# Explanation of New Logic Symbols

## TABLE OF CONTENTS

*Title* *Page*

1. INTRODUCTION ..... 319
2. SYMBOL COMPOSITION ..... 319
3. QUALIFYING SYMBOLS ..... 321
   3.1 General Qualifying Symbols ..... 321
   3.2 Qualifying Symbols for Inputs and Outputs ..... 321
   3.3 Symbols Inside the Outline ..... 325
4. DEPENDENCY NOTATION ..... 326
   4.1 General Explanation ..... 326
   4.2 G, AND ..... 326
   4.3 Conventions for the Application of Dependency Notation in General ..... 328
   4.4 V, OR ..... 329
   4.5 N, Negate (Exclusive OR) ..... 329
   4.6 Z, Interconnection ..... 330
   4.7 C, Control ..... 331
   4.8 S, Set and R, Reset ..... 331
   4.9 EN, Enable ..... 332
   4.10 M, Mode ..... 333
   4.11 A, Address ..... 335
5. BISTABLE ELEMENTS ..... 338
6. CODERS ..... 339
7. USE OF A CODER TO PRODUCE AFFECTING INPUTS ..... 340
8. USE OF BINARY GROUPING TO PRODUCE AFFECTING INPUTS ..... 341
9. SEQUENCE OF INPUT LABELS ..... 341
10. SEQUENCE OF OUTPUT LABELS ..... 342

## LIST OF TABLES

*Table* *Title* *Page*

I. General Qualifying Symbols ..... 322
II. Qualifying Symbols for Inputs and Outputs ..... 323
III. Symbols Inside the Outline ..... 324
IV. Summary of Dependency Notation ..... 337

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

F.A. Mann MS 84
Texas Instruments Incorporated
P.O. Box 225012
Dallas, Texas 75265
Telephone (214) 995-3746

IEEE Standards may be purchased from:

Institute of Electrical and Electronics Engineers, Inc.
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

# EXPLANATION OF NEW LOGIC SYMBOLS

by F. A. Mann

## 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.

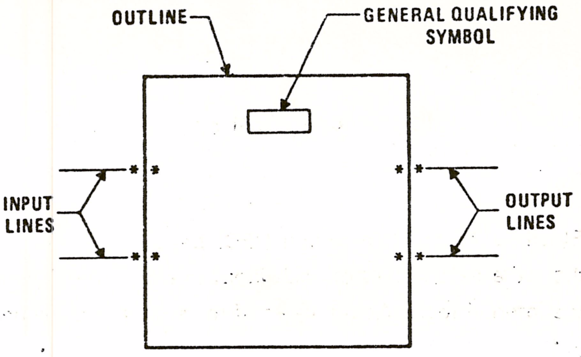
Internationally, Working Group 2 of IEC Technical Committee TC-3 is preparing a new document (Publication 617-12) that will consolidate the original work started in the mid 1960's and published in 1972 (Publication 117-15) and the amendments and supplements that have followed. Similarly for the USA, IEEE Committee SCC 11.9 is revising the publication IEEE Std 91/ANSI Y32.14. Texas Instruments is participating in the work of both organizations and this Supplement to the TTL Data Book introduces new logic symbols in anticipation of the new standards. When changes are made as the standards develop, future editions of the TTL Data Book 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 will finally 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 book; comparing the symbols with functional block diagrams and/or function tables will further help that understanding.

## 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 I shows the general qualifying symbols used in this data book. 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 11.

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 qualifying symbol or label inside the element.



*Possible positions for qualifying symbols relating to inputs and outputs

FIGURE 1 – SYMBOL COMPOSITION

The outlines of elements may be abutted or embedded in which case the following conventions apply. There is no logic connection between the elements when the line common to their outlines is in the direction of signal flow. There is at least one logic connection between the elements when the line common to their outlines is perpendicular to the direction of signal flow. The number of logic connections between elements will be clarified by the use of qualifying symbols and this is discussed further under that topic. If no indications are shown on either side of the common line, it is assumed there is only one connection.

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.

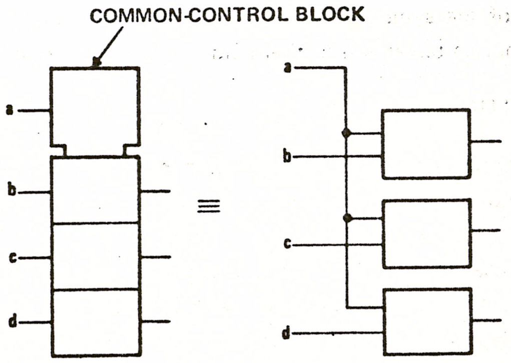


FIGURE 2 – ILLUSTRATION OF COMMON- CONTROL BLOCK

A common output depending on all elements of the array can be shown as the output of a common-output element. Its distinctive visual feature is the double line at its top. In addition the common-output element may have other inputs as shown in Figure 3. The function of the common-output element must be shown by use of a general qualifying symbol.

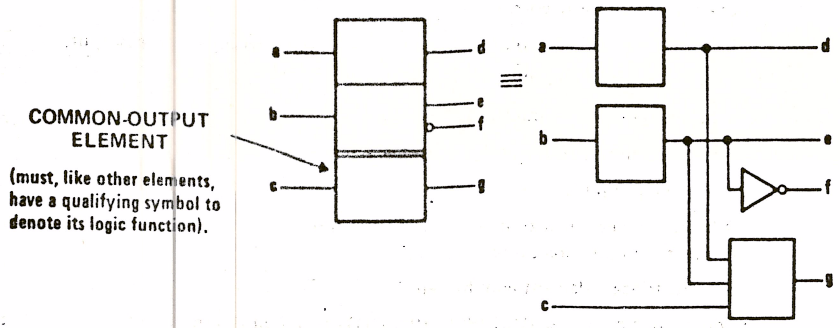

a
b
c
d
e
f
g
≡
a
b
c
d
e
f
g
COMMON-OUTPUT ELEMENT
(must, like other elements, have a qualifying symbol to denote its logic function).

FIGURE 3 – ILLUSTRATION OF COMMON-OUTPUT ELEMENT

## 3. QUALIFYING SYMBOLS

### 3.1 General Qualifying Symbols

Table I shows the general qualifying symbols used in this data book. 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.

### 3.2 Qualifying Symbols for Inputs and Outputs

Qualifying symbols for inputs and outputs are shown in Table II and will be familiar to most users with the possible exception of the logic polarity and analog signal indicators. 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 triangular 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.

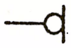
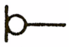
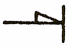
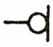
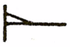
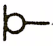
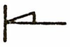
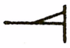
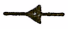
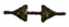
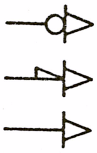
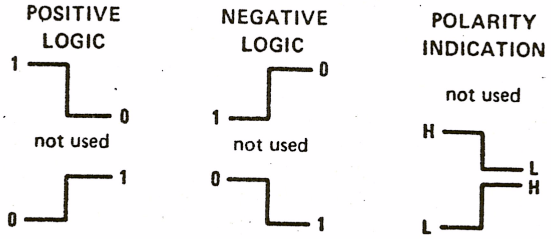
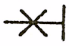
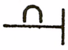
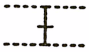
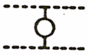
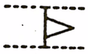
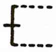
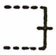
# EXPLANATION OF NEW LOGIC SYMBOLS

## TABLE I – GENERAL QUALIFYING SYMBOLS

| SYMBOL | DESCRIPTION | EXAMPLE |
|---|---|---|
| & | AND gate or function. | SN7400 |
| >1 | OR gate or function. The symbol was chosen to indicate that at least one active input is needed to activate the output. | SN7402 |
| =1 | Exclusive OR. One and only one input must be active to activate the output. | SN7486 |
| = | Logic identity. All inputs must stand at same state. | SN74180 |
| 2k | An even number of inputs must be active. | SN74180 |
| 2k+1 | An odd number of inputs must be active. | * |
| 1 | The one input must be active. | SN7404 |
| ▷ or ◁ | A buffer or element with more-than usual output capability (symbol is oriented in the direction of signal flow). | SN74S436 |
| ⎎ | Schmitt trigger; element with hysteresis. | SN74LS18 |
| X/Y | Coder, code converter (DEC/BCD, BIN/OUT, BIN/7-SEG, etc.). | SN74LS347 |
| MUX | Multiplexer/data selector. | SN74150 |
| DMUX or DX | Demultiplexer. | SN74138 |
| Σ | Adder. | SN74LS385 |
| P–Q | Subtracter. | SN74LS385 |
| CPG | Look-ahead carry generator. | SN74182 |
| π | Multiplier. | SN74LS384 |
| COMP | Magnitude comparator. | SN74LS682 |
| ALU | Arithmetic logic unit. | SN74LS381 |
| ⎍ | Retriggerable monostable. | SN74LS422 |
| 1⎍ | Non-retriggerable monostable (one-shot). | SN74121 |
| G<br>⎍⎍ | Astable element. Showing waveform is optional. | SN74LS320 |
| !G<br>⎍⎍ | Synchronously starting astable. | SN74LS624 |
| G!<br>⎍⎍ | Astable element that stops with a completed pulse. | * |
| SRGm | Shift register. m = number of bits. | SN74LS595 |
| CTRm | Counter. m = number of bits; cycle length = $2^m$. | SN54LS590 |
| CTR DIVm | Counter with cycle length = m. | SN74LS668 |
| ROM | Read-only memory. | * |
| RAM | Random-access read/write memory. | SN74170 |
| FIFO | First-in, first-out memory. | SN74LS222 |

*Not all of the general qualifying symbols have been used in this book, but they are included here for the sake of completeness.

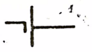
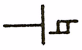
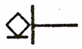
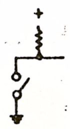
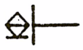
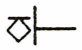
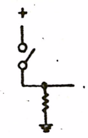
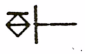
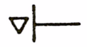
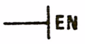
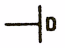
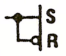
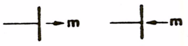
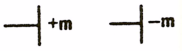
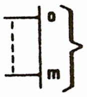
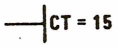
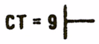
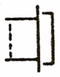
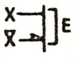
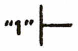
TABLE II – QUALIFYING SYMBOLS FOR INPUTS AND OUTPUTS

| | |
|---|---|
|  | 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. |
|  | Dynamic inputs active on indicated transition   POSITIVE LOGIC<br>1<br>0<br>not used<br>0<br>1<br>NEGATIVE LOGIC<br>0<br>1<br>not used<br>0<br>1<br>POLARITY INDICATION<br>not used<br>H<br>L<br>H<br>L |
|  | Nonlogic connection. A label inside the symbol will usually define the nature of this pin. |
|  | Input for analog signals. |
|  | 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. |
|  | Internal input (virtual input). It always stands at its internal 1 state unless affected by an overriding dependency relationship. |
|  | Internal output (virtual output). Its effect on an internal input to which it is connected is indicated by dependency notation. |

The internal connections between logic elements abutted together in a symbol may be indicated by the symbols shown. Each logic connection may be shown by the presence of qualifying symbols at one or both sides of the common line and if confusion can arise about the numbers of connections, use can be made of one of the internal connection symbols.

The internal (virtual) input is an input originating somewhere else in the circuit and is not connected directly to a terminal. The internal (virtual) output is likewise not connected directly to a terminal.

## TABLE III – SYMBOLS INSIDE THE OUTLINE

| | |
|---|---|
|  | Postponed output (of a pulse-triggered flip-flop). The output changes when input initiating change (e.g., a C input) returns to its initial external state or level. See § 5. |
|  | Bi-threshold input (input with hysteresis) |
|  | NPN 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 NPN open-collector output but is suplemented with a built-in passive pull-up. |
|  | NPN 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 NPN open-emitter output but is supplemented with a built-in passive pull-down. |
|  | 3-state output |
|  | Enable input<br>When at its internal 1-state, all outputs are enabled.<br>When at its internal 0-state, open-collector and open-emitter outputs are off, three-state outputs are at normally defined internal logic states and at external high-impedance state, and all other outputs (e.g., 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. |
|  | Counting up (down) inputs, m = 1, 2, 3 etc. If m = 1, it is usually not shown. |
|  | Binary grouping. m is highest power of 2. |
|  | The contents-setting input, when active, causes the content of a register to take on the indicated value. |
|  | The content output is active if the content of the register is as indicated. |
|  | Input line grouping . . . . indicates two or more terminals used to implement a single logic input.<br>e.g., The paired expander inputs of SN7450.  |
|  | Fixed-state output always stands at its internal 1 state. For example, see SN74185. |

The application of internal inputs and outputs requires an understanding of dependency notation, which is explained in Section 4.

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 SN54LS440 symbol illustrates this principle.

### 3.3 Symbols Inside the Outline

Table III shows some symbols used inside the outline. Note particularly that open-collector, open-emitter, and three-state outputs have distinctive symbols. Also note that an EN input affects all of the outputs of the circuit and has no effect on inputs. When an enable input affects only certain outputs 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 8. 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 data book 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. See Figure 28. 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.

Other symbols are used inside the outlines in this data book in accordance with the IEC/IEEE standards but are not shown here. Generally these are associated with arithmetic operations and are self-explanatory.

When nonstandardized information is shown inside an outline, it is usually enclosed in square brackets [like these].

# 4 DEPENDENCY NOTATION

## 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, ten types of dependency have been defined and all of these are used in this data book. They are listed below in the order in which they are presented and are summarized in Table IV following 4.11.

| 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 | C, Control |
| 4.8 | S, Set and R, Reset |
| 4.9 | EN, Enable |
| 4.10 | M, Mode |
| 4.11 | A, Address |

## 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 nine other forms of dependency have since been defined, the ways to invoke AND dependency are now reduced to one.

In Figure 4 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.

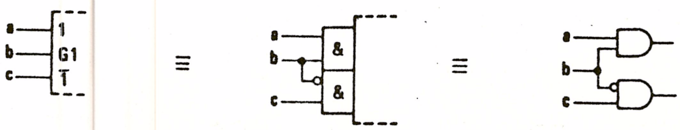


FIGURE 4 – G DEPENDENCY BETWEEN INPUTS

In Figure 5, 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 6 shows input a to be ANDed with a dynamic input b.

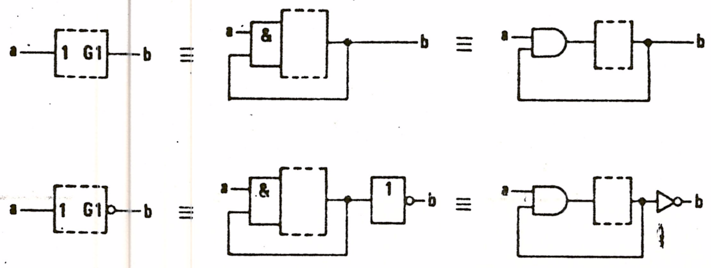


FIGURE 5 – G DEPENDENCY BETWEEN OUTPUTS AND INPUTS

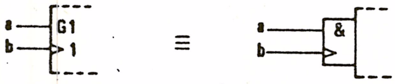


FIGURE 6 – G DEPENDENCY WITH A DYNAMIC INPUT

The rules for G dependency can be summarized thus:

When a G*m* input or output (*m* is a number) stands at its internal 1 state, all inputs and outputs affected by G*m* stand at their normally defined internal logic states. When the G*m* input or output stands at its 0 state, all inputs and outputs affected by G*m* 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. See Figure 4.

If two affecting inputs or outputs have the same letter and same identifying number, they stand in an OR relationship to each other. See Figure 7.

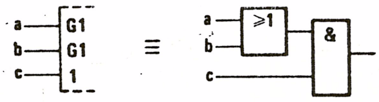


FIGURE 7 – OR'ED AFFECTING INPUTS

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. See 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. See Figure 12.

If the labels denoting the functions of affected inputs or outputs must be numbers, (e.g., outputs of a coder), the identifying numbers to be associated with both affecting inputs and affected inputs or outputs will be replaced by another character selected to avoid ambiguity, e.g., Greek letters. See Figure 8.

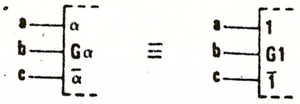


FIGURE 8 – SUBSTITUTION FOR NUMBERS

## 4.4 V (OR) Dependency

The symbol denoting OR dependency is the letter V. See Figure 9.

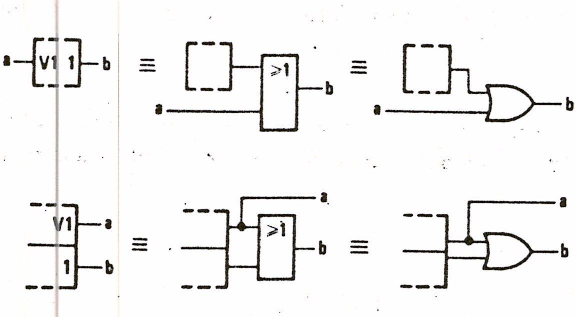
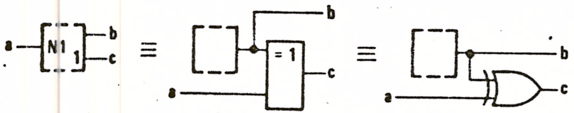


FIGURE 9 – V (OR) DEPENDENCY

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

## 4.5 N (Negate) (X-OR) Dependency

The symbol denoting negate dependency is the letter N. See Figure 10. Each input or output affected by an N*m* input or output stands in an exclusive-OR relationship with the N*m* input or output.



If a = 0, then c = b
If a = 1, then c = $\bar{b}$

FIGURE 10 – N (NEGATE) (X-OR) DEPENDENCY

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

### 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 Z*m* input or output will be the same as the internal logic state of the Z*m* input or output, unless modified by additional dependency notation. See Figure 11.

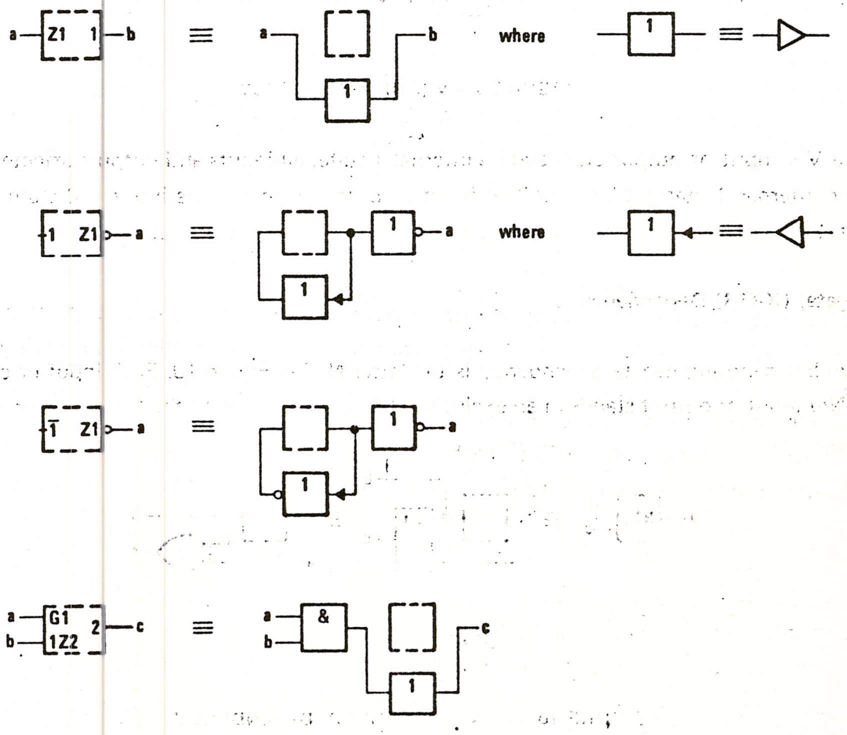


FIGURE 11 – Z (INTERCONNECTION) DEPENDENCY