Computer Society has it backwards

Reference citation formats and logic diagram symbols both have IEEE Standards. Computer Society publications enforce one of them—a bad standard on citations—and ignore the other—a very good standard on logic symbols. I suggest that the readers of these publications would benefit from a change or a relaxation in the citation standard and some encouragement of the logic symbol standard.

Computer Society publications enforce thoroughly the requirement that all bibliographic citations be numbered consecutively throughout a document (article), in the order of occurrence, and be listed in numerical order at the end of the document. A far better form of citation is the author-date form as specified in [Univ. of Chicago Press 82]. This form has the following advantages over the numerical system:

1. The reader can often identify the referenced publication without turning to the end of the document.
2. Even if it is necessary to look up the reference, because of the information contained in the citation, it is usually not necessary to look up subsequent occurrences of the citation.
3. From the author’s standpoint, the author-date form permits changes in the references with minimal work. The numerical system is illusory, since authors’ names are often included in the text anyway.

The total neglect of the IEEE (also ANSI) Standard for logic symbols is much more serious. One of the primary responsibilities of the IEEE is the continued education of its members. Encouraging the use of the Logic Symbol Standard would help members to become familiar with it. Several manufacturers have already adopted the Standard for their publications—notably Texas Instruments and Signetics—and others are sure to follow. A few books use the Standard: [Peatman 80] and [McCluskey 86]. [Kampel 85] and [Mann 84] explain it, and [IEEE 84] is the Standard document.

Clarity and ease of understanding are other reasons for Computer Society publications to encourage the use of the Standard. The present policy of letting each author invent and use his own private set of symbols often means that the reader has to spend time trying to decode the symbols to understand the article. The reader may even misinterpret the article content because of a misunderstanding about the symbols used.

There is no competing standard. In fact, the IEEE Standard is the only systematic scheme for representing MSI and higher level constructs.

It should be clear by now that I think the Standard symbols should be used and also that they aren’t used very much in IEEE publications. Maybe it’s appropriate to consider why this is so.

Learning the Standard isn’t trivial. It does take some study. Without the encouragement of having to learn it in a logic design class or in order to read one’s favorite magazine, it’s easy not to bother. Also, those of us who have been working with digital logic for a number of years have developed our own set of symbols, which we are comfortable with.

I didn’t like the Standard symbols when I first saw them. It took me a fair amount of effort to learn them. As I used them, I came to like them more and more. I now use them exclusively in the logic design course I teach. I find that the students in my course take to the Standard symbols very easily and seem to like them. Perhaps, this is because they aren’t burdened with the prejudices developed over years of using some other symbols. Most are graduate students, and several are working in local industry, so they have varying amounts of experience with digital logic.

The IEEE Logic Symbol Standard is a modification of the IEC Standard and is very, very similar to the symbols used in other countries. In fact, the book [Kampel 85] discusses the IEC rather than the IEEE Standard, but the differences are minor.

The accompanying figure illustrates some of the features of the IEEE Standard symbols. Parts a, b, and c show the different types of outputs. The correspondence between the distinctive shape symbols, preferred symbols, and dependency notation is shown in parts d, e, and f. The control dependency notation for bistable elements is shown in part g and the notation to distinguish latches and flip-flops is given in parts h and i. A distinctive feature of the IEEE symbols is the common control block shown in parts j, k, l, and m. The inputs to the top box are understood to be connected to all the boxes below it. The last three parts—k, l, and m—show symbols for some of the MSI parts.

I hope this discussion will encourage the editorial management to change some policies. David McCluskey did the camera-ready IEEE Standard symbols in the figure on an Apple Macintosh.

Edward McCluskey Center for Reliable Computing ERL-460 Stanford University Stanford, CA 94305-4055

REFERENCES


Got an opinion? a joke? a cartoon? a funny story? Send all to D&T for publication in our new department “D&T Forum”
Address submissions to Managing Editor IEEE Design & Test
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
Anonymous entries are permitted.
IEEE Standard Logic Symbols

Enable
(a) Tristate Output

Tristate Output
(f) AND Dependency

Passive Pullup Output
(b) Passive Pullup Output

Passive Pulldown Output
(c) Passive Pulldown Output

Distinctive Shape Symbols
(d) Distinctive Shape Symbols

F=W'V+WX Example
(e) Preferred Symbols

Preferred Symbols

Control Dependency
(g) Control Dependency

Latch (Transparent)
(h) Latch (Transparent)

Flip-flop edge-triggered
(i) Flip-flop edge-triggered

Register with Common Control Block
(j) Register with Common Control Block

74155
(k) Dual 1-line to 4-line Demultiplexer

74153
(l) Dual 4-line to 1-line Multiplexer

SH/LOAD M1[SHIFT] M2[LOAD] C3/1+

SER 13D 23D 1D 23D 23D 23D 23D 23D 03

(m) Serial in, Parallel in, Serial out Shift Register. M(Mode) Dependency

February 1987