

Recognized as an
American National Standard (ANSI)

ANSI/IEEE Std 91-1984
ANSI/IEEE Std 91a-1991

IEEE Standard Graphic Symbols for Logic Functions

(Including and incorporating IEEE Std 91a-1991, Supplement to IEEE Standard Graphic Symbols for Logic Functions)

Sponsor
IEEE Standards Coordinating Committee 11,
Graphic Symbols and Designations

ANSI/IEEE Std 91-1984

Approved 17 September 1981, IEEE Standards Board
Approved 26 October 1982, American National Standards Institute
Adopted for Mandatory Use 10 May 1984 Adopted U.S. Department of Defense (DoD)

ANSI/IEEE Std 91a-1991

Approved 27 June 1991 IEEE Standards Board
Approved 3 June 1994 American National Standards Institute
Adopted for Mandatory Use 1 July 1991 U.S. Department of Defense (DoD)

Abstract: Graphic symbols for representing logic functions or physical devices capable of carrying out logic functions are presented. Descriptions of logic functions, the graphic representation of these functions, and examples of their applications are provided. The symbols are presented in the context of electrical applications, but most may also be applied to nonelectrical systems (for example, pneumatic, hydraulic, or mechanical). The supplement provided additional internationally approved graphic symbols and made corrections as needed to IEEE Std 91-1984.

Keywords: dependency notation, industry standards, logic diagrams, logic function, logic symbols, military standards

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

Copyright © 1996 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1996. Printed in the United States of America

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

ANSI/IEEE Std 91-1984
10 May 1984
(Superseding
ANSI Y32.14-1973
IEEE Std 91-1973)

Acceptance Notice

This non-government document was adopted on 10 May 1984, and is approved for use by the Department of Defense (DoD). The IEEE has furnished the clearances required by existing regulations. Copies of the document are stocked at the Standardization Documents Order Desk, Bldg 4D, 700 Robbins Avenue, Philadelphia, PA 19111-5094 for issue to DoD activities only. Other Government activities, contractors, private concerns, or other requestors must obtain the document from:

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017-2394 USA

Title of Document: **IEEE Standard Graphic Symbols for Logic Functions**

Document No: **ANSI/IEEE Std 91-1984**

Date of Specific Issue Adopted: **9 June 1984**

Releasing Industry Group: **The Institute of Electrical and Electronics Engineers, Inc.**

Custodians:	Military Coordinating Activity:
Army—AR	Army—AR
Navy—SH	
Air Force—13	(Project DRPR—0231)

Review Activities:
Army—AV, MI, AM CR, ER
Navy—OS, AS
Air Force—15, 17, 11

User Activities:
Army—AT
Navy—MC

NOTICE: When reaffirmation, amendment, revision, or cancellation of this standard is initially proposed, the industry group responsible for this standard shall inform the military coordinating activity of the proposed change and request participation.

NOTICE OF ADOPTION
ADOPTION NOTICE 1
1 JULY 1991
IEEE Std 91a-1991
27 JUNE 1991

The Institute of Electrical and Electronics Engineers, Inc., IEEE Std 91a-1991, was adopted on 1 July, 1991 and is approved for use by the Department of Defense (DoD). Copies of this document are stocked at the Standardization Documents Order Desk, Bldg 4D, 700 Robbins Avenue, Philadelphia, PA 19111-5094 for issue to DoD activities only. Other Government activities, contractors, private concerns, or other requestors must obtain the document from:

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017

Title of Document: **Supplement to IEEE Standard Graphic Symbols for Logic Functions**

Date of Specific Issue Adopted: **27 June 1991**

Releasing Industry Group: **The Institute of Electrical and Electronics Engineers, Inc.**

Custodians:	Military Coordinating Activity:
Army—AR	Army—AR
Navy—OS	
Air Force—13	(Project DRPR—0314)

Review Activities:

- Army—AT, AV, CR, ER, MI
- Navy—AS, EC, MC, SH
- Air Force—11, 13, 15, 16, 17, 99
- DLA—DH, ES
- NSA—NS

User Activities:

- Army—AT

AMSC N/A
DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

AREA DRPR

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying all patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (508) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

(This introduction is not part of IEEE Std 91-1984, IEEE Standard Graphic Symbols for Logic Functions, or of IEEE Std 91a-1991, Supplement to IEEE Std 91-1984.)

Two standards are included in this document: IEEE Std 91-1984, IEEE Standard Graphic Symbols for Logic Functions, and IEEE Std 91a-1991, Supplement to IEEE Std 91-1984. In this edition (published in 1996), the two standards have been merged to make it more convenient for the user.

The following sections of the original IEEE Std 91-1984 were extensively revised and replaced by IEEE Std 91a-1991:

Pictorial Table of Contents

Table of Contents

Section 6: Symbols for Highly Complex Functions

Appendix A: Recommended Symbol Proportions

Appendix D: Integrated Circuits Used as Examples, Commercial Part Numbers vs. Symbol Numbers

Index

Numerous other revisions to IEEE Std 91-1984 in Sections 1 through 5 were made. The new Table of Contents and the new Index indicate specifically where revisions were made. Further, change bars marked in the outside margins indicate the locations of significant changes. Each standard has a unique foreword, included below.

IEEE Std 91-1984 foreword

This standard defines an international *language* by which it is possible to determine the functional behavior of a logic circuit as described on a logic or circuit diagram with minimal reference to supporting documentation. Like natural languages, the *language* set forth in this standard has been designed to allow a single concept to be expressed in one of several different ways, according to the demands of a particular situation. Consequently, this standard does not attempt, nor intend to establish single *correct* symbols for particular devices. A symbol appropriate for one application of a device may not be appropriate for another.

The contributors to this standard represent a broad range of institutions, technologies, and documentation needs. They include industrial, governmental, and educational organizations, producers and consumers of devices and equipment, users and non-users of computer-aided design and drafting, and a considerable range of aesthetic preferences. That a consensus of such diverse interests could be achieved in producing this standard is indicative of not only the utility of the approach, but more importantly, of the increasing need among designers and maintainers of digital systems for a common and more nearly complete symbolic language.

This revision is the result of a continuing activity to arrive at a useful notation to permit free interchange of information on the design of binary-operated controls and systems. It is the latest step in a program that began in 1956 within the IEEE to develop a comprehensive single standard, consistent with ongoing developments in technology and logic symbology, from several ad hoc, industry, military, and international standards. In 1960, an ad hoc group on logic diagram graphic symbols was formed within the American National Standards Institute in order to develop a draft American Standard. In 1961, this committee became a permanent subcommittee, Y32.14, of the Graphic Symbols Committee, Y32, under the cosecretariat of ASME and IEEE. Its work resulted in the publication of IEEE Std 91-1962 (ANSI Y32.14-1962), adopted in 1965 by the US Navy. The subcommittee was reorganized in 1969 to prepare a new draft standard that would have broader acceptance and be in accord with the developments within the International Electrotechnical Commission (IEC). ANSI/IEEE Std 91-1973 (Y32.14-1973) subsequently received approval from ANSI, and the US Department of Defense, and was substantially compatible with IEC Pub 117-15, Recommended Graphical Symbols: Binary Logic Elements. Since 1977 the preparing committee, IEEE SCC 11.9, has worked closely with IEC Technical Committee 3 to prepare major new revisions of this standard and IEC Pub 617, Part 12 (the successor to Pub 117, Part 15). The aim was for a US standard that would be mutually compatible with the IEC standard, broadly acceptable, and that would provide notation or guidelines by which any SSI through VLSI

device might be usefully and accurately represented. Fourteen drafts of this document were prepared in parallel with drafts of the new IEC document by an ad hoc working group of SCC 11.9 before the preparers and reviewers believed that these goals had been met.

Symbology, such as language and technology, will continue to evolve, and IEEE SCC 11.9 will continue to work with IEC TC3 to update logic symbol standards as future needs dictate. Suggestions for improvement of this standard are welcomed. They should be addressed to:

Secretary
IEEE Standards Board
The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017-2394 USA

When this standard was approved, the IEEE Standards Coordinating Committee on Graphic Symbols and Designations, Subcommittee SC 11.9 on Logic Symbols, had the following membership:

Thomas R. Smith, *Chair**

Charles J. Andrasco
Richard R. Barta
M. Paul Chinitz*
Edwin D. Edwards†
Philip H. Enslow, Jr.
Charles A. Fricke†
Arthur C. Gannett
James J. George
Tommy J. Gibbons
Edward J. Harrington*

Verle W. Harris
Gordon A. Knapp*
Joseph Kochanski*
George E. Lemieux†
Fred A. Mann*‡
Rollin P. Mayer
Mark Mrozowski*
Conrad R. Muller*
John B. Peatman*
Richard W. Pinger*

V. Tom Rhyne*
John P. Russell* **
Henry H. Seaman
Thomas R. Smith*
Robert Tobias
James L. Vargo*
Frank B. Wahl, Jr.*
R. J. Yuhas

US Air Force Logistics Command LOLME (Robert C. Gardner)
US Army Armament Research and Development Command DRDAR-TST-S (M. E. Taylor)

*Member of ad hoc working group

†Resigned

**Secretary, September 1979 to April 1980; prepared all drafts through 1981

‡Head of delegation to IEC TC3/SC3A/WG2 on Logic Symbols

The IEEE Standards Coordinating Committee on Graphic Symbols and Designations SCC 11 had the following membership:

C. A. Fricke†, *Chair*

C.R. Muller, *Secretary*

Robert B. Angus, Jr.
John M. Carroll
Gordon A. Knapp

John B. Peatman
J. William Siefert
Thomas R. Smith

Roger M. Stern
Leter H. Warren
Steven A. Wasserman

†Resigned

At the time the standard was approved on September 17, 1981, the IEEE Standards Board had the following membership:

Irvin N. Howell, Jr., Chair
Irving Kolodny, Vice Chair
Sava I. Sherr, Secretary

G. Y. R. Allen
J. J. Archambault
James H. Beall
John T. Boettger
Edward Chelotti
Edward J. Cohen
Len S. Corey

Jay Forster
Kurt Greene
Loering M. Johnson
Joseph L. Koepfinger
John E. May
Donald T. Michael*
J. P. Riganati

Frank L. Rosa
Robert W. Seelbach
Jay A. Stewart
Virginius N. Vaughan, Jr.
Art Wall
Robert E. Weiler

*Member emeritus

IEEE Std 91a-1991 foreword

IEEE Std 91-1991 provides a way to describe a single logic function with a symbol. The symbols that result from using IEEE Std 91-1984 have sometimes been called “the new logic symbols,” despite the fact that one can still find distinctive-shape AND and OR gates, J and K inputs to flip-flops, and most other time-honored features of logic symbols. However, prior to IEEE Std 91-1984, there were no standardized symbols or symbology methods, for example, for memories, demultiplexers, arithmetic elements, multifunction registers, open-collector outputs, digital switches, or devices with multiple clocks or address ports. What was truly new was that these features were integrated into a comprehensive system for constructing logic symbols to meet not only current, but future needs. Where the “old” system relied on an inventory of symbols, the “new” system relies on an inventory of symbol elements and techniques. The inherent ability of the “language” to describe new devices can be seen in the relatively few additions that have been required in this supplement.

Because IEEE Std 91-1984 is not based on English-language mnemonics, symbols based on it may communicate less to the uninitiated reader who has already memorized a component catalog. However, if the reader knows the language, and especially if he or she does not intimately know the device, the system can communicate far more than any nonstandard mnemonic-based system, with a smaller set of things to be remembered. The notation does more than suggest to the knowledgeable user what various inputs and outputs do. Instead, it provides a rather detailed description.

The use of this system requires training, as does any other system of notation. The difference between this system and those that preceded it is that, once trained, the user has available more information in less space, from more sources, and with less need for supporting documentation. Once the user has learned the rules, he or she has learned them not only for existing devices, but for new ones as well. That IEEE Std 91-1984 is virtually identical to the national standards of dozens of other countries is more than a small bonus.

Future editions of IEEE Std 91 and IEC 617-12 are expected to continue to evolve to address developments in the technology of logic devices. Suggestions for the improvement of this standard are welcomed.

The supplement to IEEE Std 91-1984, IEEE Std 91a-1991 was the result of nearly ten years of ongoing work by an ad hoc Working Group of IEEE SCC 11.9 and IEC TC3 (International Electrotechnical Commission Technical Committee 3) to harmonize IEEE Std 91 and its international counterpart, IEC 617-12. The supplement brings IEEE Std 91-1984 into conformance with IEC 617-12 (1991) and, once more, includes some additional symbols and techniques published in advance of their publication by IEC.

In addition to corrections and clarifications to IEEE Std 91-1984, the supplement included new material to cover simplification of arrays (2.3.1.2), inputs with special amplification (3.3-9.5), in-line negation indication (4.3.1), definitions of codes used in coders (5.4.1, 5.4.2), display elements (5.15), bus indicators (6.1.9), and representation of data paths on internal diagrams (6.2.2).

This document was prepared by an ad hoc Working Group of Subcommittee 11.9 on Logic Symbols of the IEEE Standards Coordinating Committee 11 on Graphic Symbols and Designations. The members of the working group were:

Thomas R. Smith, *Chair*[†]

C. Andrews	L. Hanson*	J. A. Pendell
R. B. Angus	G. F. Iacovoni	J. Rouse
R. R. Barta	S. Ingram	J. P. Russell
L. Burns	G. A. Knapp	L. Sandoval
N. Caparossi	D. Kollmorgan	J. Sents
T. Connors	R. Landry	P. Simison
L. Ciskowski	D. Lauer	R. Sohan
E. Draper	F. Mann†	R. M. Stern
R. Farni	J. Massaro	J. Vargo
B. Fiekel	J. F. Morongiello	L. Wong*
J. George	C. R. Muller	

*Assisted in the preparation of drafts and publication copy.

†U.S. member of IEC TC3/SC3A/WG2 on Logic Symbols

The IEEE Standards Coordinating Committee 11 on Graphic Symbols and Designations delegated the balloting of this supplement to the following persons who constituted the members of SCC11.9 that approved this supplement for submission to the IEEE Standards Board:

R. B. Angus	A. Hartikainen	C. R. Muller
R. R. Barta	G. F. Iacovoni	N. Raskhodoff
J. Bloodgood	S. Ingram	J. Rouse
N. Caporossi	S. Johnson	R. Sandige
G. Carnovale	R. Kaskowitz	L. Sandoval
L. Ciskowski	G. A. Knapp	T. R. Smith
D. M. Clark	R. Landry	T. Suzuki
P. H. Enslow	F. Mann	P. Tremblay
A. Gannett	J. Massaro	J. Vargo
J. George	R. P. Mayer	L. H. Warren
L. Hanson	C. A. Mitchell	A. Wiseman

When the IEEE Standards Board approved this standard on June 27, 1991, it had the following membership:

Marco W. Migliaro, *Chair*
Donald C. Loughry, *Vice Chair*
Andrew G. Salem, *Secretary*

Dennis Bodson	Donald N. Heirman	Lawrence V. McCall
Paul L. Borrill	Kenneth D. Hendrix	Donald T. Michael*
Clyde Camp	John W. Horch	Stig L. Nilsson
James M. Daly	Ben C. Johnson	John L. Rankine
Donald C. Fleckenstein	Ivor N. Knight	Ronald H. Reimer
Jay Forster*	Joseph L. Koepfinger*	Gary S. Robinson
David F. Franklin	Irving Kolodny	Terrance R. Whittemore
Ingrid Fromm	Michael A. Lawler	
Thomas L. Hannan	John E. May, Jr.	

Deborah A. Czyz *IEEE Standards Project Editor*

* Member Emeritus

Acknowledgements

This standard owes its existence to the cooperative efforts of many people and organizations, most especially to the excellent technical contributions of the members of the ad hoc working group and the support given each of them by their staffs and employers. Particular thanks goes to the following organizations for their invaluable assistance:

For preparation and distribution of meeting, draft, and final documents: General Electric Co, Texas Instruments, Inc, Control Data Corp, Digital Equipment Corp, Allen-Bradley, and D.A.T.A. Inc.

For financial support of US delegations at IEC working-group meetings, without which this work would not have progressed successfully: Texas Instruments, Inc, McGraw-Hill Book Co, Hewlett-Packard, Digital Equipment Corp, and Control Data Corp.

1.	Introduction.....	1
1.1	Purpose.....	1
1.2	Scope.....	1
1.3	Organization.....	1
1.4	Applicable documents.....	2
1.5	Definitions.....	2
1.6	Orientation of qualifying symbols	4
2.	Symbol construction	5
2.1	Composition of a symbol	5
2.2	Outlines	6
2.3	Use and combination of outlines.....	7
3.	Qualifying symbols associated with inputs, outputs, and other connections.....	14
3.1	Negation, polarity, and dynamic input symbols	14
3.2	Internal connections	16
3.3	Symbols inside the outline	17
3.4	Subsidiary (formerly nonlogic) connections and signal-flow indicators	28
4.	Dependency notation.....	31
4.1	General explanation	31
4.2	Summary of types of dependencies	31
4.3	Application of dependency notation	32
4.4	Special techniques used in dependency notation	51
5.	Combinational and sequential elements.....	59
5.	General notes.....	59
5.1	Basic combinational elements.....	59
5.2	Buffers with special amplification, drivers, receivers, and bidirectional switches.....	66
5.3	Elements exhibiting hysteresis Schmitt triggers Bithreshold detectors	69
5.4	Coders	70
5.5	Signal-level converters.....	81
5.6	Multiplexers and demultiplexers.....	81
5.7	Arithmetic elements	85
5.8	Delay elements.....	90
5.9	Basic bistable elements	90
5.10	Bistable elements with special switching properties.....	94
5.11	Monostable elements.....	95
5.12	Astable elements	96
5.13	Shift registers and counters	100
5.14	Memories	106
5.15	Display elements	111
6.	Symbols for highly complex functions	115
6.1	General symbol and basic rules	115
6.2	Internal diagrams.....	119
6.3	Examples of complex elements.....	121

Annex A Recommended symbol proportions (Informative)	130
Annex B Revised or deleted symbols (Informative).....	134
Annex C Composite chart showing relationship of graphic symbols for logic diagrams from IEEE Std 91-1984, IEEE Std 91a-1991, and the superseded standards (Informative)	137
Annex D Integrated circuits used as examples commercial part numbers versus symbol numbers (Informative)	147

An American National Standard

IEEE Standard Graphic Symbols for Logic Functions

1. Introduction

1.1 Purpose

This standard (the combined documents of ANSI/IEEE 91-1984 and its supplement, ANSI/IEEE Std 91a-1991) establishes symbols for representing logic functions or devices. These symbols enable users to understand the logic characteristics of these functions or devices without requiring specific knowledge of their internal characteristics. ANSI/IEEE Std 91a-1991 supplements ANSI/IEEE 91-1984 by providing additional internationally approved graphic symbols for logic functions or devices as well as corrections to the 1984 document.

1.2 Scope

This standard contains graphic symbols for representing logic functions or physical devices capable of carrying out logic functions. Descriptions of logic functions, the graphic representation of these functions, and examples of their applications are given. The symbols are represented in the context of electrical applications, but most may also be applied to nonelectrical systems (for example, pneumatic, hydraulic, or mechanical).

1.3 Organization

This document includes material from both ANSI/IEEE Std 91-1984 and its supplement document, ANSI/IEEE Std 91a-1991.

1.3.1 Supplementary material from ANSI/IEEE Std 91a-1991

All material from the supplement standard, ANSI/IEEE Std 91a-1991, has been included in this document. This supplement either replaces or adds to the original text in ANSI/IEEE 91-1984. Change bars in the margin mark the location where the supplementary material has been implemented.

1.3.2 Revised or deleted symbols

Symbols from superseded editions of this standard that have been revised or deleted are shown in Appendix B.

1.4 Applicable documents

1.4.1 Industry standards

The latest editions of the following industry documents may also be of interest:

*American National Standards*¹

ANSI Y1.1-1972, Abbreviations for Use on Drawings and in Text.

ANSI Y10.20-1975 with Supplement Y10.20a-1975, Mathematical Signs and Symbols for Use in Physical Science and Technology.

ANSI/IEEE Std 315-1975 (Y32.2-1975), Graphic Symbols for Electrical and Electronics Diagrams. (Including Reference Designation Class Designation Letters).

ANSI/IEEE Std 200-1975 (Y32.16-1975), Reference Designations for Electrical and Electronics Parts and Equipments.

1.4.2 Military standards²

The latest edition of the following military document may also be of interest:

MIL-STD-12, Military Standard Abbreviations for Use on Drawings, Specifications, Standards, and in Technical Documents.

1.4.3 International standards³

The following International Electrotechnical Commission (IEC) Publications may be of interest:

617 Part 12, Recommended Graphical Symbols, Binary Logic Elements. (All of the symbols in 617 Part 12 are included in this national standard, ANSI/IEEE Std 91-1984.)

113 Part 7, Diagrams, Charts and Tables; Application of Logic Symbols and the Preparation of Logic Diagrams.

All symbols and rules not shown, either explicitly or implicitly, in IEC Publications 113 or 617 at the time of publication of this standard are identified by the mark “•” (which is not a part of the symbol).⁴

1.5 Definitions

1.5.1 logic state: One of two possible abstract states that may be taken on by a logic (binary) variable.

1.5.1.1 0-state: The logic state represented by the binary number 0 and usually standing for an inactive or false logic condition.

¹ANSI publications are available from the Sales Department of American National Standards Institute, 11 West 42nd Street, New York, NY 10036.

²MIL publications are available from Customer Service, Defense Printing Service, 700 Robbins Avenue, Philadelphia, PA 19111-5094.

³IEC Standards are available in the US from American National Standards Institute, 11 West 42nd Street, New York, NY 10036.

⁴Except as individually noted, these rules and symbols are, in the opinion of the preparing committee, likely to be published in future editions of IEC Publications 113 or 617.

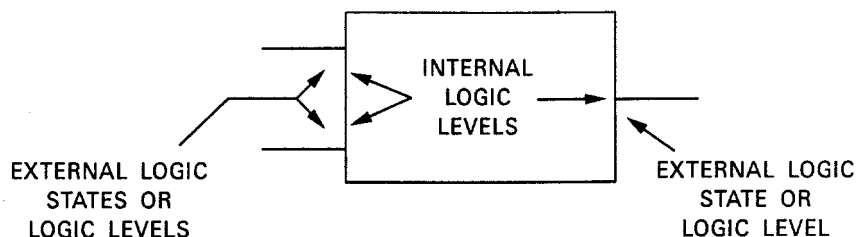
1.5.1.2 1-state: The logic state represented by the binary number 1 and usually standing for an active or true logic condition.

1.5.1.3 external logic state: A logic state assumed to exist outside a symbol outline

- 1) On an input line prior to any external qualifying symbol at that input, or
- 2) On an output line beyond any external qualifying symbol at that output.

1.5.1.4 internal logic state: A logic state assumed to exist inside a symbol outline at an input or an output.

Illustration of the concept



1.5.2 logic level: Any level within one of two nonoverlapping ranges of values of a physical quantity used to represent the logic states.

NOTE — A logic variable may be equated to any physical quantity for which two distinct ranges of values can be defined. In this standard, these distinct ranges of values are referred to as *logic levels* and are denoted H and L.

H is used to denote the logic level with the more positive algebraic value, and L is used to denote the logic level with the less positive algebraic value.

In the case of systems in which logic state are equated with other physical properties (for example, positive or negative pulses, presence or absence of a pulse), H and L may be used to represent these properties or may be replaced by more suitable designations.

1.5.2.1 high (H) level: A level within the more positive (less negative) of the two ranges of the logic levels chosen to represent the logic states.

1.5.2.2 low (L) level: A level within the more negative (less positive) of the two ranges of logic levels chosen to represent the logic states.

1.5.3 Logic conventions and polarity indication:

1.5.3.1 positive logic convention: The representation of the external 1-state and the external 0-state by the high (H) and low (L) levels, respectively.

1.5.3.2 negative logic convention: The representation of the external 1-state and the external 0-state by the low (L) and high (H) levels, respectively.

1.5.3.3 direct polarity indication: The indication of the relationship between the internal logic state and the external logic level at each input and output of every logic element directly by means of the presence or absence of the polarity symbol (∇). (See Symbols 3.1-4 through 3.1-8.)

1.5.4 logic function: A definition of the relationships that hold among a set of input and output logic variables.

1.5.4.1 combinational logic function: A logic function in which there exists one and only one resulting combination of states of the outputs for each possible combination of input states.

NOTE — The terms combinative and combinatorial have also been used to mean combinational.

1.5.4.2 sequential logic function: A logic function in which there exists at least one combination of input states for which there is more than one possible resulting combination of states at the outputs.

NOTE — The outputs are functions of variables in addition to the present states of the inputs, such as time, previous internal states of the element, etc.

Copyright © 1996 IEEE All Rights Reserved

1.5.4.3 bistable logic function (flip-flop): A sequential logic function that has two and only two stable internal output states.

1.5.5 element: As used within this standard, a representation of all or part of a logic function within a single outline, which may, in turn, be subdivided into smaller elements representing subfunctions of the overall function. Alternatively, the function so represented.

1.5.6 qualifying symbol: A symbol added to the basic outline of an element to designate the physical or logic characteristics of an input or output of the element or the overall logic characteristics of the element.

1.5.7 dependency notation: A means of obtaining simplified symbols for complex elements by denoting the relationships between inputs, outputs, or inputs and outputs, without actually showing all the elements and interconnections involved. (See Section 4.)

1.5.8 distributed function (dot logic, wired logic): A logic function (either AND or OR) implemented by connecting together outputs of the appropriate type (see Symbols 3.3-3 through 3.3-7); these outputs are the inputs of the logic function thus formed; the joined connection is the output.

1.6 Orientation of qualifying symbols

All lettering inside a symbol, including alphanumeric qualifying symbols, should be oriented parallel to the input and output lines.

The following symbols shall be oriented as described within this standard with respect to the inputs, outputs, and edge of the elements in which they appear.

2.3.2-1	Common control block
2.3.3-1	Common output element
3.3-1 through 3.1-11	Negation, polarity, and dynamic input symbols
3.2-1 through 3.2-6	Internal connection symbols
3.3-25 3.3-26	Bit-grouping symbols
3.3-37 3.3-38	Line-grouping symbols
▷	Amplifier symbol (Symbol 3.3-9 and 5.2)

All other qualifying symbols shall be oriented as shown with respect to the text inside the element.

2. Symbol construction

2.1 Composition of a symbol

2.1.1 General composition

A symbol comprises an outline or combination of outlines together with one or more qualifying symbols. Application of the symbols requires, in addition, the representation of input and output lines.

Symbols often make use of internal labels. It is essential to maintain clarity between internal labels that apply to signal lines and internal labels that serve as general qualifying symbols.

Labels shall be clearly separated, both horizontally and vertically, from each other and from other symbols or parts of the outline with which they are not meant to be associated. Signal-line labels shall also be placed such that they are not likely to be interpreted as general qualifying symbols.

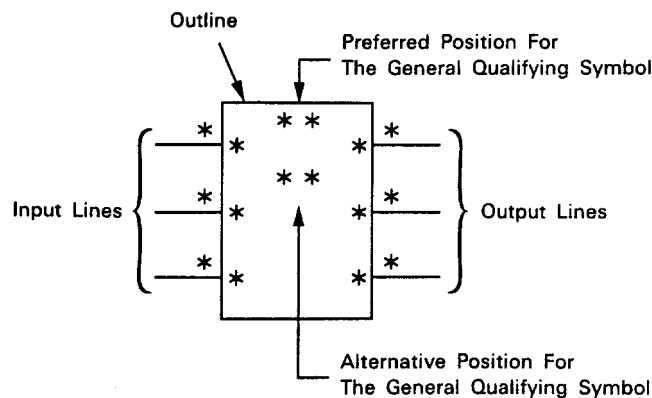
The preferred location for the general qualifying symbol is in the top of the element to which it applies, centered horizontally. An alternative position is centered vertically and horizontally. Application of these positioning rules for the general qualifying symbols should take into account the positions of embedded elements or labels associated with signal lines that alter the horizontal area available for placement of the general qualifying symbol.

Labels associated with signal lines shall be centered vertically next to those lines. Horizontally, there shall be a space between the signal-line label and the outline of the symbol or symbols attached to the outline. This horizontal space shall be large enough to allow a distinct separation, yet small enough to unambiguously associate the label with the given input or output.

See Appendix A for recommended proportions of symbols and IEEE Std 991 for more detailed recommendations on sizes and spacing.

Because of limitations in the publishing systems used, symbols in this document do not always follow the preceding recommendations.

Illustration of the concept



NOTE 1—The single * denotes possible positions for qualifying symbols relating to inputs and outputs.

NOTE 2—If, and only if, the function of an element is completely determined by the qualifying symbols associated with its inputs or outputs or both, no general qualifying symbol is needed.

General additional information may be included in a symbol outline as permitted by an applicable standard for preparation of diagrams (for example, IEC Publication 111-3 clause 3.4).

For supplementary information relating to the function of a complex logic element, see Section 6.

In some figures, lowercase letters that are not part of the symbols have been shown outside the outline just to identify the inputs and outputs as referenced in the description.

2.1.2 Nonstandardized information

Information not standardized in this standard relating to a specific input (output) may be shown in square brackets inside the outline adjacent to the relevant input (output) and should follow (precede) any qualifying symbols applying to the input (output), as shown in Symbol 5.13-17.

Other nonstandardized information may also be shown in square brackets inside the symbol outline, for example, Symbols 5.2-3 and 5.4-2.

2.1.3 Logic states of outputs

All outputs of a single, unsubdivided element always have identical internal logic states determined by the function of the element except when indicated otherwise by an associated qualifying symbol or label inside the outline.

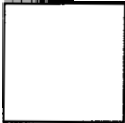
The subdivision of an element and the qualifying symbols referred to here include those explicitly shown and also those only implied according to the simplification rules of 2.3.1.1.

2.2 Outlines

The length-width ratio of rectangular outlines is arbitrary.

For permitted distinctive-shape symbol outlines see Section 5. The proportions of the symbol should remain constant in accordance with Appendix A.

For combination of outlines see 2.3.

No	Symbol	Description
2.2-1		Element outline (square shown, see 2.2)

2.3 Use and combination of outlines

2.3.1 Embedded and abutted elements

To reduce the space required for the representation of a group of associated elements, the outlines of the elements may be embedded or abutted provided the following rules are observed.

- 1) Embedded or abutted symbols may use nonrectangular outlines composed of vertical and horizontal lines to make efficient use of space (for example, Symbol 5.12-6).
Usage of distinctive-shape symbols in combination to form complex symbols (for example, used as embedded symbols) is discouraged.
- 2) There is no logic connection between elements when the line common to their outlines is in the direction of signal flow.
NOTE — This rule does not necessarily apply in those arrays in which there exist two or more directions of signal flow, for example, in symbols with a common control block or a common output element.

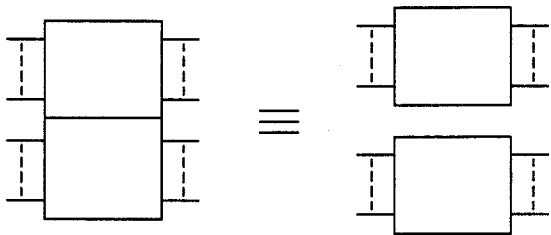
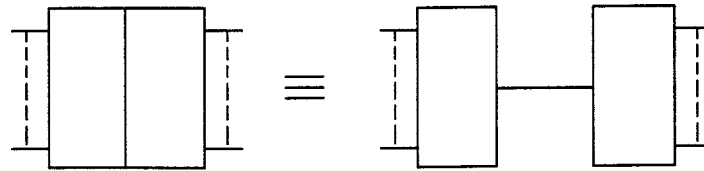
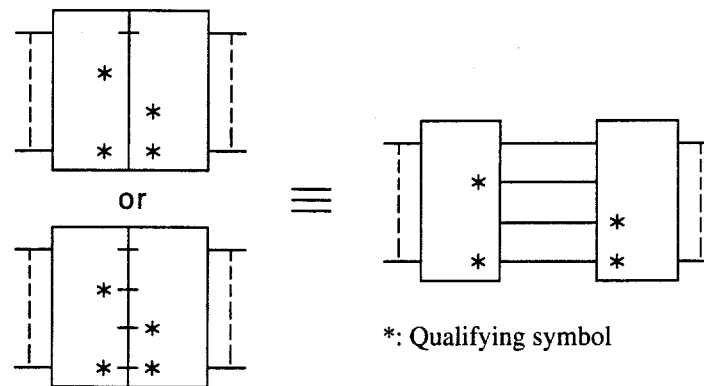


Illustration of the concept

- 3) There is at least one logic connection between elements if the line common to the 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 exists only one logic connection. Because common control blocks are not elements, no logic connections to or from a common control block exist except those to the attached array and those that are explicitly shown.

**Illustration of the concept**

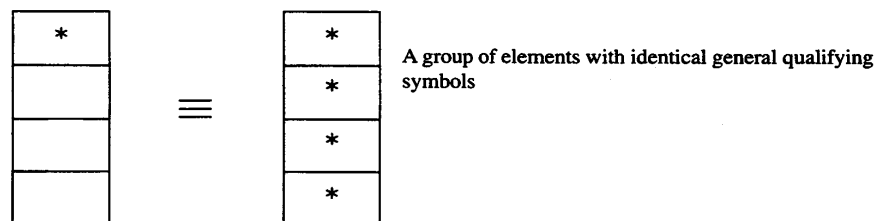
- 4) Each connection can be shown by the presence of qualifying symbols at one or both sides of the common line. If confusion is likely about the number of logic connections, use can be made of the internal connection symbol (Symbol 3.2-1).

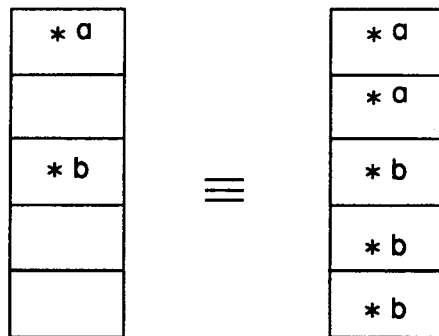
**Illustration of the concept**

For an example of combining the preceding concepts, see Symbol 5.1-19.

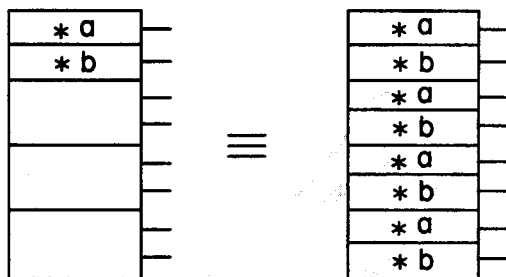
2.3.1.1

To represent an array of elements having the same qualifying symbols, it may be sufficient to show the symbols that are inside the outlines in only the first of the outlines, provided no confusion is likely. Similarly, identical elements that are subdivided may each be represented by an unsubdivided outline.

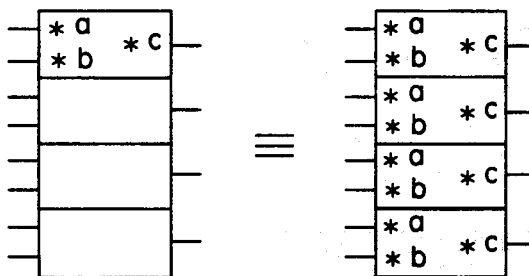
Illustration of the concept



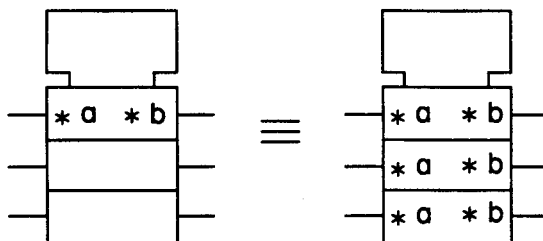
Two successive groups of elements



Two interlaced groups of four elements each



A group of elements with identical input and output qualifying symbols



A group of elements with identical input and output qualifying symbols and common control block (Symbol 2.3.2-1)

If there are affecting inputs or outputs in the sense of dependency notation (Section 4) within the elements of the array, it is assumed that the identifying numbers of those inputs or outputs and of inputs or outputs affected thereby differ in each element of the array.

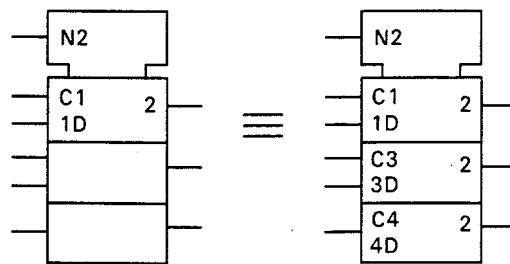
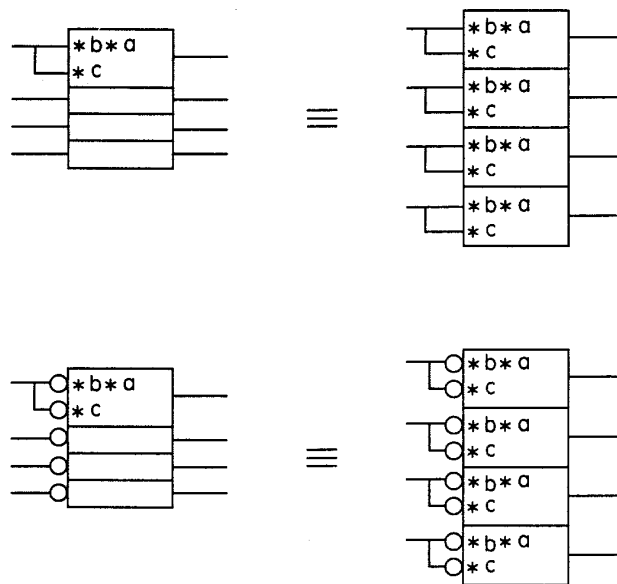


Illustration of the concept

2.3.1.2

If, in a simplified array of identical elements, the representation of the functions of a terminal requires two or more lines connected together outside the outline, it is sufficient to show these lines with only the first element and represent them by a single line with each simplified element. Symbols outside the outline common to all the lines connected together shall be shown with this single line. Symbols outside the outline not common to all the lines connected together may be omitted, or the more suitable set may be shown.



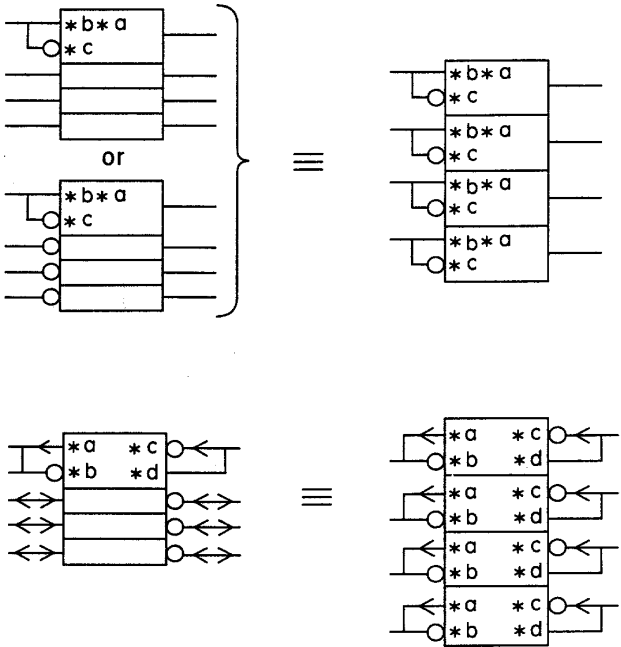


Illustration of the concept

2.3.2 Common control block

The common control block may be used in conjunction with an array of related elements as a point of placement for inputs or outputs associated with more than one element of the array, or with no element of the array. Such inputs and outputs shall be labeled if appropriate.

When an input shown at a common control block is an affecting input in the sense of dependency notation, it is connected as an input only to those elements of the array in which its identifying number appears. When an input shown at a common control block is not an affecting input in the sense of dependency notation, it is an input common to or affecting all elements of the array.

The common control block may be placed at either end of an array of related elements.

No	Symbol	Description
2.3.2-1		Common control block outline

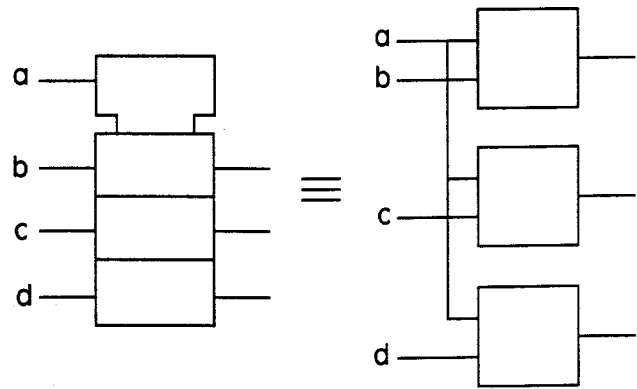


Illustration of the concept

2.3.2.1 Weighted arrays

If the elements in an array have an intrinsic weighted relationship (for example, as in an up/down counter) and it is not otherwise indicated, the element next to the common control block is assumed to be the element with the smallest weight.

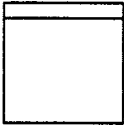
2.3.3 Common output element

A common output depending on all elements of the array can be shown as the output of a common output element. In the case where any array element has more than one output, the common output element may be used only if those outputs always have identical internal logic states. There is one internal connection from each of the elements to the common output element and these shall not be shown. In addition, the common output element may have other inputs and they shall be explicitly shown. The function of the common output element shall be indicated.

Each input of a common output element that is an output of the array has the same internal logic state as that output.

A common output element shall be shown either

- 1) Inside a common control block, or
- 2) At either end of the array, but opposite the common control block if there is one.

No	Symbol	Description
2.3.3-1		Common output element outline NOTE — When embedded in a common control block, the double line shall be placed at the edge of the element away from the array to which the control block is connected. Otherwise, the double line shall be placed at the edge nearest the array.

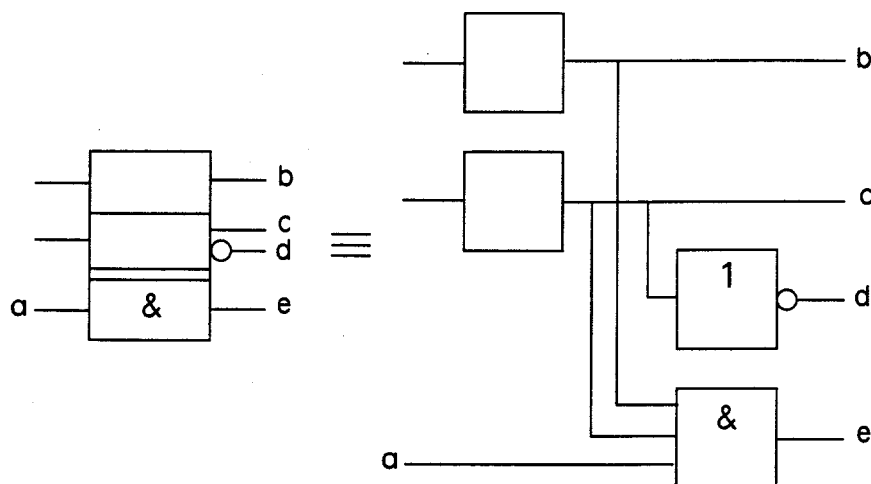
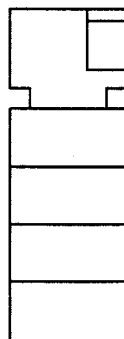


Illustration of the concept

NOTE — The common output element, like any element, must have at least one qualifying symbol to indicate its function. The “&” (Symbol 5.1-3) is the qualifying symbol for AND.

Array with common control block and common output element:



2.3.3.1 Array of common output elements

If it is appropriate to show an array of common output elements, the double line need be shown only once if no confusion is likely. In this case, there is one internal connection from each of the elements of the array to each of the common output elements. There is no connection between common output elements unless explicitly shown.

Array with two common output elements:

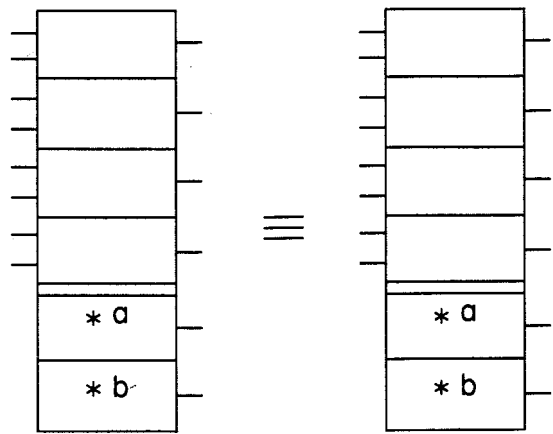


Illustration of the concept

3. Qualifying symbols associated with inputs, outputs, and other connections

3.1 Negation, polarity, and dynamic input symbols

In this section qualifying symbols are shown that indicate the relationship between internal logic state and external logic state or level.

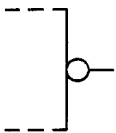
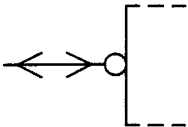
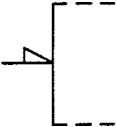
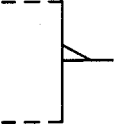
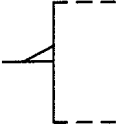
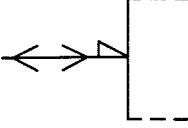
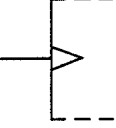
If none of the qualifying symbols of this section is shown at an input or output, it is assumed that the internal logic 1-state corresponds with

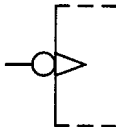
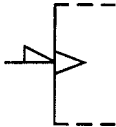
- 1) The external logic 1-state in a diagram using the positive logic convention or the negative logic convention, or
- 2) The logic H-level in a diagram using direct polarity indication

3.1.1

The qualifying symbols for negation and polarity shall not be used on the same diagram to relate internal logic states to external logic states or levels. However, the polarity symbol may be used on the same diagram with an internal negation symbol as defined in Symbols 3.2-2 or 3.2-4. For example, see Symbol 5.4-7.

No	Symbol	Description
3.1-1		Negated input The external 0-state produces the internal 1-state. NOTE — The connecting line may be drawn through the negation symbol.

No	Symbol	Description
3.1-2		Negated output The internal 1-state produces the external 0-state. See NOTE to Symbol 3.1-1.
3.1-3		Negated input/output port, shown on the left-hand side. See 3.4-5.
3.1-4		Active-low input The L-level on the input produces the internal 1-state NOTE — This qualifying symbol is designated the polarity symbol and shall point in the direction of signal flow.
3.1-5		Active-low input in the case of signal flow from right to left See NOTE to Symbol 3.1-4.
3.1-6		Active-low output The internal 1-state produces the L-level on the output. See NOTE to Symbol 3.1-4.
3.1-7		Active-low output in the case of signal flow from right to left See NOTE to Symbol 3.1-4.
3.1-8		Active-low input/output port, shown on the left-hand side. See 3.4-5.
3.1-9		Dynamic input The transition from the external 0-state to the external 1-state produces a transitory internal 1-state. At all other times the internal logic state is 0. On diagrams using direct polarity indication, the transition from the L-level to the H-level on the input produces a transitory internal 1-state. At all other times the internal logic state is 0.

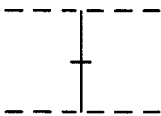
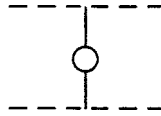
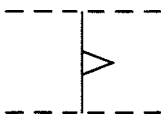
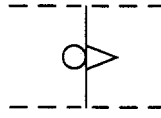
No	Symbol	Description
3.1-10		Dynamic input with negation The transition from the external 1-state to the external 0-state on the input produces a transitory internal 1-state. At all other times the internal logic state is 0.
3.1-11		Dynamic input with polarity symbol The transition from the H-level to the L-level on the input produces a transitory internal 1-state. At all other times the internal logic state is 0.

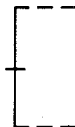
3.2 Internal connections

An internal connection is a connection within an element.

In this section, qualifying symbols are shown that indicate the relationship between internal logic states at internal connections.

The symbols in this section may be used to show the logic relationships between elements whose outlines are combined.

No	Symbol	Description
3.2-1		Internal connection The internal 1-state (0-state) of the output of the element on the left produces the 1-state (0-state) at the input of the element on the right. NOTE — The internal connection symbol may be omitted if no confusion is likely. (See also 2.3.)
3.2-2		Internal connection with negation The internal 1-state (0-state) of the output of the element on the left produces the 0-state (1-state) at the input of the element on the right.
3.2-3		Internal connection with dynamic character The transition from the internal 0-state to the internal 1-state of the output of the element on the left produces a transitory 1-state at the input of the element on the right. At all other times the logic state at the input of the element on the right is 0.
3.2-4		Internal connection with negation and dynamic character Transition from the internal 1-state to the internal 0-state of the output of the element on the left produces a transitory 1-state at the input of the element on the right. At all other times the logic state at the input of the element on the right is 0.

No	Symbol	Description
3.2-5		Internal input (virtual input) This input always stands at its internal 1-state unless it is affected by a dependency relationship that has an overriding effect. (See Symbols 5.9-13 and 5.13-3.) NOTES: 1 — Internal inputs and outputs have internal logic states only. 2 — The qualifying symbols of 3.1 shall not be applied to internal inputs and outputs, except for the dynamic input symbol. 3 — This symbol should not be confused with Symbol 3.2-1, which is used for abutted elements.
3.2-6		Internal output (virtual output) The effect of this output must be indicated by dependency notation. See NOTES to Symbol 3.2-5.

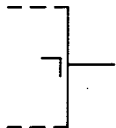
3.3 Symbols inside the outline

3.3.1

If two or more inputs have the same qualifying symbol for their functions, they are assumed to stand in an OR relationship, except in the cases of bithreshold and extension inputs (Symbols 3.2-2 and 3.3-10) where the relationship should be appropriately indicated.

3.3.2

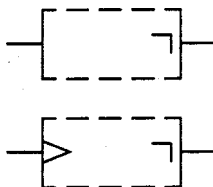
The descriptions of Symbols 3.3-14 through 3.3-22 may give the reader the impression that these are always dynamic inputs. This is not the case, as it must be remembered that the internal logic state, as determined by the external logic state or level, may be modified by the state of other inputs (for example, Cm inputs, 4.3.7). If such inputs have a dynamic character, Symbol 3.1-9 should be added (for example, Symbol 5.13-13).

No	Symbol	Description
3.3-1		Postponed output The change of the internal state of this output is postponed until the input signal that initiates the change returns to its initial external logic state or logic level. The internal logic state of any input(s) affecting or affected by the <i>initiating</i> input must not change while this <i>initiating</i> input stands at its internal 1-state or the resulting output state will not be specified by the symbol. If the input signal that initiates the change appears at an internal connection, the change of state is postponed until the output at the internal connection returns to its initial logic state. NOTES: 1 — If the postponed output symbol is shown without a prefix, it should be assumed that the output is postponed with respect to each Cm, +, →, ←, -, or T input (for the Cm input see 4.3.7); in all other cases the identifying numbers (or if necessary the full labels) of all inputs with respect to which the output is postponed must be shown as a prefix to this symbol (for example, Symbol 5.13-17). 2 — The postponed output symbol should be a right angle with legs of equal length to avoid confusion with other symbols, for example, the number 7. 3 — For the application of this symbol and additional explanation, see 5.9.

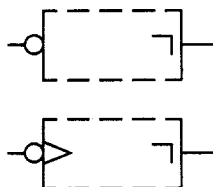
The following illustrations assume there are no other inputs with overriding effects.

When using the positive or the negative logic convention, the transition at the output takes place when the input changes:

- 1) From its external 1-state to its external 0-state, or

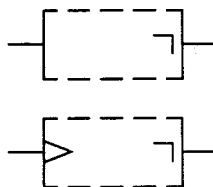


- 2) From its external 0-state to its external 1-state

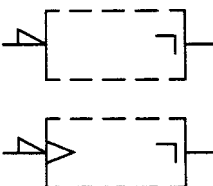


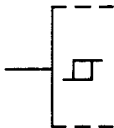
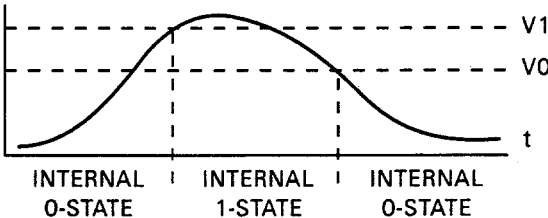
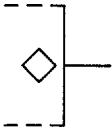
On a diagram using direct polarity indication, the transition at the output takes place when the input changes:

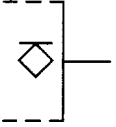
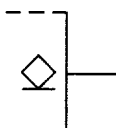
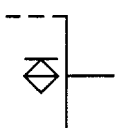
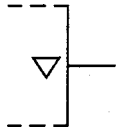
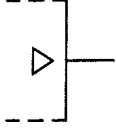
- 1) From its H-level to its L-level, or

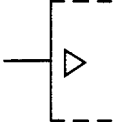
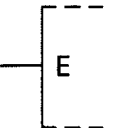
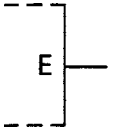


- 2) From its L-level to its H-level

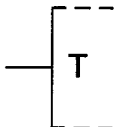
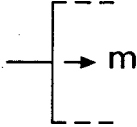
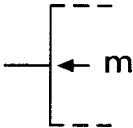
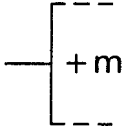
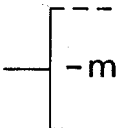
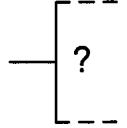
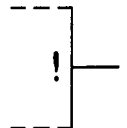


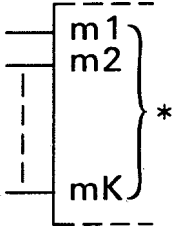
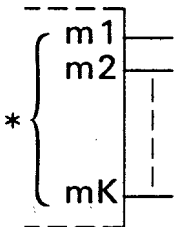
No	Symbol	Description
3.3-2	 EXTERNAL SIGNAL (POSITIVE-LOGIC CONVENTION, NO NEGATION SYMBOL)	Input with hysteresis Bithreshold input The input takes on its internal 1-state when the external signal level reaches a threshold value, V_1. It maintains this state until the external signal level has returned through V_1 and reaches another threshold value, V_0. If this symbol (without the negation symbol or polarity symbol) appears on a diagram that uses either direct polarity indication or the positive-logic convention, V_1 is more positive than V_0. If it appears on a diagram that uses the negative-logic convention, V_1 is more negative than V_0. If the negation symbol or polarity symbol is present, then the relationship between V_1 and V_0 is reversed.  INTERNAL 0-STATE INTERNAL 1-STATE INTERNAL 0-STATE NOTES: 1 — When used with a dynamic input symbol (Symbols 3.1-9 through 3.1-11), the hysteresis symbol shall be shown following the dynamic input symbol even though its effect occurs first. For use with the special amplification symbol (Symbol 3.3-9), the line-grouping symbol (Symbol 3.3-37), or dependency notation, see 4.4.3. 2 — Most practical devices exhibit this characteristic to some extent. This symbol is only used when an identification of the characteristic is important to the application of the device. For use as a general qualifying symbol, see 5.3.
3.3-3		Open-circuit output (for example, open-emitter, open-collector, open-source, or open-drain output) One of the two possible internal logic states of this type of output corresponds to an external high-impedance condition. In order to produce the proper logic level in this condition, an externally connected component, often a resistor, is required. This type of output is usually capable of forming part of a distributed function. NOTES: 1 — This symbol shall be drawn adjacent to but not touching the output line, except when using the bit-grouping symbol (Symbol 3.3-26) in the manner defined in 4.4.4, which permits an alternative position. 2 — Although this symbol is shown inside the outline, it refers to external states and levels only. The meaning of this symbol is not altered by the presence of a negation or polarity symbol (Symbol 3.1-2, 3.1-6, or 3.1-7). 3 — To indicate which logic level is the one with the low impedance, use of Symbol 3.3-4 or 3.3-5 is recommended. 4 — If all outputs in a device family (e.g., 10 K ECL) are of the same open-circuit type, and if no confusion is likely, this symbol may be omitted from all outputs in the family.

No	Symbol	Description
3.3-4		Open-circuit output (H-type) (for example, PNP open-collector, NPN open-emitter, P-channel open-drain, or N-channel open-source) When not in its external high-impedance state, this type of output produces a relatively low-impedance H-level. When used as part of a distributed function, a positive-logic OR function or a negative-logic AND function is performed. See also Symbols 5.1-15 and 5.1-16. See NOTES 1, 2, and 4 to Symbol 3.3-3.
3.3-5		Open-circuit output (L-type) (for example, NPN open-collector, PNP open-emitter, N-channel open-drain, or P-channel open-source) When not in its external high-impedance state, this type of output produces a relatively low-impedance L-level. When used as part of a distributed function, a positive-logic AND function or a negative-logic OR function is performed. See also Symbols 5.1-15 and 5.1-16. See NOTES 1, 2, and 4 to Symbol 3.3-3.
3.3-6		Passive-pulldown output This type of output (like the H-type open-circuit output, Symbol 3.3-4) can be used as part of a distributed function to perform a positive-logic OR function or a negative-logic AND function but it produces both the H-level and the L-level without the need for an additional external component. See also Symbols 5.1-15 and 5.1-16. See NOTES 1 and 2 to Symbol 3.3-3.
3.3-7		Passive-pullup output This type of output (like the L-type open-circuit output, Symbol 3.3-5) can be used as part of a distributed function to perform a positive-logic AND function or a negative-logic OR function but it produces both the H-level and the L-level without the need for an additional external component. See also Symbols 5.1-15 and 5.1-16. See NOTES 1 and 2 to Symbol 3.3-3.
3.3-8		3-state output This output can take on a third external state, which is a high-impedance state, having no logic significance. For example, see Symbol 5.2-4. See NOTES 1 and 2 to Symbol 3.3-3.
3.3-9		Output with special amplification (drive capability) The symbol ▷ emphasizes the function of amplification. It shall point in the direction of signal flow. The absence of this symbol does not necessarily indicate the absence of special amplification. 5.2 describes its use as a general qualifying symbol for an element. See NOTE 1 to Symbol 3.3-3. NOTE — If this symbol is used with Symbols 3.3-3 through 3.3-8, those symbols are placed between the amplification symbol and the edge of the element.

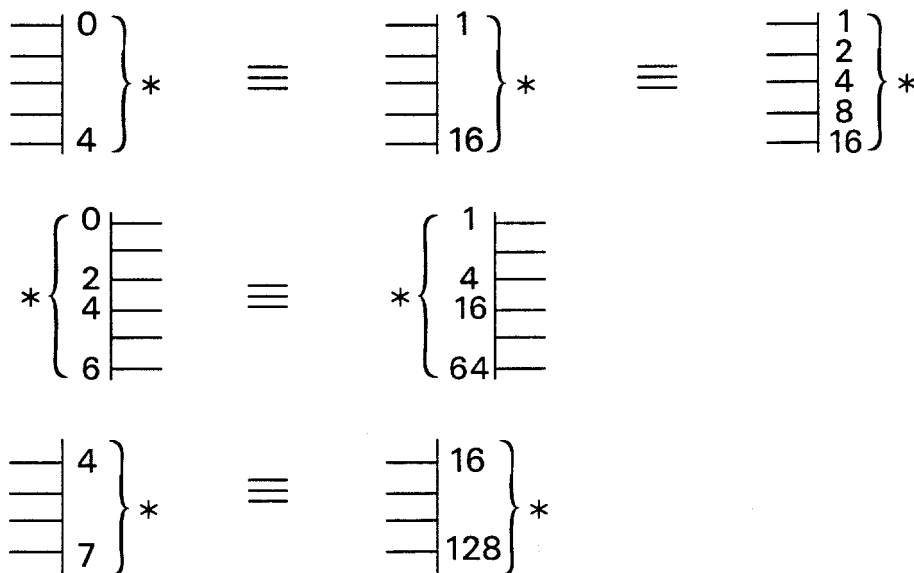
No	Symbol	Description
3.3-9.5		Input with special amplification (sensitivity) The symbol ▷ emphasizes the function of amplification. It shall point in the direction of signal flow. The absence of this symbol does not necessarily indicate the absence of special amplification. 5.2 describes its use as a general qualifying symbol for an element. Its use at an input, rather than as a general qualifying symbol, shows that the input is unusually sensitive rather than that the output has increased drive capability. NOTE — For use with the hysteresis symbol (Symbol 3.3-2), the line-grouping symbol (Symbol 3.3-37), or dependency notation, see 4.4.3.
3.3-10		Extension input An input of an element to which an extender output may be connected (see Symbol 3.3-11). NOTE — The description that characterizes the relationship between the external logic states of binary variables and their corresponding physical quantities is normally not valid for extension inputs and extender outputs.
3.3-11		Extender output An output of an element that may be connected to the extension input of another combinational element in order to extend the number of inputs to that element. See NOTE to Symbol 3.3-10.
3.3-12		Enable input When this input stands at its internal 1-state, all outputs stand at their normally defined internal logic states and have their normally defined effect on elements or distributed functions that may be connected to the outputs, provided no other inputs or outputs have an overriding and contradicting effect. When this input stands at its internal 0-state, all open-circuit outputs stand at their external high-impedance states, all passive-pulldown outputs stand at their high-impedance L-levels, all passive-pullup outputs stand at their high-impedance H-levels, all 3-state outputs stand at their normally defined internal logic states and at their external high-impedance states, and all other outputs stand at their internal 0-states. In a composite symbol this input affects all outputs that are not shown as internal connections or internal outputs (making use of one of the Symbols 3.2-1 through 3.2-6). See also 4.3.9. The “composite symbol” excludes outputs that are not otherwise shown to be influenced by the unsubdivided portion of the symbol containing the EN Input.

No	Symbol	Description
	a affects b but not c and d.	
	e affects f, g, and h but not i.	
	j affects l and m but not n. If k is also an EN input, it will also affect m as well as n but not l.	
3.3-13		D input The internal logic state of the D input is stored by the element. NOTE — This symbol is used only in conjunction with dependency notation. For example, see Symbol 5.9-2.
3.3-14		J input of a JK bistable element When this input takes on its internal 1-state, a 1 is stored by the element. When the input stands at its internal 0-state, it has no effect on the element. Each occurrence of the combination J=K=1 causes a single change of the internal state of the output to its complement. See 3.3.2.
3.3-15		K input of a JK bistable element When this input takes on its internal 1-state, a 0 is stored by the element. When the input stands at its internal 0-state, it has no effect on the element. Each occurrence of the combination J=K=1 causes a single change of the internal state of the output to its complement. See 3.3.2.
3.3-16		R input When this input takes on its internal 1-state, a 0 is stored by the element. When the input stands at its internal 0-state, it has no effect on the element. The effect of the combination R=S=1 is not specified by the symbol; this effect may be indicated by means of set/reset dependency (see 4.3.8), if necessary. See 3.3.2.
3.3-17		S input When this input takes on its internal 1-state, a 1 is stored by the element. When the input stands at its internal 0-state, it has no effect on the element. The effect of the combination R=S=1 is not specified by the symbol; this effect may be indicated by means of set/reset dependency (see 4.3.8), if necessary. See 3.3.2.

No	Symbol	Description
3.3-18		T input of a bistable element Each time this input takes on its internal 1-state, a single change of the internal state of the output to its complement takes place. When the input stands at its internal 0-state, it has no effect on the element. See 3.3.2.
3.3-19		Shifting input of a register Each time an input with a right-pointing arrow takes on its internal 1-state, the information contained in the element will be shifted once m positions from left to right or from top to bottom. When the input stands at its internal 0-state, it has no effect on the element. See 3.3.2. NOTES: 1 — m must be replaced by the relevant value. If m = 1, the 1 may be omitted if no confusion is likely. 2 — All directions above are relative to the orientation of the text within the symbol.
3.3-20		Shifting input of a register Each time an input with a left-pointing arrow takes on its internal 1-state, the information contained in the element will be shifted once m positions from right to left or from bottom to top. When the input stands at its internal 0-state, it has no effect on the element. See 3.3.2, also NOTE 1 to Symbol 3.3-19.
3.3-21		Count-up input Each time this input takes on its internal 1-state, the content of the element is increased once by m. When the input stands at its internal 0-state, it has no effect on the element. See 3.3.2, also NOTE 1 to Symbol 3.3-19.
3.3-22		Count-down input Each time this input takes on its internal 1-state, the content of the element is decreased once by m. When the input stands at its internal 0-state, it has no effect on the element. See 3.3.2, also NOTE 1 to Symbol 3.3-19.
3.3-23		Query input (interrogate) of an associative memory When this input takes on its internal 1-state, an interrogation of the content of the element takes place. When the input stands at its internal 0-state, it has no effect on the element.
3.3-24		Compare output (match) of an associative memory. The internal 1-state at this output indicates a match.

No	Symbol	Description
3.3-25		Bit-grouping symbol for inputs (qualifying symbol for multibit input), general symbol Inputs grouped by this symbol produce a value that is the sum of the individual weights of the inputs standing at their internal 1-states. The individual inputs shall be shown in ascending or descending order by weight. This value can be regarded as: 1) A value on which a mathematical function is performed, or 2) Defining an identifying number in the sense of dependency notation according to 4.4.2, or 3) A value to become the content of the element. The labels m1...mk shall be replaced by the decimal equivalents of the actual weights. If all the weights are powers of 2, m1...mk may be replaced by the exponents of the powers of 2 if no confusion is likely. Labels between m1 and mk may be omitted to the extent that no confusion is likely. The “*” must be replaced by an indication of the operand on which the mathematical function is performed (for example, P or Q), by an indication in the sense of dependency notation (see 4.4.2), by CT, or by an appropriate label in the case of a “gray box” (Section 6). In the case of CT the value is produced by the inputs is the value that is loaded into the element. For example, see Symbol 5.7-5.
3.3-26		Bit-grouping symbol for outputs (qualifying symbol for multibit output), general symbol Outputs grouped by this symbol represent a value that is the sum of the individual weights of the outputs standing at their internal 1-states. The individual outputs shall be shown in ascending or descending order by weight. This value can be regarded as: 1) The result of the performance of a mathematical function, or 2) The value of the content of the element. The labels m1...mk shall be replaced by the decimal equivalents of the actual weights. If all the weights are powers of 2, m1...mk may be replaced by the exponents of the powers of 2 if no confusion is likely. Labels between m1 and mk may be omitted to the extent that no confusion is likely. The “*” must be replaced by an indication of the result of the performance of the mathematical function, by CT, or by an appropriate label in the case of a “gray box” (Section 6). In the case of CT the value represented by the outputs is the actual value of the content of the element. For example, see Symbol 5.13-18.

Illustrations of the concept for Symbols 3.3-25 and 3.3-26



No	Symbol	Description
3.3-27		Operand input of an arithmetic element (P input shown) This input represents one bit of an operand on which one or more mathematical functions are performed. NOTES: 1 — m shall be replaced by the decimal equivalent of the weight of the bit. If the weights of all P inputs of the element are powers of 2 and if no confusion is likely, at each P input m may be replaced by the exponent of the power of 2. 2 — If an operand consists of two or more bits represented by adjacent input lines, the bit-grouping symbol (Symbol 3.3-25) may be used. 3 — Preferred letters for operands are P and Q. When these letters are not suitable or if more than two operands are involved, other characters may be used providing no confusion is likely.
3.3-28		Greater-than input of a magnitude comparator NOTES: 1 — This symbol is intended for use when representing cascaded comparators. For example, see Symbol 5.7-11. 2 — Symbols 3.3-28, 3.3-29, and 3.3-30 may be combined to qualify other inputs of magnitude comparators such as $\geq$. The symbol $\neq$ shall be used instead of the combination $\leq$. 3 — Care should be taken that this symbol is not drawn touching the outline to avoid confusion with the dynamic input indicator (Symbol 3.1-9).

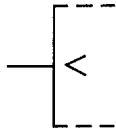
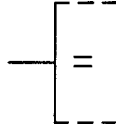
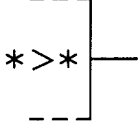
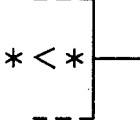
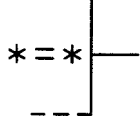
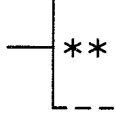
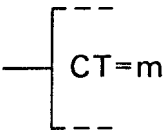
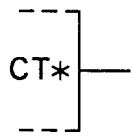
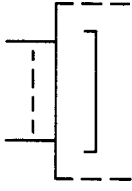
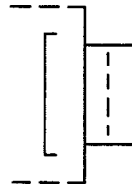
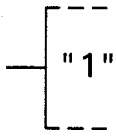
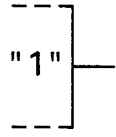
No	Symbol	Description
3.3-29		Less-than input of a magnitude comparator See NOTES 1 and 2 to Symbol 3.3-28.
3.3-30		Equal input of a magnitude comparator See NOTES 1 and 2 to Symbol 3.3-28.
3.3-31		Greater-than output of a magnitude comparator NOTES: 1 — Each “*” must be replaced by the designation of an operand (for example, P or Q). 2 — Symbols 3.3-31, 3.3-32, and 3.3-33 may be combined to qualify outputs such as $* \geq *$. The symbol $* \neq *$ shall be used instead of the combination $* \lesseqgtr *$. 3 — If this symbol appears in one element of a series of cascaded comparators, the output marked with this symbol is affected not only by the operands, but also by inputs marked with the Symbol 3.3-28, 3.3-29, or 3.3-30.
3.3-32		Less-than output of a magnitude comparator See NOTES to Symbol 3.3-31.
3.3-33		Equal-output of a magnitude comparator See NOTES to Symbol 3.3-31. NOTE — If this symbol is not combined with Symbol 3.3-31 or Symbol 3.3-32, the designations of the operands may be omitted if no confusion is likely.
3.3-34		Symbols for commonly used logic signals (shown at an input) NOTES: 1 — Table Clause lists and describes symbols available for designating commonly used logic signals. 2 — Each “**” symbol must be replaced by an appropriate alphabetic signal designator. 3 — A numerical suffix may be added to this alphabetic designator, if appropriate, to indicate the weight or position of the signal within a group of related signals.

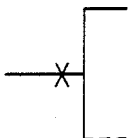
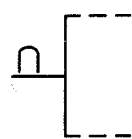
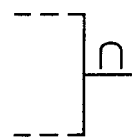
Table 3.1—Commonly used logical signal designations

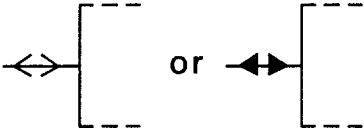
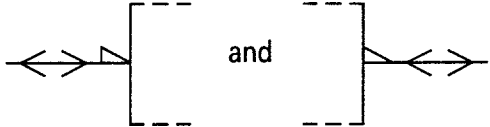
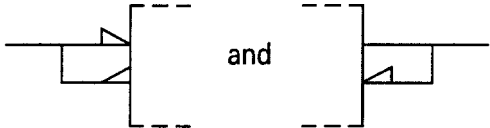
Designation	Input/Output	Description
BI (Borrow input)	Input only	If at its internal 1-state, indicates that a subtraction operation performed by a lower-order arithmetic element produces an arithmetic borrow.
BO (Borrow output)	Output only	If at its internal 1-state, indicates that a subtraction operation performed by an arithmetic element produces an arithmetic borrow (see BI signal above).
BG (Borrow generate)	Output	If at its internal 1-state, indicates that an arithmetic element performing subtraction is in the “Borrow-Generate” state, that is, that the subtrahend applied to the element is larger than the minuend, causing a borrow from that element independent of the state of the BI input to that element.
	Input	If at its internal 1-state, indicates to a borrow-acceleration element that the arithmetic element that produces the BG signal is in the “Borrow-Generate” state (see BG output above). The borrow-acceleration element uses its BG, BP, and BI input signals to determine, with reduced propagation delays, the states of the arithmetic borrow signals for a group of arithmetic elements performing binary subtraction.
BP (Borrow propagate)	Output	If at its internal 1-state, indicates that an arithmetic element performing subtraction is in the “Borrow-Propagate” state, that is, that the subtrahend and the minuend applied to the element are equal in value, so that the BO output will stand at its internal 1-state if and only if the BI input is at its internal 1-state.
	Input	If at its internal 1-state, indicates to a borrow-acceleration element that the arithmetic element that produces the BP signal is in the “Borrow-Propagate” state.
CI (Carry input)	Input only	If at its internal 1-state, indicates that an addition operation performed by a lower-order arithmetic element produces an arithmetic carry.
CO (Carry output)	Output only	If at its internal 1-state, indicates that an addition operation performed by an arithmetic element produces an arithmetic carry (see CI signal above).
CG (Carry generate)	Output	If at its internal 1-state, indicates that an arithmetic element performing addition is in the “Carry-Generate” state, that is, that the sum of its addends is sufficiently large to cause a carry from the element independent of the state of the CI input to that element.
	Input	If at its internal 1-state, indicates to a carry-acceleration element whether or not the arithmetic element that produces the CG signal is in the “Carry-Generate” state (see CG output above). The carry-acceleration element uses its CG, CP, and CI input signals to determine with reduced propagation delays, the states of the arithmetic carry signals for a group of arithmetic elements performing addition.
CP (Carry propagate)	Output	If at its internal 1-state, indicates that an arithmetic element performing addition is in the “Carry-Propagate” state, that is, that the sum of its addends is one less than the value at which the element produces an output carry. As a result, the CO output will stand at its internal 1-state if and only if its CI input is at its internal 1-state.
	Input	If at its internal 1-state, indicates to a carry-acceleration element that the arithmetic element that produces the CP signal is in the “Carry-Propagate” state.

No	Symbol	Description
3.3-35		Content input m shall be replaced by an appropriate indication of the value of the content of the element (for example, a counter) that results whenever this input takes on its internal 1-state. For example, see Symbol 5.13-18.
3.3-36		Content output “*” shall be replaced by an appropriate indication of those values of the content of the element (for example, a counter) for which the output stands at its internal 1-state. For example, see Symbol 5.13-14.
3.3-37		Line-grouping symbol for inputs This symbol indicates that two or more terminals are needed to implement a single logic input. For example, see Symbol 5.1-22. NOTES: 1 — The description that characterizes the relationship between the external logic states of binary variables and their corresponding physical quantities may not be valid for lines grouped by symbol. 2 — For use with the hysteresis symbol (Symbol 3.3-2), the special amplification symbol (Symbol 3.3-9.5), or dependency notation, see 4.4.3.
3.3-38		Line-grouping symbol for outputs This symbol indicates that two or more terminals are needed to implement a single logic output. See NOTE to Symbol 3.3-37. For example, see Symbol 5.1-23.
3.3-39		Fixed-mode input This representation may be used to identify an input that always must be in the internal 1-state for the element to perform the function indicated by the complete symbol. A fixed-mode input must neither be affected by dependency notation nor have other functions. See Symbols 5.13-9 and 5.13-10.
3.3-40		Fixed-state output This representation may be used to identify an output that always stands at its internal 1-state. A fixed-state output must neither be affected by dependency notation nor have other functions. For example, see Symbol 5.4-10.

3.4 Subsidiary (formerly nonlogic) connections and signal-flow indicators

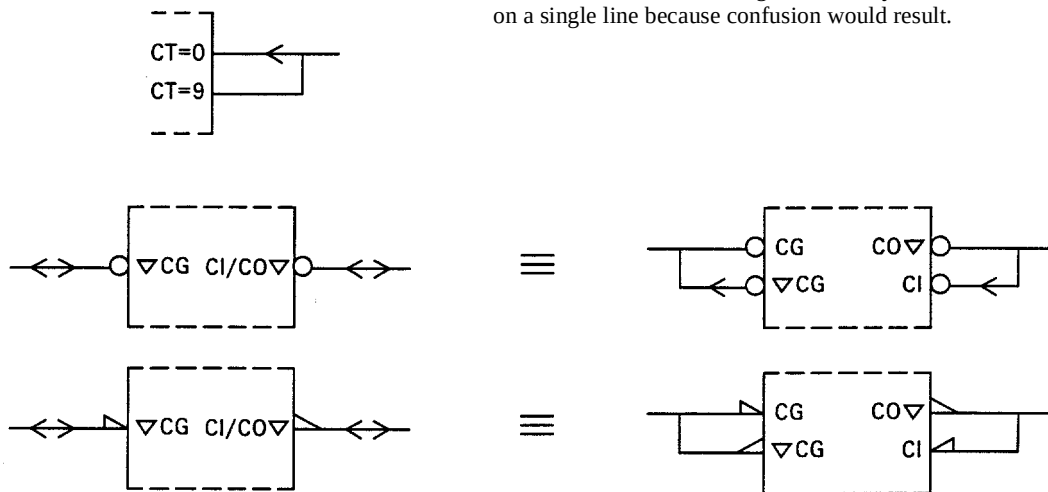
In principle, the direction of signal flow within a symbol is from left to right. If the direction of signal flow on any input or output line is not obvious, that line should be marked with an arrowhead (Symbol 3.4-4) pointing in the direction of signal flow. Such an arrowhead shall not touch the outline or any qualifying symbol.

No	Symbol	Description
3.4-1		Subsidiary connection This symbol may be used to designate an input supplying power to the device or a connection, the knowledge of whose level is not important to understand the function of the element and the circuit (such as a connection to an external supplementary resistor or capacitor). NOTES: 1 — This symbol may be shown on any side of the outline (including the top and bottom). 2 — Additional information associated with subsidiary connections may be shown without brackets inside the outline. See Symbol 5.12-7. 3 — In a symbol representing an array of elements, any subsidiary connection shown at either end of the array is considered to be common to all elements of the array. See Symbol 5.12-8.
3.4-2		Analog input NOTE — This symbol shall be used only when it is necessary to distinguish analog signals. See NOTE 2 to Symbol 3.4-1.
3.4-3		Analog output See NOTE 2 to Symbol 3.4-1 and NOTE to Symbol 3.4-2.
3.4-4		Right-to-left signal flow, shown on the left-hand side

No	Symbol	Description
3.4-5		Bidirectional signal flow, shown on the left-hand side For an example of use, see Symbol 5.2-8. NOTES: 1 — On a connecting line with bidirectional signal flow, any symbol for logic polarity should point to the right or down, e.g.,  2 — In some cases, use of the symbol for logic polarity makes the use of the symbol for bidirectional signal flow unnecessary, e.g.,  3 — Bidirectional signal flow shall not be shown on a single input-output line: —if the negation or polarity indications are different for the input and output functions (except as permitted by 2.3.1.2), or —if the labels associated with the input and output functions are different and confusion is likely regarding which portion of the labels applies to the input and which to the output.

Illustrations of the concept

In this case, bidirectional signal flow may not be shown on a single line because confusion would result.



4. Dependency notation

4.1 General explanation

Dependency notation is a means of denoting the relationships between inputs, outputs, or inputs and outputs, without actually showing all the elements and interconnections involved.

Apart from its use in complex elements, dependency notation should not be used to replace the symbols for combinational elements.

The information provided by dependency notation supplements that provided by the qualifying symbols for an element's function.

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

Because of the existence of feedback in some complex elements, outputs sometimes have an effect on inputs and other outputs. For the sake of simplicity, the text of the following sections often refers to “affecting inputs” only, but it should be understood that the recommended notation applies to affecting outputs also.

4.2 Summary of types of dependencies

Eleven types of dependencies are defined.

AND, OR, and Negate dependencies are used to denote Boolean relationships between inputs and outputs, in any combination.

Interconnection dependency is used to indicate connections inside the symbol.

Transmission dependency is used to indicate controlled transmission paths between affected ports.

Control dependency is used to identify a timing input or a clock input of a sequential element and to indicate which inputs are controlled by it.

Set and reset dependencies are used to specify the internal logic states of an RS bistable element when the R and S inputs both stand at their internal 1-states.

Enable dependency is used to identify an enable input and to indicate which inputs and outputs are controlled by it (for example, which outputs take on their high-impedance state).

Mode dependency is used to identify an input that selects the mode of operation of an element and to indicate the inputs and outputs depending on that mode.

Address dependency is used to identify the address inputs of a memory.

Table Clause lists the various dependencies and summarizes their effects. More detailed definitions appear in the following sections, together with illustrations of the concept. In this Table the word “action” implies:

- 1) That affected inputs will have their normally defined effect on the function of the element, and
- 2) That affected outputs will take on the internal logic states determined by the function of the element.

Table 4.1—Summary of dependency notation

Type of dependency	Letter*	Effect on internal logic state of, or action of, the affected input or output:	
		Affecting input at its 1-state	Affecting input at its 0-state
ADDRESS	A	Permits action (address selected)	Prevents action (address not selected)
CONTROL	C	Permits action	Prevents action
ENABLE	EN	Permits action	1)Prevents action of affected inputs 2)Imposes external high-impedance state on open-circuit and 3-state outputs (internal state of 3-state output is unaffected) 3)Imposes high-impedance L-level on passive-pulldown outputs and high-impedance H-level on passive-pullup outputs 4)Imposes 0-state on other outputs
AND	G	Does not alter state (permits action)	Imposes 0-state
MODE	M	Permits action (mode selected)	Prevents action (mode not selected)
NEGATE	N	Complements state	Does not alter state (no effect)
RESET	R	Affected output reacts as it would to S=0, R=1	No effect
SET	S	Affected output reacts as it would to S=1, R=0	No effect
OR	V	Imposes 1-state	Does not alter state (permits action)
TRANSMISSION	X	Transmission path established	No transmission path established
INTER-CONNECTION	Z	Imposes 1-state	Imposes 0-state

*These letters appear at the affecting input (or output) and are followed by a number represented in the general cases in Section 4 by the letter "m." Each input or output affected by that input (or output) is labeled with that same number.

4.2.1 Comparison of C, EN, and M effects on inputs

Cm, ENm, and Mm inputs all have the same effect on affected inputs. However, their intended applications are different:

- 1) Cm should be used to identify an input that produces action, for example, the edge-triggered clock of a bistable circuit or the level-operated data enable of a transparent latch
- 2) ENm should be used to identify an input that produces a single preparatory effect
- 3) Mm should be used to identify one or more inputs that singly or together produce alternative preparatory effects

4.3 Application of dependency notation

4.3.1 General techniques and conventions

Dependency notation usually defines relationships between internal logic states. However, in the case of open-circuit outputs, passive-pullup and passive-pulldown outputs, and 3-state outputs (3.3-3 through 3.3-8), enable dependency (see 4.3.9) defines relationships between the internal logic states of affecting inputs and the external states of affected outputs.

Application of dependency notation is accomplished by:

- 1) Labelling the input affecting other inputs or outputs with the letter symbol denoting the relationship involved followed by an appropriately chosen identifying number, and
- 2) Labelling each input or output affected by the affecting input with that same number.

If it is the complement of the internal logic state of the affecting input or output that does the affecting, a bar is placed over the identifying number at the affected input or output (see Symbol 5.9-12). For a technique avoiding the use of the bar, see the note with symbol 5.1-5.

- If only an in-line notation can be used, the symbol $\neg$ shall be used instead of the negation bar, and it shall precede the identifying number. The tilde ($\sim$) may be substituted on computer systems that do not have the logic negation symbol as part of their character set.

If the affected input or output requires a label to denote its function, this label shall be prefixed by the identifying number of the affecting input.

If an input or output is affected by more than one affecting input, the identifying numbers of each of the affecting inputs shall appear in the label of the affected one separated by commas. The left-to-right reading order of these identifying numbers is the same as the sequence of the affecting relationships (see also 4.4.3 and 4.4.4).

Two affecting inputs labelled with different letters shall not have the same identifying number, unless one of the letters is A (see 4.3.11).

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

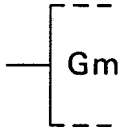
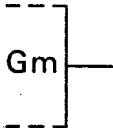
If the labels denoting the functions of affected inputs or outputs must be numbers (for example, outputs of a coder), the identifying numbers to be associated with both affecting inputs and affected inputs or outputs shall be replaced by another character selected to avoid ambiguity (for example, Greek letters as in Symbol 5.4-6).

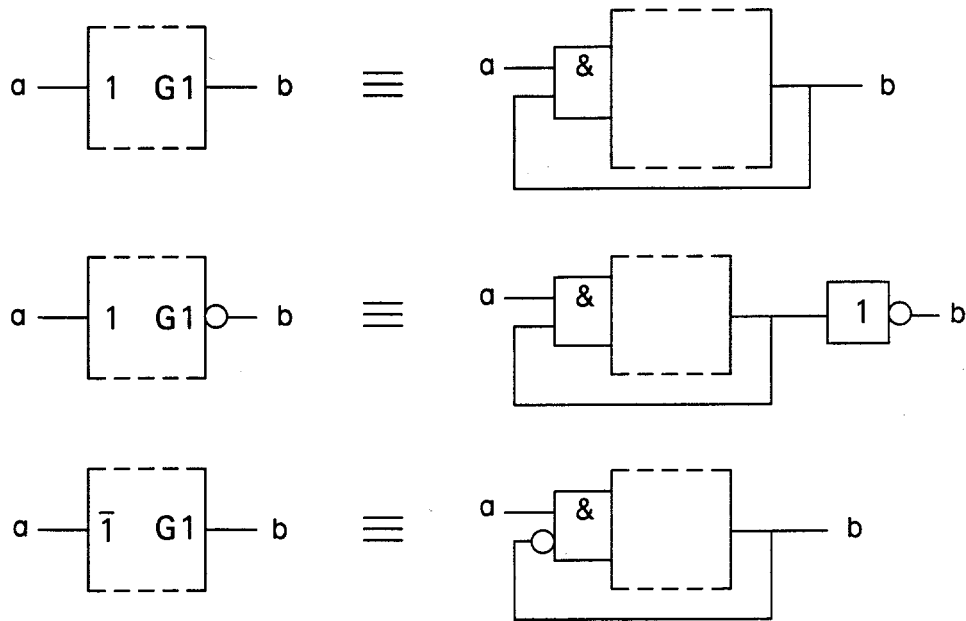
An affecting input affects only the corresponding affected inputs and outputs of the symbol.

4.3.2 AND dependency (G dependency)

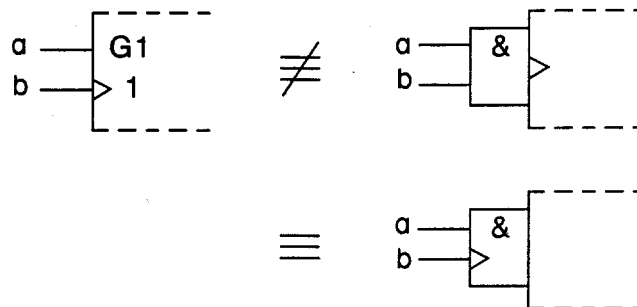
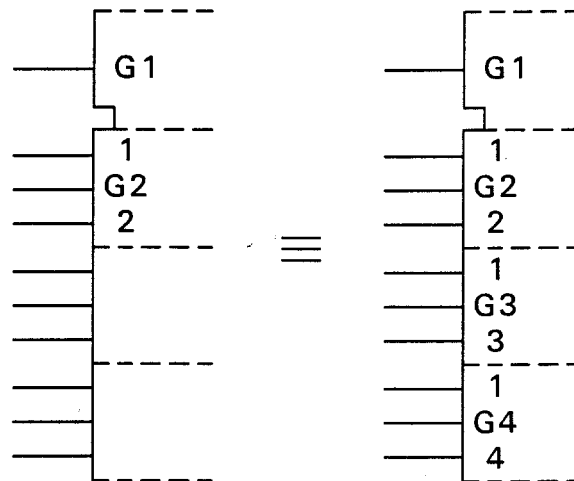
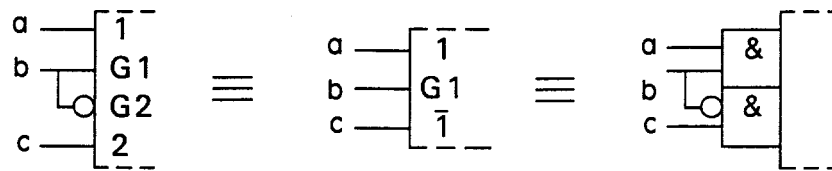
The symbol denoting AND dependency is the letter G.

Each input or output affected by a Gm input or Gm output stands in an AND relationship with this Gm input or Gm output.

No	Symbol	Description
4.3.2-1		Gm input
4.3.2-2		Gm output When a Gm input or Gm output stands at its internal 1-state, all inputs and outputs affected by this Gm input or Gm output stand at their normally defined internal logic states. When a Gm input or Gm output stands at its internal 0-state, all inputs and outputs affected by this Gm input or Gm output stand at their internal 0-states. NOTE — m must be replaced by the identifying number.



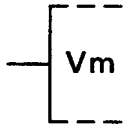
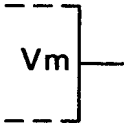
Illustrations of the concept

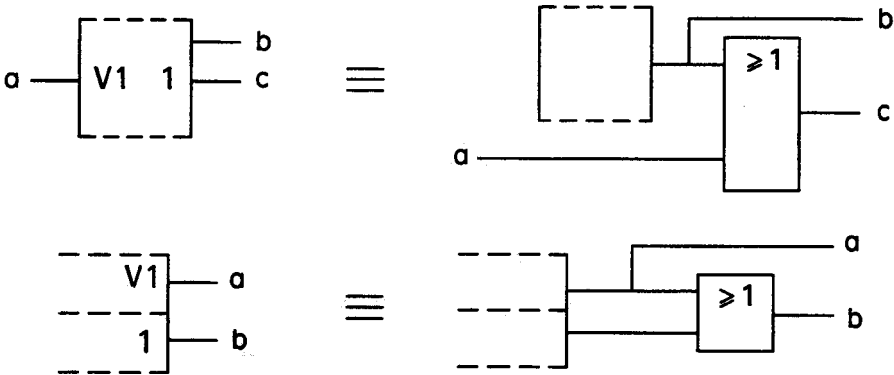


4.3.3 OR dependency (V dependency)

The symbol denoting OR dependency is the letter V.

Each input or output affected by a V_m input or V_m output stands in an OR relationship with this V_m input or V_m output.

No	Symbol	Description
4.3.3-1		Vm input
4.3.3-2		Vm output When a Vm input or Vm output stands at its internal 1-state, all inputs and outputs affected by this Vm input or Vm output stand at their internal 1-states. When a Vm input or Vm output stands at its internal 0-state, all inputs and outputs affected by this Vm input or Vm output stand at their normally defined internal logic states. See the NOTE to Symbols 4.3.2-1 and 4.3.2-2.

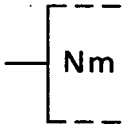
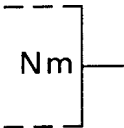


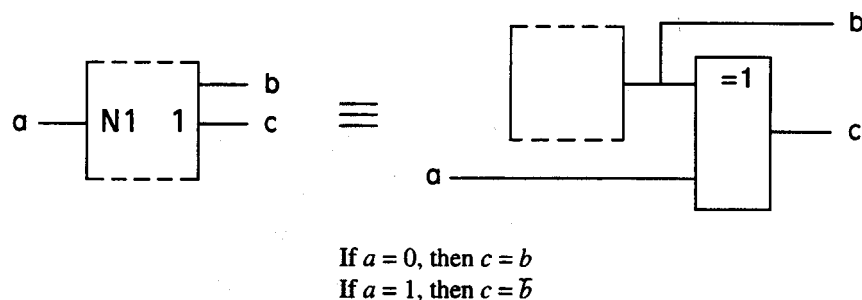
Illustrations of the concept

4.3.4 Negate dependency (N dependency)

The symbol denoting negate dependency is the letter N.

Each input or output affected by an Nm input or Nm output stands in an Exclusive-OR relationship with this Nm input or Nm output.

No	Symbol	Description
4.3.4-1		Nm input
4.3.4-2		Nm output When an Nm input or Nm output stands at its internal 1-state, the internal logic state of each input and each output affected by this Nm input or Nm output is the complement of the normally defined internal logic state of that input or output. When an Nm input or Nm output stands at its internal 0-state, all inputs and outputs affected by this Nm input or Nm output stand at their normally defined internal logic states. See the NOTE to Symbols 4.3.2-1 and 4.3.2-2.



Illustrations of the concept

4.3.5 Interconnection dependency (z 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 internal outputs, in any combination (see Symbol 5.1-29).

The internal logic state of an input or an output affected by a Zm input or Zm output is identical to the internal logic state of its affecting Zm input or Zm output, unless modified by additional dependency notation.