the case.

Internationally, Working Group 2 of IEC Technical Committee TC-3 has prepared a new document (Publication 617-12) that consolidates the original work started in the mid 1960's and published in 1972 (Publication 117-15) and the amendments and supplements that have followed. Similarly for the USA, IEEE Committee SCC 11.9 has revised the publication IEEE Std 91/ANSI Y32.14. Now numbered simply ANSI/IEEE Std 91-1984, the IEEE standard contains all of the IEC work that has been approved, and also a small amount of material still under international consideration. Texas Instruments is participating in the work of both organizations and this document introduces new logic symbols in accordance with the new standards. When changes are made as the standards develop, future editions will take those changes into account.

The following explanation of the new symbolic language is necessarily brief and greatly condensed from what the standards publications now contain. This is not intended to be sufficient for those people who will be developing symbols for new devices. It is primarily intended to make possible the understanding of the symbols used in this data book and is somewhat briefer than the explanation that appears in several of TI's data books on digital logic. However, it includes a new section (6.0) that explains several symbols for actual devices in detail. This has proven to be a powerful learning aid.

2 Symbol Composition

A symbol comprises an outline or a combination of outlines together with one or more qualifying symbols. The shape of the symbols is not significant. As shown in Figure 1, general qualifying symbols are used to tell exactly what logical operation is performed by the elements. Table 1 shows general qualifying symbols defined in the new standards. Input lines are placed on the left and output lines are placed on the right. When an exception is made to that convention, the direction of signal flow is indicated by an arrow as shown in Figure 9.



*Possible positions for qualifying symbols relating to inputs and outputs

Figure 1. Symbol Composition

3 Qualifying Symbols

3.1 General Qualifying Symbols

Table 1 shows general qualifying symbols defined by ANSI/IEEE Standard 91. These characters are placed near the top center or the geometric center of a symbol or symbol element to define the basic function of the device represented by the symbol or of the element.

X/Y is the general qualifying symbol for identifying coders, code converters, and level converters. X and Y may be used in their own right to stand for some code or either or both may be replaced by some other indication of the code or level such as BCD or TTL. As might be expected, interface circuits make frequent use of this set of qualifying symbols.

Table 1. General Qualifying Symbols



SYMBOL	DESCRIPTION
&	AND gate or function
≥ 1	OR gate or function. The symbol was chosen to indicate that at least one active input is needed to activate the output.
= 1	Exclusive OR. One and only one input must be active to activate the output.
1	A simple 1-input gate or element
\triangleright or \triangleleft	A buffer or element with more than usual output capability (symbol is oriented in the direction of signal flow).
\square	Schmitt trigger; element with hysteresis
X/Y	Coder, code converter, level converter
The following are examples of subsets of this general class of qualifying symbol used in this book.	
BCD/7-SEG	BCD to 7-segment display driver
TTL/MOS	TTL to MOS level converter
CMOS/PLASMA DISP	Plasma-display driver with CMOS-compatible inputs
MOS/LED	Light-emitting-diode driver with MOS-compatible inputs
CMOS/VAC FLUOR DISP	Vacuum-fluorescent display driver with CMOS-compatible inputs
CMOS/EL DISP	Electroluminescent display driver with CMOS-compatible inputs
TTL/GAS DISCH DISPLAY	Gas-discharge display driver with TTL-compatible inputs
SRG _m	Shift register. m = number of bits.

3.2 Qualifying Symbols for Inputs and Outputs

Qualifying symbols for inputs and outputs are shown in Table 2 and many will be familiar to most users, a likely exception being the logic polarity symbol for directly indicating active-low inputs and outputs. The older logic negation indicator means that the external 0 state produces the internal 1 state. The internal 1 state means the active state. Logic negation may be used in pure logic diagrams; in order to tie the external 1 and 0 logic states to the levels H (high) and L (low), a statement of whether positive logic (1 = H, 0 = L) or negative logic (1 = L, 0 = H) is being used is required or must be assumed. Logic polarity indicators eliminate the need for calling out the logic convention and are used in this data book in the symbology for actual devices. The presence of the triangle polarity indicator indicates that the L logic level will produce the internal 1 state (the active state) or that, in the case of an output, the internal 1 state will produce the external L level. Note how the active direction of transition for a dynamic input is indicated in positive logic, negative logic, and with polarity indication.

When nonstandardized information is shown inside an outline, it is usually enclosed in square brackets [like these]. The square brackets are omitted when associated with a nonlogic input, which is indicated by an X superimposed on the connection line outside the symbol.

Table 2. Qualifying Symbols for Inputs and Outputs

	Logic negation at input. External 0 produces internal 1.
	Logic negation at output. Internal 1 produces external 0.
	Active-low input. Equivalent to  in positive logic
	Active-low output. Equivalent to  in positive logic
	Active-low input in the case of right-to-left signal flow
	Active-low output in the case of right-to-left signal flow
	Signal flow from right to left. If not otherwise indicated, signal flow is from left to right.
	Bidirectional signal flow

		POSITIVE LOGIC	NEGATIVE LOGIC	POLARITY INDICATION
	} Dynamic inputs active on indicated transition }			not used
		not used	not used	

	Nonlogic connection. A label inside the symbol will usually define the nature of this pin.
	Input for analog signals (on a digital symbol) (see Figure 11)
	Input for digital signals (on an analog symbol) (see Figure 11)

3.3 Symbols Inside the Outline



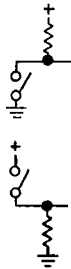








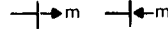


Table 3 shows some symbols used inside the outline. Note particularly that open-collector (open-drain), open-emitter (open-source), and three-state outputs have distinctive symbols. Also note that an EN input affects all of the outputs of the element and has no effect on inputs. An EN input affects all the external outputs of the element in which it is placed, plus the external outputs of any elements shown to be influenced by that element. It has no effect on inputs. When an enable input affects only certain outputs, affects outputs located outside the indicated influence of the element in which the enable input is placed, and/or affects one or more inputs, a form of dependency notation will indicate this (see 4.9). The effects of the EN input on the various types of outputs are shown.

It is particularly important to note that a D input is always the data input of a storage element. At its internal 1 state, the D input sets the storage element to its 1 state, and at its internal 0 state it resets the storage element to its 0 state.

The binary grouping symbol will be explained more fully in Section 6.11. Binary-weighted inputs are arranged in order and the binary weights of the least significant and the most significant lines are indicated by numbers. In this document weights of input and output lines will be represented by powers of two usually only when the binary grouping symbol is used, otherwise decimal numbers will be used. The grouped inputs generate an internal number on which a mathematical function can be performed or that can be an identifying number for dependency notation. This number is the sum of the weights (1, 2, 4, . . . 2ⁿ) of those input standing at their 1 states. A frequent use is in addresses for memories.

Reversed in direction, the binary grouping symbol can be used with outputs. The concept is analogous to that for the inputs and the weighted outputs will indicate the internal number assumed to be developed within the circuit.

Table 3. Symbols Inside the Outline

	Bistable input (input with hysteresis)	
	N-P-N open-collector or similar output that can supply a relatively low-impedance L level when not turned off. Requires external pull-up. Capable of positive-logic wired-AND connection.	
	Passive-pull-up output is similar to N-P-N open-collector output but is supplemented with a built-in passive pull-up.	
	N-P-N open-emitter or similar output that can supply a relatively low-impedance H level when not turned off. Requires external pull-down. Capable of positive-logic wired-OR connection.	
	Passive-pull-down output is similar to N-P-N open-emitter output but is supplemented with a built-in passive pull-down.	
	3-state output	
	Output with more than usual output capability (symbol is oriented in the direction of signal flow).	
	Enable input When at its internal 1-state, all outputs are enabled. When at its internal 0-state, open-collector, open-emitter outputs, and three-state outputs at external high-impedance state, and all other outputs (i.e., totem-poles) are at the internal 0-state.	
J, K, R, S, T	Usual meanings associated with flip-flops (e.g., R = reset, T = toggle)	
	Data input to a storage element equivalent to:	
	Shift right (left) inputs, m = 1, 2, 3, etc. If m = 1, it is usually not shown.	
	Binary grouping. m is highest power of 2. Produces a number equal to the sum of the weights of the active inputs	
	Input line grouping . . . indicates two or more terminals used to implement a single logic input. e.g., differential inputs.	

3.4 Combinations of Outlines and Internal Connections

When a circuit has one or more inputs that are common to more than one element of the circuit, the common-control block may be used. This is the only distinctively shaped outline used in the IEC system. Figure 2 shows that unless otherwise qualified by dependency notation, an input to the common-control block is an input to each of the elements below the common-control block.

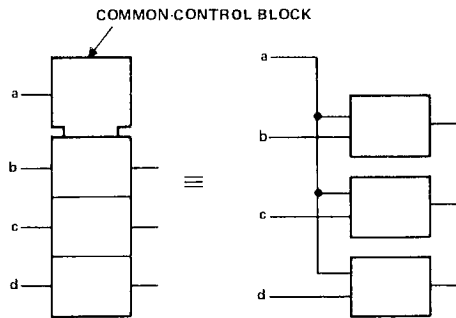


Figure 2. Common-Control Block

The outlines of elements may be embedded within one another or abutted to form complex elements, in which case the following rules apply. There is no logic connection between elements when the line common to their outlines is in the direction of signal flow. There is at least one logic connection when the line common to two outlines is perpendicular to the direction of signal flow. If no indications are shown on either side of the common line, it is assumed that there is only one logic connection. If more than one internal connection exists between adjacent elements, the number of connections will be clarified by the use of one or more of the internal connection symbols from Table 4 and/or appropriate qualifying symbols or dependency notation.

Table 4. Symbols for Internal Connections





	Internal connection. 1 state on left produces 1 state on right.
	Negated internal connection. 1 state on left produces 0 state on right.
	Dynamic internal connection. Transition from 0 to 1 on left produces transitory 1 state on right.
	Dynamic internal connection. Transition from 1 to 0 on left produces transitory 1 state on right.

Table 4 shows symbols that are used to represent internal connection with specific characteristics. The first is a simple noninverting connection, the second is inverting, the third is dynamic. As with this symbol and an external input line, the transition from 0 to 1 on the left produces a momentary 1-state on the right. The fourth symbol is similar except that the active transition on the left is from 1 to 0.

Only logic states, not levels, exist inside symbols. The negation symbol () is used internally even when direct polarity indication () is used externally.

In an array of elements, if the same general qualifying symbol and the same qualifying symbols associated with inputs and outputs would appear inside each of the elements of the array, these qualifying symbols are usually shown only in the first element. This is done to reduce clutter and to save time in recognition. Similarly, large identical elements that are subdivided into smaller elements may each be represented by an unsubdivided outline. The SN75163B symbol (see 6.5) illustrates this principle.

4 Dependency Notation

Some readers will find it more to their liking to skip this section and proceed to the explanation of the symbols for a few actual devices in 6.0. Reference will be made there to various parts of this section as it is needed. If this procedure is followed, it is recommended that 5.0 be read after 6.0 and then all of 4.0 be reread.

4.1 General Explanation

Dependency notation is the powerful tool that sets the IEC symbols apart from previous systems and makes compact, meaningful, symbols possible. It provides the means of denoting the relationship between inputs, outputs, or inputs and outputs without actually showing all the elements and interconnections involved. The information provided by dependency notation supplements that provided by the qualifying symbols for an element's function.

In the convention for the dependency notation, use will be made of the terms "affecting" and "affected." In cases where it is not evident which inputs must be considered as being the affecting or the affected ones (e.g., if they stand in an AND relationship), the choice may be made in any convenient way.

So far, eleven types of dependency have been defined but only the eight used in this book are explained. They are listed below in the order in which they are presented and are summarized in Table 5 following 4.10.2.

Section	Dependency Type or Other Subject
4.2	G, AND
4.3	General Rules for Dependency Notation
4.4	V, OR
4.5	N, Negate (Exclusive-OR)
4.6	Z, Interconnection
4.7	X, Transmission
4.8	C, Control
4.9	EN, Enable
4.10	M, Mode

4.2 G (AND) Dependency

A common relationship between two signals is to have them ANDed together. This has traditionally been shown by explicitly drawing an AND gate with the signals connected to the inputs of the gate. The 1972 IEC publication and the 1973 IEEE/ANSI standard showed several ways to show this AND relationship using dependency notation. While ten other forms of dependency have since been defined, the ways to invoke AND dependency are now reduced to one.

In Figure 3 input **b** is ANDed with input **a** and the complement of **b** is ANDed with **c**. The letter **G** has been chosen to indicate AND relationships and is placed at input **b**, inside the symbol. A number considered appropriate by the symbol designer (1 has been used here) is placed after the letter **G** and also at each affected input. Note the bar over the 1 at input **c**.

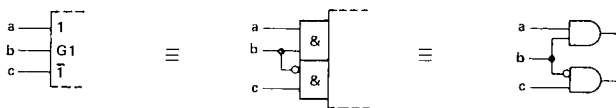


Figure 3. G Dependency Between Inputs

In Figure 4, output **b** affects input **a** with an AND relationship. The lower example shows that it is the internal logic state of **b**, unaffected by the negation sign, that is ANDed. Figure 5 shows input **a** to be ANDed with a dynamic input **b**.

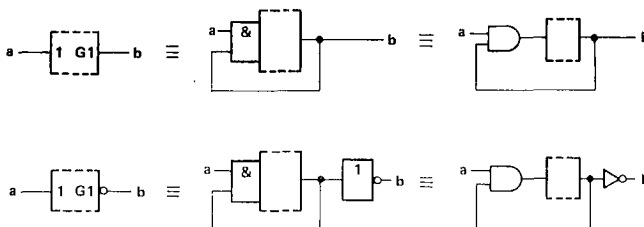


Figure 4. G Dependency Between Outputs and Inputs

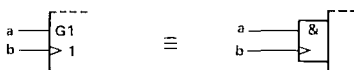


Figure 5. G Dependency with a Dynamic Input

The rules for G dependency can be summarized thus:

When a Gm input or output (m is a number) stands at its internal 1 state, all inputs and outputs affected by Gm stand at their normally defined internal logic states. When the Gm input or output stands at its 0 state, all inputs and outputs affected by Gm stand at their internal 0 states.

4.3 Conventions for the Application of Dependency Notation in General

The rules for applying dependency relationships in general follow the same pattern as was illustrated for G dependency.

Application of dependency notation is accomplished by:

1. Labeling the input (or output) *affecting* other inputs or outputs with the letter symbol indicating the relationship involved (e.g., G for AND) followed by an identifying number, appropriately chosen, and
2. Labeling each input or output *affected* by that affecting input (or output) with that same number.

If it is the complement of the internal logic state of the affecting input or output that does the affecting, then a bar is placed over the identifying numbers at the affected inputs or outputs (Figure 3).

If two affecting inputs or outputs have the same letter and same identifying number, they stand in an OR relationship to each other (Figure 6).

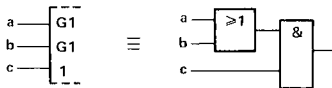


Figure 6. ORed Affecting Inputs

If the affected input or output requires a label to denote its function (e.g., "D'"), this label will be *prefixed* by the identifying number of the affecting input (Figure 12).

If an input or output is affected by more than one affecting input, the identifying numbers of each of the affecting inputs will appear in the label of the affected one, separated by commas. The normal reading order of these numbers is the same as the sequence of the affecting relationships (Figure 12).

4.4 V (OR) Dependency

The symbol denoting OR dependency is the letter V (Figure 7).

When a Vm input or output stands at its internal 1 state, all inputs and outputs affected by Vm stand at their internal 1 states. When the Vm input or output stands at its internal 0 state, all inputs and outputs affected by Vm stand at their normally defined internal logic states.

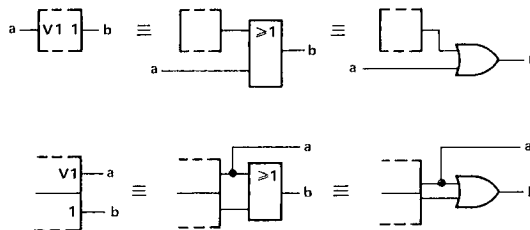


Figure 7. V (OR) Dependency

4.5 N (Negate) (Exclusive-OR) Dependency

The symbol denoting negate dependency is the letter N (Figure 8). Each input or output affected by an Nm input or output stands in an Exclusive-OR relationship with the Nm input or output.

When an Nm input or output stands at its internal 1 state, the internal logic state of each input and each output affected by Nm is the complement of what it would otherwise be. When an Nm input or output stands at its internal 0 state, all inputs and outputs affected by Nm stand at their normally defined internal logic states.

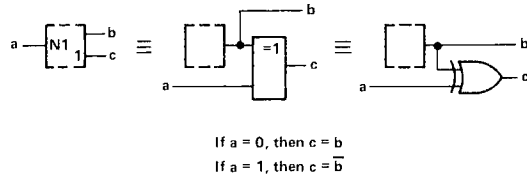


Figure 8. N (Negate) (Exclusive-OR) Dependency

4.6 Z (Interconnection) Dependency

The symbol denoting interconnection dependency is the letter Z.

Interconnection dependency is used to indicate the existence of internal logic connections between inputs, outputs, internal inputs, and/or internal outputs.

The internal logic state of an input or output affected by a Zm input or output will be the same as the internal logic state of the Zm input or output, unless modified by additional dependency notation (Figure 9).

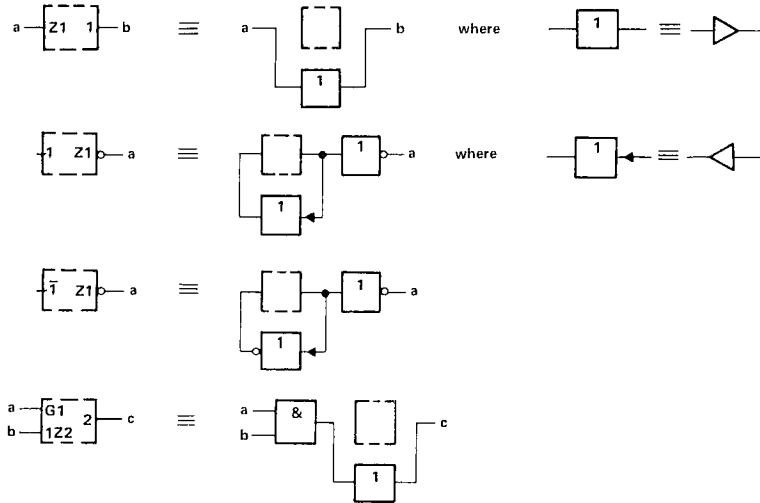


Figure 9. Z (Interconnection) Dependency

4.7 X (Transmission) Dependency

The symbol denoting transmission dependency is the letter X.

Transmission dependency is used to indicate controlled bidirectional connections between affected input/output ports (Figure 10).

When an Xm input or output stands at its internal 1 state, all input-output ports affected by this Xm input or output are bidirectionally connected together and stand at the same internal logic state or analog signal level. When an Xm input or output stands at its internal 0 state, the connection associated with this set of dependency notation does not exist.

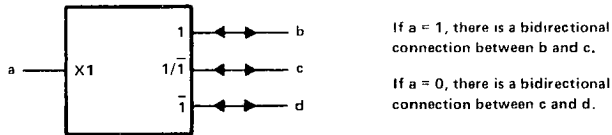


Figure 10. X (Transmission) Dependency

Although the transmission paths represented by X dependency are inherently bidirectional, use is not always made of this property. This is analogous to a piece of wire, which may be constrained to carry current in only one direction. If this is the case in a particular application, then the directional arrows shown in Figures 10 and 11 would be omitted.

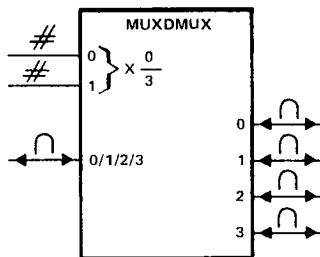


Figure 11. Analog Data Selector (Multiplexer/Demultiplexer)

4.8 C (Control) Dependency

The symbol denoting control dependency is the letter C.

Control inputs are usually used to enable or disable the data (D, J, K, R, or S) inputs of storage elements. They may take on their internal 1 states (be active) either statically or dynamically. In the latter case the dynamic input symbol is used as shown in the second example of Figure 12.

When a Cm input or output stands at its internal 1 state, the inputs affected by Cm have their normally defined effect on the function of the element, i.e., these inputs are enabled. When a Cm input or output stands at its internal 0 state, the inputs affected by Cm are disabled and have no effect on the function of the element.

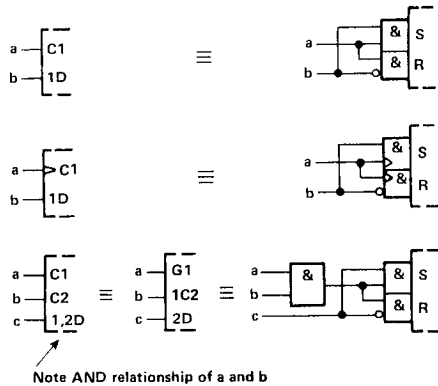


Figure 12. C (Control) Dependency

4.9 EN (Enable) Dependency

The symbol denoting enable dependency is the combination of letters EN.

An EN_m input has the same effect on outputs as an EN input, see 3.3, but it affects only those outputs labeled with the identifying number m . It also affects those inputs labeled with the identifying number m . By contrast, an EN input affects all outputs and no inputs. The effect of an EN_m input on an affected input is identical to that of a C_m input (Figure 13).

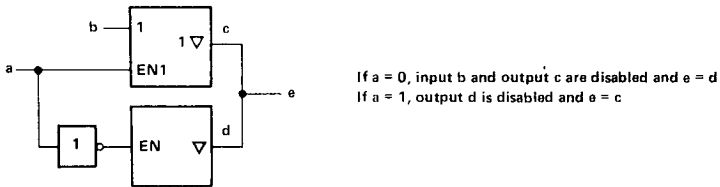


Figure 13. EN (Enable) Dependency

When an EN_m input stands at its internal 1 state, the inputs affected by EN_m have their normally defined effect on the function of the element and the outputs affected by this input stand at their normally defined internal logic states, i.e., these inputs and outputs are enabled.

When an EN_m input stands at its internal 0 state, the inputs affected by EN_m are disabled and have no effect on the function of the element, and the outputs affected by EN_m are also disabled. Open-collector outputs are turned off, three-state outputs stand at their high-impedance state, and all other outputs (e.g., totem-pole outputs) stand at their internal 0 states.

4.10 M (MODE) Dependency

The symbol denoting mode dependency is the letter M.

Mode dependency is used to indicate that the effects of particular inputs and outputs of an element depend on the mode in which the element is operating.

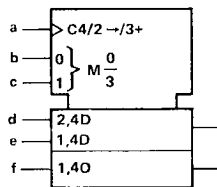
If an input or output has the same effect in different modes of operation, the identifying numbers of the relevant affecting M_m inputs will appear in the label of that affected input or output between parentheses and separated by solidi, e.g., $(1/2)CT=0 \equiv 1CT=0/2CT=0$ where 1 and 2 refer to M1 and M2.

4.10.1 M Dependency Affecting Inputs

M dependency affects inputs the same as C dependency. When an *Mm* input or *Mm* output stands at its internal 1 state, the inputs affected by this *Mm* input or *Mm* output have their normally defined effect on the function of the element, i.e., the inputs are enabled.

When an *Mm* input or *Mm* output stands at its internal 0 state, the inputs affected by this *Mm* input or *Mm* output have no effect on the function of the element. When an affected input has several sets of labels separated by solidi (e.g., C4/2→/3+), any set in which the identifying number of the *Mm* input or *Mm* output appears has no effect and is to be ignored. This represents disabling of some of the functions of a multifunction input.

The circuit in Figure 14 has two inputs, *b* and *c*, that control which one of four modes (0, 1, 2, or 3) will exist at any time. Inputs *d*, *e*, and *f* are D inputs subject to dynamic control (clocking) by the *a* input. The numbers 1 and 2 are in the series chosen to indicate the modes so inputs *e* and *f* are only enabled in mode 1 (for parallel loading) and input *d* is only enabled in mode 2 (for serial loading). Note that input *a* has three functions. It is the clock for entering data. In mode 2, it causes right shifting of data, which means a shift away from the control block. In mode 3, it causes the contents of the register to be incremented by one count.



Note that all operations are synchronous.

In MODE 0 ($b = 0, c = 0$), the outputs remain at their existing states as none of the inputs has an effect.

In MODE 1 ($b = 1, c = 0$), parallel loading takes place thru inputs *e* and *f*.

In MODE 2 ($b = 0, c = 1$), shifting down and serial loading thru input *d* take place.

In MODE 3 ($b = c = 1$), counting up by increment of 1 per clock pulse takes place.

Figure 14. M (Mode) Dependency Affecting Inputs

4.10.2 M Dependency Affecting Outputs

When an *Mm* input or *Mm* output stands at its internal 1 state, the affected outputs stand at their normally defined internal logic states, i.e., the outputs are enabled.

When an *Mm* input or *Mm* output stands at its internal 0 state, at each affected output any set of labels containing the identifying number of that *Mm* input or *Mm* output has no effect and is to be ignored. When an output has several different sets of labels separated by solidi (e.g., 2,4/3,5), only those sets in which the identifying number of this *Mm* input or *Mm* output appears are to be ignored.

Figure 15 shows a symbol for a device whose output can behave like either a 3-state output or an open-collector output depending on the signal applied to input *a*. Mode 1 exists when input *a* stands at its internal 1 state and, in that case, the three-state symbol applies and the open-element symbol has no effect. When $a = 0$, mode 1 does not exist so the three-state symbol has no effect and the open-element symbol applies.

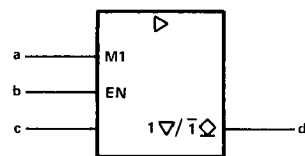


Figure 15. Type of Output Determined by Mode

Table 5. Summary of Dependency Notation

TYPE OF DEPENDENCY	LETTER SYMBOL*	AFFECTING INPUT AT ITS 1-STATE	AFFECTING INPUT AT ITS 0-STATE
Control	C	Permits action	Prevents action
Enable	EN	Permits action	Prevents action of inputs ◇ outputs turned off ▽ outputs at external high impedance Other outputs at internal 0 state
Mode	G	Permits action	Imposes 0 state
Mode	M	Permits action (mode selected)	Prevents action (mode not selected)
Negate (Ex-NOR)	N	Complements state	No effect
OR	V	Imposes 1 state	Permits action
Transmission	X	Bidirectional connection exists	Bidirectional connection does not exist
Interconnection	Z	Imposes 1 state	Imposes 0 state

* These letter symbols appear at the AFFECTING input (or output) and are followed by a number. Each input (or output) AFFECTED by that input is labeled with that same number.

5 Bistable Elements

The dynamic input symbol and dependency notation provide the tools to identify different types of bistable elements and make synchronous and asynchronous inputs easily recognizable (Figure 16).

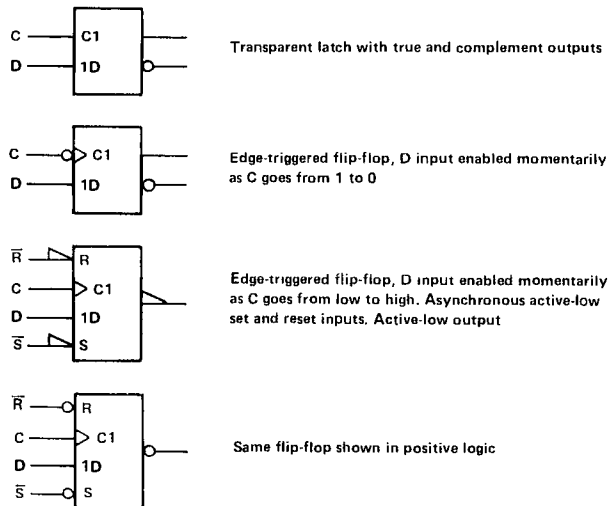


Figure 16. Latches and Flip-Flops

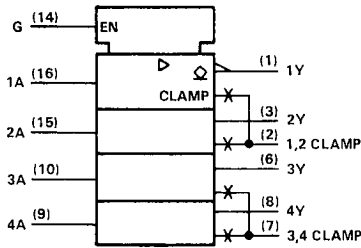
Transparent latches have a level-operated control input. The D input is active as long as the C input is at its internal 1 state. The outputs respond immediately. Edge-triggered elements accept data from D, J, K, R, or S inputs on the active transition of C.

Notice that synchronous inputs can be readily recognized by their dependency labels (a number preceding the functional label, 1D in these examples) compared to the asynchronous inputs (S and R), which are not dependent on the C inputs. Of course if the set and reset inputs were dependent on the C inputs, their labels would be similarly modified (e.g., 1S, 1R).

6 Examples of Actual Device Symbols

The symbols explained in this section include some of the most complex in this book. These were chosen, not to discourage the reader, but to illustrate the amount of information that can be conveyed. It is likely that if one reads these explanations and follows them reasonably well, most of the other symbols will seem simple indeed. The explanations are intended to be independent of each other so they may seem somewhat repetitious. However each illustrates new principles. They are arranged more or less in the order of complexity.

6.1 SN75437A Quadruple Peripheral Driver

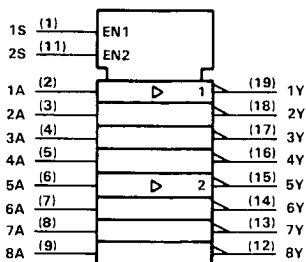


There are four identical sections. The symbology is complete for the first element; the absence of any symbology for the other elements indicates they are identical. The top two elements share a common output clamp, pin 2. This is shown to be a nonlogic connection by the superimposed X on the line. The function for this type of connection is indicated briefly and not necessarily exactly by a small amount of text within the symbol. The bottom two elements likewise share a common clamp.

Each element is shown to be an inverter with amplification (indicated by \triangleright). Taking TTL as a reference, this means that either the input is sensitive to lower level signals, or the output has greater drive capability than usual. The latter applies in this case. The output is shown by \triangleleft to be open collector.

All the outputs share a common EN input, pin 14. See Figure 2 for an explanation of the common control block. When EN = 0 (pin 14 is low), the outputs, being open-collector types, are turned off and would be pulled high by an external pullup resistor.

6.2 SN75128 8-Channel Line Receiver

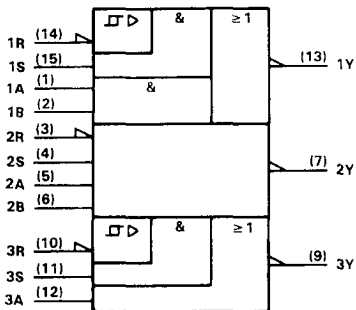


There are eight identical sections. The symbology is complete for the first element; the absence of any symbology for the next three elements indicates they are identical. Likewise the symbology is complete for the fifth element; the absence of any symbology for the next three elements indicates they are identical to the fifth.

Each element is shown to be an inverter with amplification (indicated by \triangleright). Taking TTL as a reference, this means that either the input is sensitive to lower level signals, or the output has greater drive capability than usual. The former applies in this case. Since neither the symbol for open-collector (\triangleleft) or 3-state (∇) outputs is shown, the outputs are of the totem-pole type.

The top four outputs are shown to be affected by affecting input number 1, which is EN1, meaning they will be enabled if EN1 = 1 (pin 1 is high). See 4.9 for an explanation of EN dependency. If pin 1 is low, EN1 = 0 and the affected outputs will go to their inactive (high) levels. Similarly, the lower four outputs are controlled by pin 11.

6.3 SN75122 Triple Line Receivers

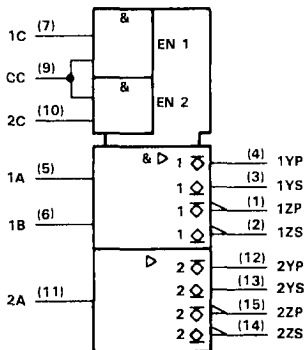


There are two identical sections. The symbology is complete for the first section; the absence of any symbology for the next section indicates it is identical. Likewise the symbology is complete for the third section, which is similar, but not identical, to the first and second.

The top section may be considered to be an OR element (≥ 1) with two embedded ANDs (&), one of which has an active-low amplified input (\triangleright) with hysteresis (\square), pin 14. This is ANDed with pin 15 and the result is ORed with the AND of pins 1 and 2. The output of the OR, pin 13, is active-low.

The third section is identical to the first except that pin 12 has no input ANDed with it. Since neither the symbol for open-collector (\diamond) or 3-state (∇) outputs is shown, the outputs are of the totem-pole type.

6.4 SN75113 Differential Line Drivers with Split 3-State Outputs



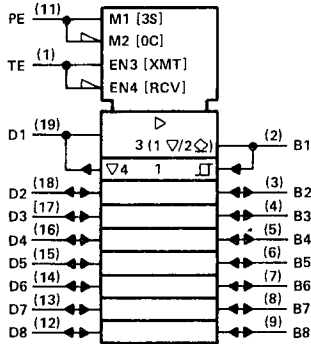
There are two similar elements in the array. The first is a 2-input AND element (indicated by &); the second has only a single input. Both elements are shown to have special amplification (indicated by \triangleright). Taking TTL as a reference, this means that either the input is sensitive to lower level signals, or the output has greater drive capability than usual. The latter applies in this case.

Each element has four outputs. Pins 4 and 3 are a pair consisting of one open-emitter output (\diamond) and one open-collector output (\square). Relative to the AND function, both are active high. Pins 1 and 2 are a similar pair but relative to the AND function, both are active low. All outputs of a single, unsubdivided element always have identical internal logic states determined by the function of the element except when otherwise indicated by an associated symbol or label inside the element. Here there is no such contrary indication. All four outputs are shown to be affected by affecting input number 1, which is EN1,

meaning they will all be enabled if $EN1 = 1$. See 4.9 for an explanation of EN dependency. If $EN1 = 0$, all the affected outputs will be turned off. EN1 is the output of an AND gate (indicated by &) whose active-high inputs are pins 7 and 9. Both pins 7 and 9 must be high to enable the outputs of the top element. Assuming they are enabled and that pins 5 and 6 are both high, the internal state of all four outputs will be a 1. Pins 4 and 3 will both be high, pins 1 and 2 will both be low. The part is designed so that pins 3 and 4 may be connected together creating an active-high 3-state output. Likewise pins 1 and 2 may be connected together to create an active-low 3-state output.

All that has been said about the first element regarding its outputs and their enable inputs also applies to the second element. Pins 9 and 10 are the enable inputs in this case.

6.5 SN75163B Octal General-Purpose Interface Bus Transceiver



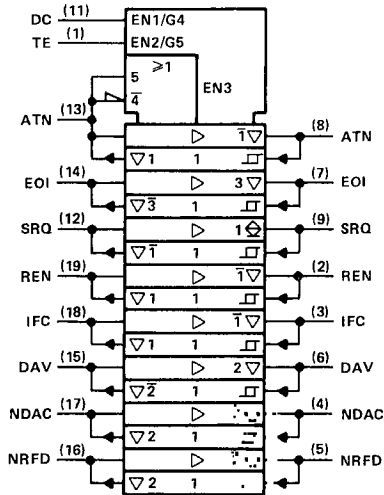
There are eight I/O ports on each side, pins 2 through 9 and 12 through 19. There are eight identical channels. The symbology is complete for the first channel; the absence of any symbology for the other channels indicates they are identical. The eight bidirectional channels each have amplification from left to right, that is, the outputs on the right have increased drive capability (indicated by \triangleright), and the inputs on the right all have hysteresis (indicated by \square).

The outputs on the left are shown to be 3-state outputs by the ∇ . They are also shown to be affected by affecting input number 4, which is EN4, meaning they will be enabled if EN4 = 1 (pin 1 is low). See 4.9 for an explanation of EN dependency. If EN4 = 0 (pin 1 is high), the affected outputs will go to their high-impedance (off) states.

The labeling at pin 2, which applies to all the outputs on the right, is unusual because the outputs themselves have an unusual feature. The label includes both the symbol for a 3-state output (∇) and for an open-collector output (\circ), separated by a slash indicating that these are alternatives.

The symbol for the 3-state output is shown to be affected by affecting input number 1, which is M1, meaning the ∇ label is valid when M1 = 1 (pin 11 is high), but is to be ignored when M1 = 0 (pin 11 is low). See 4.10 for an explanation of M (mode) dependency. Likewise the symbol for the open-collector output is shown to be affected by affecting input number 2, which is M2, meaning the \circ label is valid when M2 = 1 (pin 11 is low), but is to be ignored when M2 = 0 (pin 11 is high). These labels are enclosed in parentheses (used as in algebra); the numeral 3 indicates that in either case the output is affected by EN3. Thus the right-hand outputs will be off if pin 1 is low. It can now be seen that pin 1 is the direction control and pin 11 is used to determine whether the outputs are of the 3-state or open-collector variety.

6.6 SN75161B Octal IEEE Std 488 Interface Bus Transceiver



There are eight I/O ports on each side, pins 2 through 9 and 12 through 19. Pin 13 is not only an I/O port; the line running into the common-control block (see Figure 2) indicates that it also has control functions. Pins 1 and 11 are also controls. The eight bidirectional channels each have amplification from left to right, that is, the outputs on the right have increased drive capability (indicated by \triangleright), and the inputs on the right all have hysteresis (indicated by \square). All of the outputs are shown to be of the 3-state type by the ∇ symbol except for the outputs at pins 9, 4, and 5, which are shown to have passive pullups by the \oplus symbol.

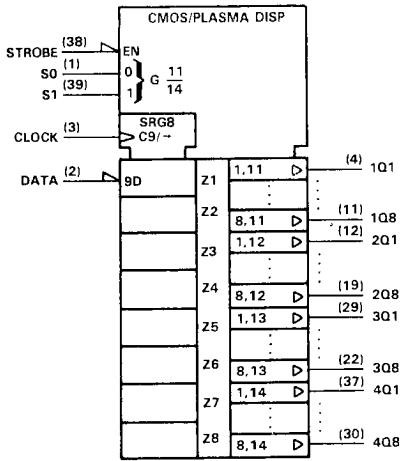
Starting with a typical I/O port, pin 18, the output portion is identified by an arrow indicating right-to-left signal flow and the three-state output symbol (∇). This output is shown to be affected by affecting input number 1, which is EN1, meaning it will be enabled as an output if EN1 = 1 (pin 11 is high). See 4.9 for an explanation of EN dependency. If pin 11 is low, EN1 = 0 and the output at pin 18

will be in its high-impedance (off) state. This also applies to the 3-state outputs at pins 13 and 19 and to the passive-pullup output at pin 9. On the other hand, the outputs at pins 8, 2, 3, and 12 all are affected by the complement of EN1. This is indicated by the bar over the 1 at each of those outputs. They are enabled only when pin 11 is low. Thus one function of pin 11 is to serve as direction control for the first, third, fourth, and fifth channels.

Similarly it can be seen that pin 1 serves as direction control for the sixth, seventh, and eighth channels. If pin 1 is high, transmission will be from left to right in the sixth channel, right to left in the seventh and eighth. These transmissions are reversed if pin 1 is low.

The direction control for the second channel, EN3, is more complex. EN3 is the output of an OR (≥ 1) function. One of the inputs to this OR is the active-high signal on pin 13. This signal is shown to be affected at the input to the OR gate by affecting input number 5, which is G5, meaning that pin 13 is ANDed with pin 1 before entering the OR gate. See 4.2 for an explanation of G (AND) dependency. The other input to the OR is the active-low signal on pin 13. This signal is ANDed with the complement of pin 11 before entering the OR gate. This is indicated by the G4 at pin 1 and the 4 with a bar over it at pin 13. Thus for EN3 to stand at the 1 state, which would enable transmission from pin 14 to pin 7, both pins 13 and 1 must be high or both pins 13 and 11 must be low.

6.7 SN75500E AC Plasma Display Driver with CMOS-Compatible Inputs



the internal labels and the pin numbers are both consecutive. Thus it should be clear that the input of the element whose output is pin 5 is affected by affecting input number 2, just as the input of the element whose output is pin 4 is affected by affecting input number 1. Affecting inputs 1 through 8 are Z inputs (Z1 through Z8), which means their signals are transferred directly to the output elements. See 4.6 for an explanation of Z dependency.

The inputs of the 32 implicitly shown output elements are also shown to be affected by affecting inputs numbers 11, 12, 13, and 14 in four blocks of eight each. These inputs will be found in the common control block preceded by a letter G and a brace. The brace is called the binary grouping symbol. It is equivalent to a decoder with outputs in this case driving four G inputs (G11, G12, G13, and G14). The weights of the inputs to the coder are shown to be 2^0 and 2^1 for pins 1 and 39, respectively. The decoder has four outputs corresponding to the four possible sums of the weights of the activated decoder inputs. If pins 1 and 39 are both low, the sum of the weights = 0 and G11 = 1. If pin 1 is low while pin 39 is high, the sum = 2 and G13 = 1 and so forth. G indicates AND dependency, see 4.2. Only one of the four affecting G inputs at a time can take on the 1 state. The block of eight output elements affected by that G input are enabled; the 0 state is imposed on the other 24 output elements and externally those output pins are low.

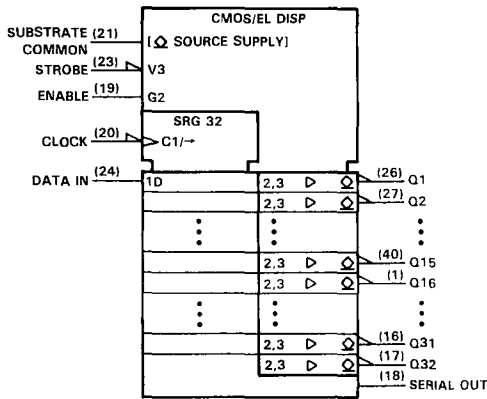
Because of their high-current, high-voltage characteristics, the outputs are labeled with the amplification symbol \triangleright . All the outputs share a common EN input, pin 38. See Figure 2 for an explanation of the common control block. When EN = 0 (pin 38 is high), the outputs take on their internal 0 states. Being active high, that means they are forced low.

The heart of this device and its symbol is an 8-bit shift register. It has a single D input, pin 2, which is shown to be affected by affecting input number 9, which is C9, meaning it will be enabled if C9 = 1. See 4.8 for an explanation of C dependency and 5.0 for a discussion of bistable elements. Since the C input is dynamic, the storage elements are edge-triggered flip-flops. While C9 = 1, which in this case will occur on the transition of pin 3 from low to high, the state of the D input will be stored. Pin 2 is shown to be active low so to store a 1, pin 2 must be low.

In addition to controlling the D input, pin 3 is shown by \rightarrow to have an additional function. As pin 3 goes from low to high, data stored in the shift register is shifted one position. The right-pointing arrow means that the data is shifted away from the control block (down).

On the right side of the symbol an abbreviation technique has been used that is practical only when

6.8 SN75551 Electroluminescent Row Driver with CMOS-Compatible Inputs



The heart of this device and its symbol is a 32-bit shift register. It has a single D input, pin 24, which is shown to be affected by affecting input number 1, which is C1, meaning it will be enabled if $C1 = 1$. See 4.8 for an explanation of C dependency and 5.0 for a discussion of bistable elements. Since the C input is dynamic, the storage elements are edge-triggered flip-flops. While $C1 = 1$, which in this case will occur on the transition of pin 20 from high to low, the state of the D input will be stored. Pin 24 is shown to be active high so to store a 1, pin 24 must be high.

In addition to controlling the D input, pin 20 is shown by \rightarrow to have an additional function. As pin 20 goes from high to low, data stored in the shift register is shifted one position. The

right-pointing arrow means that the data is shifted away from the control block (down). The internal inputs of the output buffers are all shown to be affected by affecting inputs 2 and 3. Affecting input 2 is G2, meaning that pin 19 is ANDed with each of the internal register outputs, which are the buffer inputs. If pin 19 is high, the affected buffer inputs are enabled. If pin 19 is low, the 0 state is imposed on the affected buffer inputs. See 4.2 for an explanation of G (AND) dependency. Affecting input 3 is V3, meaning that pin 23 (active low) is ORed with each of the internal register outputs. If pin 23 is high, $V3 = 0$ and the affected buffer inputs are enabled. If pin 23 is low, $V3 = 1$ and the 1 state is imposed on the affected buffer inputs. See 4.4 for an explanation of V (OR) dependency. The effect of V3 is taken into account after that of G2 because of the order in which the labels appear. This means that the imposition of the 1 state on the internal buffer inputs by pin 23 would take precedence over the imposition of the 0 state by pin 19 in case both inputs were active. Pin 18 is shown to be an output directly from the thirty-second stage of the shift register. Pins 19 and 23 do not affect this output.

An abbreviation technique has been used for the shift register elements and associated the output lines. This technique is practical only when the pin numbers and pin names are both consecutive.

The symbol \square designates an n-p-n open-collector or similar output. In this device, the outputs are actually open-drain n-channel field-effect transistors. Instead of being grounded, the sources of these transistors are all connected to pin 21. This pin is used as an input to control the output voltage.