Practical Digital Logic

Part 2: The New Symbols

David Botto

In Part 1 last month we were mainly concerned with the basics of Boolean algebra in relation to digital circuitry. You'll also recall Pete's comment on the new logic symbols – why change the symbols and what was wrong with the old ones? The answer to this is that there's nothing wrong with the conventional symbols which should, strictly speaking, now be referred to as logic diagrams. They will continue to be used. In fact the tendency seems to be to use the old logic symbols to represent simple gates and inverters while, in the same circuit diagram, using the new logic symbols to represent the more complex digital devices. You will of course encounter some digital circuit diagrams that consist entirely of the new logic symbols. However the circuit is presented, it's obviously necessary to understand all these symbols.

Personally I like the shapes of the conventional symbols. As with an analogue watch with hands, you can see the situation at a glance. With the new logic symbols you have, at least to start with, to look hard and think about their meaning. The new symbols are a completely new way of representing logic circuitry – it's not just a case of changing the shapes of the symbols. They are used to reduce the tangle of interconnections that would otherwise be present in a circuit diagram and to give the engineer more information about the devices used in the circuit. To obtain this information with the old symbols you's normally have to consult the relevant data books. In addition there's a powerful aid known as dependency notation. This may appear to be difficult to grasp but is actually quite straightforward. We'll return to it later.

Qualifying Symbols

Before we consider the composite logic symbols themselves it's necessary to understand the meaning of the signs that accompany them. Fig. 1 shows the qualifying symbols that form an integral part of the new logic circuitry. Just to make things a little more interesting, you may find that some manufacturers use their own variations on these symbols!

Logic Gate Symbols

The new logic symbols consist of a block or blocks together with the appropriate qualifying symbols. A number of logic gates and circuits may be enclosed within the same logic symbol block. Normally all inputs feed in from the left, with the output signals taken from the right. Where this is not so the qualifying symbols shown at (39) and (40) in Fig. 1 are used.

Fig. 2 shows the new symbol for a gate with two inputs, A and B. The & sign indicates that it's an and gate while the little triangle on its side shows that the gate contains a buffer amplifier. The symbol placed near the output C tells us that the device has an open-collector npn output stage. The symbol on the outside of the box at C – see Fig. 1-36 – is an output polarity indicator showing that negative logic applies here, i.e. one is low. The complete diagram thus represents a nand gate. With positive logic (one high) the output polarity indicator shown in Fig. 1-35 would be used.

Fig. 3 shows the internal circuit of the nand gate represented in Fig. 2 – it's one section of an SN7439 quad nand gate. Notice that an external load or "pull-up" resistor is required – this is what's meant by an open-collector output.

Fig. 4 represents the complete SN7439 quad nand gate chip. Notice that the qualifying symbols are used in only the top block. This indicates that each of the three blocks below the top one is identical to it.

Common Outputs

Fig. 5(a) shows two or gates with inputs A, B, C and D driving an and gate, using conventional symbols. Fig. 5(b) shows the same circuit using the new logic symbols. The upper two blocks are identified in the top one as being or gates incorporating buffer amplifiers. The bottom block is identified by the double line as a common-output element, and is further identified as being an and gate. The common output block could have inputs other than those from the two or gates. The dotted lines show two possible additional inputs. These would make the common-output element block a four-input and gate.

Common Control Block

Fig. 6(a) shows, with conventional symbols, four exclusive-or gates. Input A is common to all four. The Boolean equation for an exclusive-or gate with inputs A and B and output C is

$$C = \overline{A}B + A\overline{B}$$

which is usually written as $A \oplus B$. Fig. 6(b) shows this circuit using the new method, with the top block representing a common input affecting all the blocks below. This shape is always used for a common control section, and it's important to remeber that the input to the common control block represents an input to each block below. The symbol = identifies the four gates as being exclusive-or devices.

Fig. 7(a) shows, with conventional symbols, the contents of an SN54LS366A chip. It has six inverting buffers plus an and gate with an inverter at each input, making it a nand gate. The buffers can't operate unless a binary one signal is present at point Z. Thus two binary zero signals are required at the inputs to the nand gate to enable the buffers. These buffers are referred to as tri-state devices: the output is either high or low or, if the buffer is not enabled, is in a high-impedance state with the logic level undefined.

Fig. 7(b) shows the arrangement using the new method. The control block is shown as including an and gate which, with inverters at its inputs, acts as a nand gate. The letters EN denote the enabling action of the control block. The little triangle on its side in the first block beneath the control block indicates a buffer amplifier while the upside-down triangle indicates that the amplifier has a tristate output. Outputs Y1-6 are shown as being inverted for logic one.

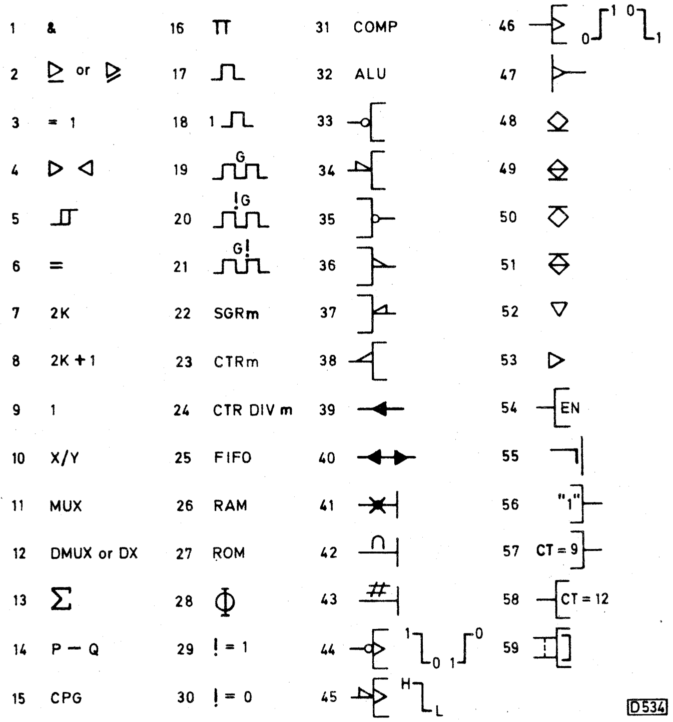


Fig. 1: Signs (qualifying symbols) used with the new logic circuit symbols. 1 And gate or function. 2 Or gate or function. 3 Exclusive-or gate. 4 Buffer or driver amplifier – arrow indicates direction of signal flow. 5 Schmitt trigger. 6 Logic identity – all inputs must stand at the same state. 7 Even number of signal inputs must be active. 8 Odd number of signal inputs must be active. 9 One input must be active. 10 Code Converter. 11 Multiplexer. 12 Demultiplexer. 13 Adder. 14 Subtractor. 15 Carry pulse generator. 16 Multiplier. 17 Retriggerable monostable. 18 One-shot (non-retriggerable monostable). 19 Astable multivibrator (waveform not always shown). 20 Synchronously starting astable multivibrator. 21 Astable – halts when pulse ends. 22 Shift register – m is number of stages. 23 Counter – m is number of stages. 24 Counter with cycle length. 25 First in/first out memory. 26 Random access read/write memory. 27 Read only memory. 28 Gray box. 29 Element powers up and sets to one. 30 Powers up and sets to zero. 31 Comparator. 32 Arithmetic logic unit. 33 Input inverter in chip. 34 Logic one low input (inverter in chip). 35 Output inverted. 36 Logic one low output. 37 Logic one low input with signal movement right to left. 38 Logic one low output with signal movement from right to left. 39 Signal flow right to left (no symbol means flow is left to right). 40 Signal flow in both directions. 41 Non-logic connection (usually labelled). 42 Analogue signal(s) entering digital circuit. 43 Input for digital signal on an analogue symbol. 44 Binary one produces temporary zero output, positive logic. Inverse with negative logic. 45 as 44, negative logic. 46 Binary zero produces temporary one output, positive logic, inverse with negative logic. 47 Zero in produces temporary one out at right. 48 NPN open-collector output. 49 NPN collector output with internal resistor. 50 NPN open-emitter output. 51 NPN emitter output with internal resistor. 52 Tristate output. 53 Buffer amplifier. 54 Enable input. Binary one enables all outputs. 55 No output until applied input logic signal ceases. 56 Fixed state output remains at one. 57 Output active only when register contains indicated number. 58 When input is active internal register will be as number indicated. 59 Input line grouping – two or more signals are required to implement a single logic input.

Dependency Notation

Dependency notation, or in plain English representing logic functions by means of little signs and figures, is the thing that makes the new logic symbols completely different from the old system. At first sight these dependency notations may make the long-suffering TV/video engineer turn quite pale, but their use shows the relationship between a device's inputs and outputs

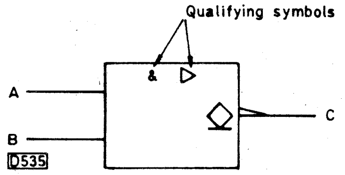
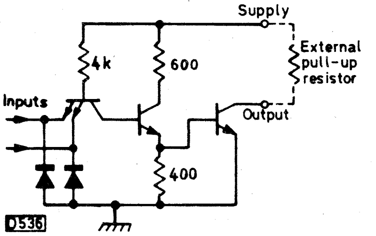


Fig. 2: Nand gate symbol with negative logic output.



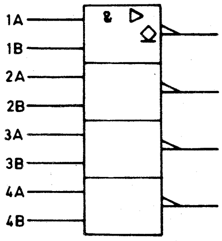


Fig. 3 (left): Internal circuit of the nand gate shown symbolically in Fig. 2.

Fig. 4 (right): Quad nand gate symbol (negative logic output).

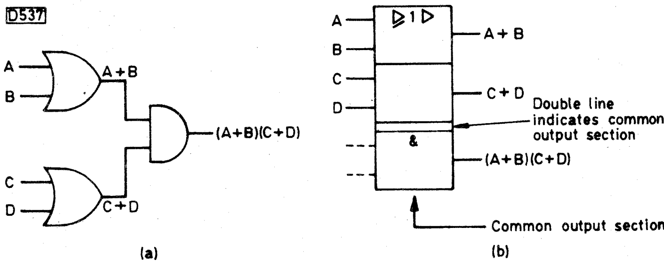


Fig. 5: Old and new symbols, two or gates feeding an and output gate.

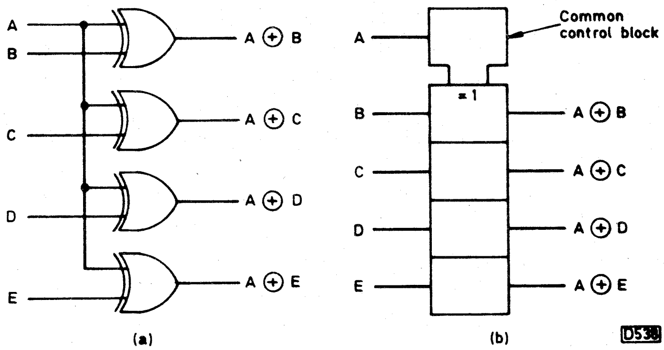


Fig. 6: Old and new symbols, showing a common control block.

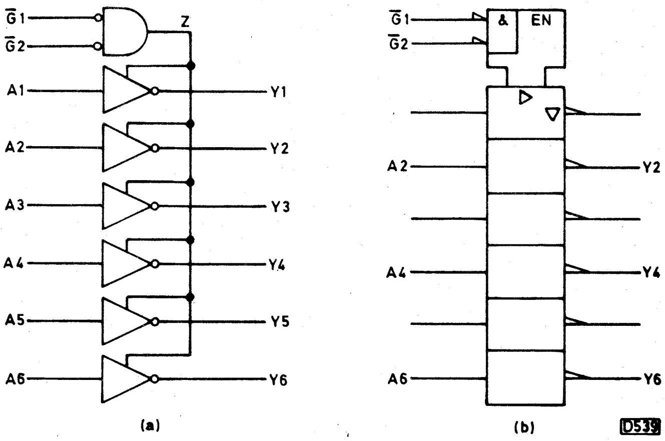


Fig. 7: Further example of old and new symbols.

without the need to show all the separate gates and their interconnections.

There are two simple rules to understanding dependency notation. First, the input or output that affects other inputs or outputs is labelled with a recognised letter symbol – see Table 1 – that shows the logic function, also an identification number. Secondly, each input or output that's affected by the labelled input or output is marked with the same identification number.

We'll consider and dependency first, signified by the letter G.

Fig. 8(a) shows part of an SN74LS257B selector/multiplexer chip drawn using conventional symbols. A multiplexer circuit is used to select one from a number of inputs and route it to a single output. For and gate 1 to produce a binary one signal at point L there must be a binary one at input 1A and a binary zero at input $\overline{A}$B. For and gate 2 to produce a binary one at M there must be a binary one at 1B and at $\overline{A}$B. Outputs L and M are fed to an or gate which is of a type that has to be enabled by a binary one at Z (zero at input $\overline{Z}$). So provided a binary one is present at points Z and either L or M a binary one will be present at output 1Y.

Let's now look at this in the new system, see Fig. 8(b). Input signal $\overline{A}$B is shown connected to a control block input labelled G1. This indicates that it's anded with inputs 1A and 1B to the block below, labelled $\overline{1}$ and 1 respectively. The letters MUX indicate that the block is a multiplexer, the two triangle symbols indicating that the block contains a buffer and a tristate device (the enabled or gate). The blocks below, shown in dashed outline, contain identical circuitry to the one above and all are subject to the common control block at the top. For simplicity the inputs and outputs of the lower block have been omitted.

A point to note in all this is that while the conventional way of drawing a circuit tells us exactly how the circuit works the new system tells us what a device does without showing the circuitry within.

Fig. 9(a) shows an or gate whose output feeds one input of an and gate. The Boolean equations for the conditions at points Y and Z are A + B and (A + B)C respectively. The same circuit can, as part of a complex symbol, be represented as shown in Fig. 9(b). Two inputs labelled with the same letter indicate the inputs to a gate: since the letter used is G this indicates that the gate's output is fed to an and gate whose other input C is also labelled 1.

Or dependency is signified by the letter V. Fig. 10(a) shows an or gate with a signal from point A fed to one input while a signal from elsewhere in the device is fed to its other input. Using dependency notation we can draw the circuit as shown in Fig. 10(b).

The rule with negate (exclusive-or) dependency, letter N, is that every input or output bearing the same number stands in an exclusive-or relationship with the relevant N numbered input or output. This is illustrated in Fig. 11 where the external input to the exclusive-or gate is labelled N1 and the output 1.

The Z dependency symbol shows that the logic device contains internal logic connections between inputs and

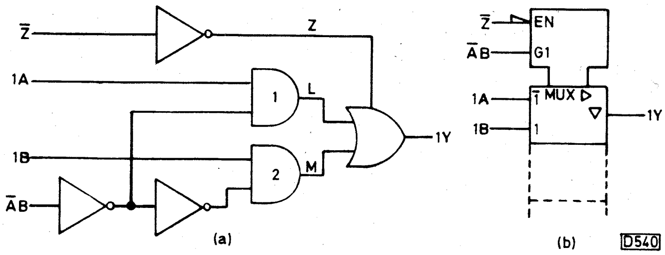
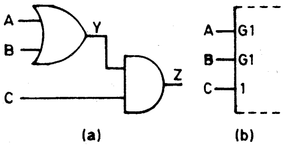


Fig 8: Comparison of old and new symbols with the latter illustrating dependency notation.



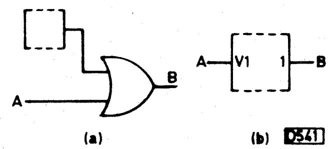
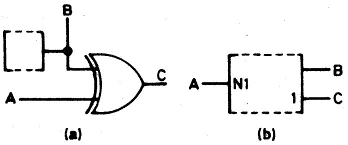


Fig. 9 (left): And gate dependency.

Fig. 10 (right): Or gate dependency.



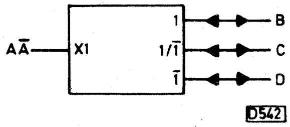


Fig. 11 (left): Negate (exclusive-or) dependency.

Fig. 12 (right): Example of transmission (X) dependency.

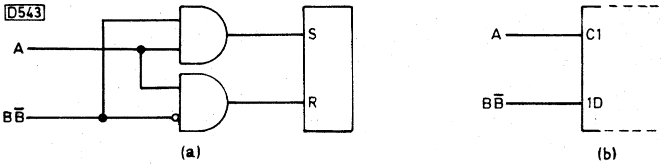
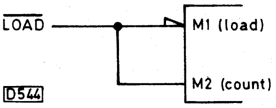


Fig. 13: Control dependency, (a) old and (b) new system.



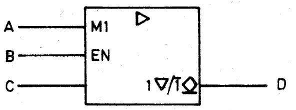
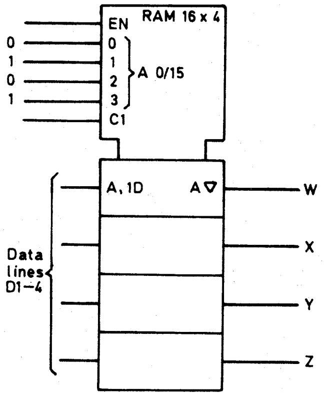


Fig. 14 (left): Input mode dependency.

Fig. 15 (right): Output mode dependency.



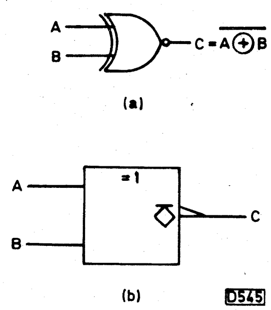


Fig. 16 (left): Address dependency with a 16 × 4 RAM.

Fig. 17 (right): Exclusive-or gate symbols.

Table 1: Dependency notation letter symbols.

Letter symbol	*Function*
G	And
V	Or
N	Negate (exclusive-or)
C	Control
EN	Enable
M	Mode
A	Address
R	Reset
S	Set
X	Transmit
Z	Interconnect

outputs bearing the same number, also internal inputs and outputs. A number such as Z1 is used with all the affected inputs and outputs labelled with the same number 1.

X indicates controlled two-way connections between various input and output ports. In Fig. 12, if $A\overline{A}$ is at binary one B is connected internally to C while if $A\overline{A}$ is at binary zero C is connected to D.

The control dependency denoted by C usually indicates enabling or disabling of the inputs to digital storage circuitry such as bistables. Fig. 13(a) shows a simple example. The circuit can work only when control signal A is at binary one. With inputs A and $B\overline{B}$ at one there will be a one at S (set) and a zero at R (reset). The circuit now stores a binary one. With $B\overline{B}$ at binary zero R will be at one and S at zero. This time the circuit stores zero. The symbol C1 in Fig. 13(b) indicates a common control point, with all inputs controlled by C standing at binary one when A is at one. When A stands at zero all inputs controlled by C1 will be at zero. With A at binary one $B\overline{B}$ must also be at one for the storage device to hold the binary one signal: this is indicated by the symbol 1D.

S, R and EN dependency should be easy enough to follow.

M dependency shows that the various inputs and outputs of a device depend on the mode in which it's operating. For example Fig. 14 shows one small section (an input) of a complex digital chip, the SN54LS690, a synchronous counter with output registers and multiplexed tristate counters. When the input shown is low the signal is inverted, appearing as a high at the point labelled M1. In this mode digital information is loaded into the device. When the input goes high M2 is high and M1 low. The chip now operates as a synchronous counter.

Fig. 15 shows how mode dependency can affect outputs. The symbol at the top of the rectangle indicates a buffer amplifier. With a binary one at input A the device has a tristate output. When A is zero the output becomes an nnp open-collector and the tristate symbol doesn't apply. For the buffer amplifier to work at all an enabling signal must be present at input B.

One use of the address dependency symbol A is shown in Fig. 16, which represents a 16 × 4 RAM chip. The common control block shows the four address lines bracketed together and labelled with an A. 0/15 represents the fact that there are 16 memory storage areas in the chip, each one of which contains a four-bit binary word. Inputs 0 to 3 receive four binary signal values. In the example shown input 0 (the least significant bit) is at zero, input 1 at one, input 2 at zero and input 3 (the most significant bit) at one. The memory location address selected is thus 0101. Any of sixteen locations (0/15) can be selected by using different combinations of binary digits on the address lines.

To write into a memory location input, line C1 (control 1) must be at binary one. The information appearing on the data input lines D1-4 will then be stored in the memory location selected. To read the contents of this location a binary one is applied to the EN input line and zero to C1. When this is done the binary values stored at the selected location appear on output lines W, X, Y and Z.

Gray Box Symbol

When the "gray box" sign shown in Fig. 1-28 appears at the top of a logic device symbol this indicates that some information about the chip is written in words in addition to coding and dependency notation. This is done only in the case of extremely complex logic circuitry. You'll agree that it's an improvement on the blank box outlines that often appear in some skimpy TV/VCR service manuals!

The Exclusive-nor Gate

Fig. 17 shows symbols for the exclusive-nor gate. It operates in the same manner as an exclusive-or gate but with inverted (complemented) output. The Boolean equation is

$$C = \overline{\overline{A}B + A\overline{B}}.$$

We can simplify this by using the rules of Boolean algebra given last month. First by DeMorgan's theorem we can change it to

$$(\overline{\overline{A}B})\ (\overline{A\overline{B}})$$

then by the law of distribution to

$$A\overline{A} + AB + \overline{AB} + B\overline{B}.$$

By the law of complements

$$A\overline{A} = 0 \text{ and } B\overline{B} = 0$$

which leaves

$$AB + \overline{AB}$$

or, by the law of commutation,

$$\overline{AB} + AB$$

which is usually shown as

$$C = \overline{A \oplus B}.$$

Last Month's Problems

Now for the answers to the problems presented last month. When the four inputs to the gates that were shown in Fig. 20 are changed to A, B, C and D

$$E = (A + \overline{B})\ (\overline{C} + \overline{D}).$$

With the four inputs A, $\overline{B}$, C and $\overline{D}$

$$E = (\overline{A} + B)\ (\overline{C} + D).$$

Incidentally, as you may by now have realised given only a Boolean equation it's possible to draw the relevant circuit diagram. For example, if Z = (A + B) (C + D) (E + F) it's obvious that the circuit consists of three or gates which feed their outputs into a three-input and gate.

If you want to take this subject further I'd recommend that you obtain a copy of the latest edition of the Texas Instruments' publication *The TTL Data Book Volume 1.* It contains over a thousand pages of information and shows a good range of the new logic symbols together with their conventional logic diagram counterparts, some of which are extremely complex. The book gives the Boolean expressions for a number of circuits. It's obtainable from RS Components (order from Electromail, PO Box 33, Corby, Northants NN17 9EL) or Texas Intruments Ltd., Manton Lane, Bedford MK41 7PA. The price is £14.95 plus £2.50 post and packing.

The aim of these two articles has been to provide readers with a reasonable knowledge of Boolean algebra and to serve as an introduction to the understanding of the new logic symbols. You'll find that a study of these important subjects will pay off in the months to come.