4.7 C (Control) Dependency

The symbol denoting control dependency is the letter C.

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

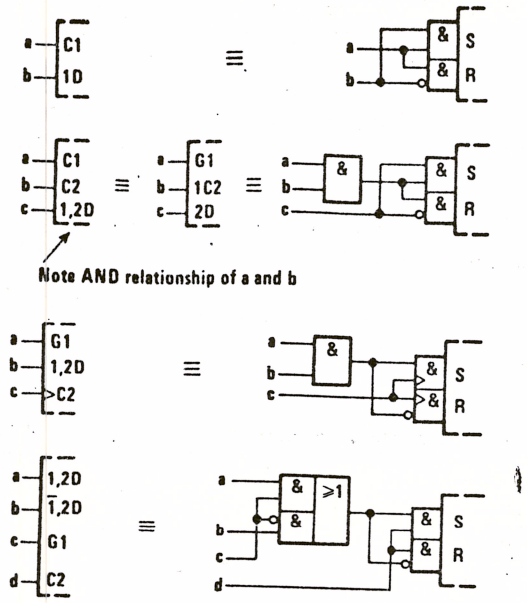


FIGURE 12 – C (CONTROL) DEPENDENCY

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

4.8 S (Set) and R (Reset) Dependencies

The symbol denoting set dependency is the letter S. The symbol denoting reset dependency is the letter R.

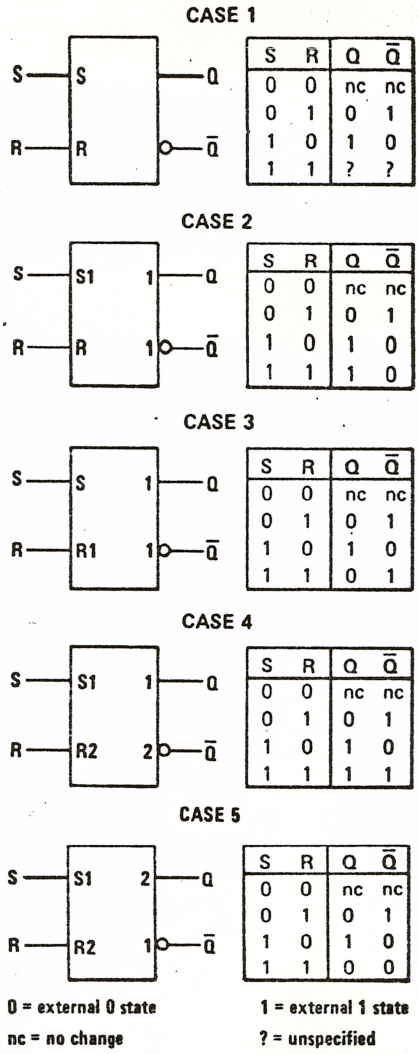
Set and reset dependencies are used if it is necessary to specify the effect of the combination R=S=1 on a bistable element. Case 1 in Figure 13 does not use S or R dependency.

When an S*m* input is at its internal 1 state, outputs affected by the Sm input will react, regardless of the state of an R input, as they normally would react to the combination S=1, R=0. See cases 2, 4, and 5 in Figure 13.

When an R*m* input is at its internal 1 state, outputs affected by the Rm input will react, regardless of the state of an S input, as they normally would react to the combination S=0, R=1. See cases 3, 4, and 5 in Figure 13.

When an S*m* or R*m* input is at its internal 0 state, it has no effect.

Note that the noncomplementary output patterns in cases 4 and 5 are only pseudo stable. The simultaneous return of the inputs to S=R=0 produces an unforeseeable stable and complementary output pattern.



S	R	Q	$\overline{Q}$
0	0	nc	nc
0	1	0	1
1	0	1	0
1	1	?	?

S	R	Q	$\overline{Q}$
0	0	nc	nc
0	1	0	1
1	0	1	0
1	1	1	0

S	R	Q	$\overline{Q}$
0	0	nc	nc
0	1	0	1
1	0	1	0
1	1	0	1

S	R	Q	$\overline{Q}$
0	0	nc	nc
0	1	0	1
1	0	1	0
1	1	1	1

S	R	Q	$\overline{Q}$
0	0	nc	nc
0	1	0	1
1	0	1	0
1	1	0	0

0 = external 0 state **1 = external 1 state**
nc = no change **? = unspecified**

FIGURE 13 – S (SET AND R (RESET) DEPENDENCIES

4.9 EN (Enable) Dependency

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

An EN*m* input has the same effect on outputs as an EN input, see 3.1, but it effects only those outputs labeled with the identifying number *m*. It also affects those inputs labeled with the identifying number *m*. By contrast, an EN input affects all outputs and no inputs. The effect of an EN*m* input on an affected input is identical to that of a C*m* input. See Figure 14.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

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

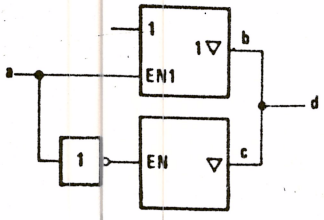


FIGURE 14 – EN (ENABLE) DEPENDENCY

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

4.10 M (Mode) Dependency

The symbol denoting mode dependency is the letter M.

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

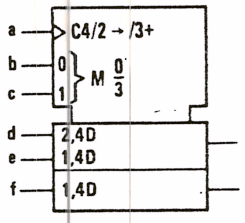
If an input or output has the same effect in different modes of operation, the identifying numbers of the relevant affecting M*m* inputs will appear in the label of that affected input or output between parentheses and separated by solidi. See Figure 19.

4.10.1 M Dependency Affecting Inputs

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

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

The circuit in Figure 15 has two inputs, **b** and **c**, that control which one of four modes (0, 1, 2, or 3) will exist at any time. Inputs **d**, **e**, and **f** are D inputs subject to dynamic control (clocking) by the **a** input. The numbers 1 and 2 are in the series chosen to indicate the modes so inputs **e** and **f** are only enabled in mode 1 (for parallel loading) and input **d** is only enabled in mode 2 (for serial loading). Note that input **a** has three functions. It is the clock for entering data. In mode 2, it causes right shifting of data, which means a shift away from the control block. In mode 3, it causes the contents of the register to be incremented by one count.



Note that all operations are synchronous.

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

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

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

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

FIGURE 15 – M (MODE) DEPENDENCY AFFECTING INPUTS

4.10.2 M Dependency Affecting Outputs

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

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

In Figure 16, mode 1 exists when the **a** input stands at its internal 1 state. The delayed 1 output symbol is effective only in mode 1 (when input **a** = 1) in which case the device functions as a pulse-triggered flip-flop. See Section 5. When input **a** = 0, the device is not in mode 1 so the delayed output symbol has no effect and the device functions as a transparent latch.

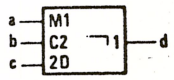


FIGURE 16 – TYPE OF FLIP-FLOP DETERMINED BY MODE

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

In Figure 17, if input a stands at its internal 1 state establishing mode 1, output b will stand at its internal 1 state only when the content of the register equals 9. Since output b is located in the common-control block with no defined function outside of mode 1, this output will stand at its internal 0 state when input a stands at its internal 0 state, regardless of the register content.

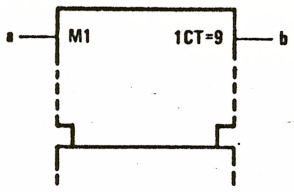


FIGURE 17 – DISABLING AN OUTPUT OF THE COMMON-CONTROL BLOCK

In Figure 18, if input a stands at its internal 1 state establishing mode 1, output b will stand at its internal 1 state only when the content of the register equals 15. If input a stands at its internal 0 state, output b will stand at its internal 1 state only when the content of the register equals 0.

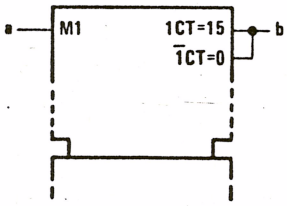


FIGURE 18 – DETERMINING AN OUTPUT'S FUNCTION

In Figure 19 inputs a and b are binary weighted to generate the numbers 0, 1, 2, or 3. This determines which one of the four modes exists.

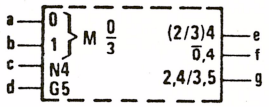


FIGURE 19 – DEPENDENT RELATIONSHIPS AFFECTED BY MODE

At output e the label set causing negation (if c = 1) is effective only in modes 2 and 3. In modes 0 and 1 this output stands at its normally defined state as if it had no labels. At output f the label set has effect when the mode is not 0 so output e is negated (if c = 1) in modes 1, 2, and 3. In mode 0 the label set has no effect so the output stands at its normally defined state. In this example 0,4 is equivalent to (1/2/3)4. At output g there are two label sets. The first set, causing negation (if c = 1), is effective only in mode 2. The second set, subjecting g to AND dependency on d, has effect only in mode 3.

Note that in mode 0 none of the dependency relationships has any effect on the outputs, so e, f, and g will all stand at the same state.

4.11 A (Address) Dependency

The symbol denoting address dependency is the letter A.

Address dependency provides a clear representation of those elements, particularly memories, that use address control inputs to select specified sections of a multidimensional array. Such a section of a memory array is usually called a word. The purpose of address dependency is to allow a symbolic presentation of the entire array. An input of the array shown at a particular element of this general section is common to the corresponding elements of all selected sections of the array. An output of the array shown at a particular element of this general section is the result of the OR function of the outputs of the corresponding elements of selected sections. If the label of an output of the array shown at a particular element of this general section indicates that this output is an open-circuit output or a three-state output, then this indication refers to the output of the array and not to those of the sections of the array.

Inputs that are not affected by any affecting address input have their normally defined effect on all sections of the array, whereas inputs affected by an address input have their normally defined effect only on the section selected by that address input.

An affecting address input is labelled with the letter A followed by an identifying number that corresponds with the address of the particular section of the array selected by this input. Within the general section presented by the symbol, inputs and outputs affected by an A*m* input are labelled with the letter A, which stands for the identifying numbers, i.e., the addresses, of the particular sections.

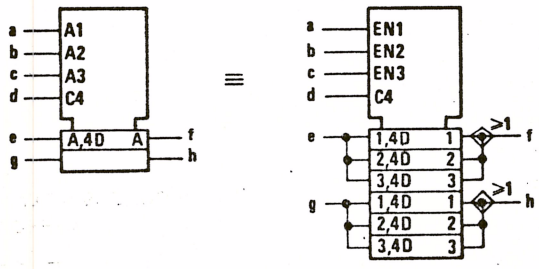


FIGURE 20 – A (ADDRESS) DEPENDENCY

Figure 20 shows a 3-word by 2-bit memory having a separate address line for each word and uses EN dependency to explain the operation. To select word 1, input a is taken to its 1 state, which establishes mode 1. Data can now be clocked into the inputs marked "1,4D". Unless words 2 and 3 are also selected, data cannot be clocked in at the inputs marked "2,4D" and "3,4D". The outputs will be the OR functions of the selected outputs, i.e., only those enabled by the active EN functions.

The identifying numbers of affecting address inputs correspond with the addresses of the sections selected by these inputs. They need not necessarily differ from those of other affecting dependency-inputs (e.g., G, V, N, . . .), because in the general section presented by the symbol they are replaced by the letter A.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

If there are several sets of affecting *Am* inputs for the purpose of independent and possibly simultaneous access to sections of the array, then the letter A is modified to 1A, 2A, . . . Because they have access to the same sections of the array, these sets of A inputs may have the same identifying numbers.

Figure 21 is another illustration of the concept.

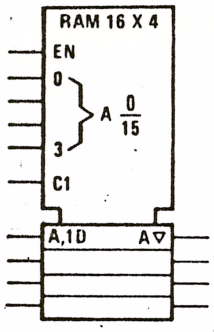


FIGURE 21

FIGURE 21 – ARRAY OF 16 SECTIONS OF FOUR TRANSPARENT LATCHES WITH 3-STATE OUTPUTS COMPRISING A 16-WORD X 4-BIT RANDOM-ACCESS MEMORY

TABLE IV – SUMMARY OF DEPENDENCY NOTATION

TYPE OF DEPENDENCY	LETTER SYMBOL*	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	Prevents action of inputs. ◇outputs off. ▽outputs at external high impedance, no change in internal logic state. Other outputs at internal 0 state.
AND	G	Permits action	Imposes 0 state
Mode	M	Permits action (mode selected)	Prevents action (mode not selected)
Negate (X-OR)	N	Complements 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	Permits action
Interconnection	Z	Imposes 1 state	Imposes 0 state

*These letter symbols appear at the AFFECTING input (or output) and are followed by a number. Each input (or output) AFFECTED by that input is labeled with that same number. When the labels EN, R, and S appear at inputs without the following numbers, the descriptions above do not apply. The action of these inputs is described under "Symbols Inside The Outline", see 3.1.

5 BISTABLE ELEMENTS

The dynamic input symbol, the postponed output symbol, and dependency notation provide the tools to differentiate four main types of bistable elements and make synchronous and asynchronous inputs easily recognizable. See Figure 22. The first column shows the essential distinguishing features; the other columns show examples.

Transparent latches have a level-operated control input. The D input is active as long as the C input is at its internal 1 state. The outputs respond immediately. Edge-triggered elements accept data from D, J, K, R, or S inputs on the active transition of C. Pulse-triggered elements require the setup of data before the start of the control pulse; the C input is considered static since the data must be maintained as long as C is at its 1 state. The output is postponed until C returns to its 0 state. The data-lock-out element is similar to the pulse-triggered version except that the C input is considered dynamic in that shortly after C goes through its active transition, the data inputs are disabled and data does not have to be held. However, the output is still postponed until the C input returns to its initial external level.

Notice that synchronous inputs can be readily recognized by their dependency labels (1D, 1J, 1K, 1S, 1R) compared to the asynchronous inputs (S, R), which are not dependent on the C inputs.

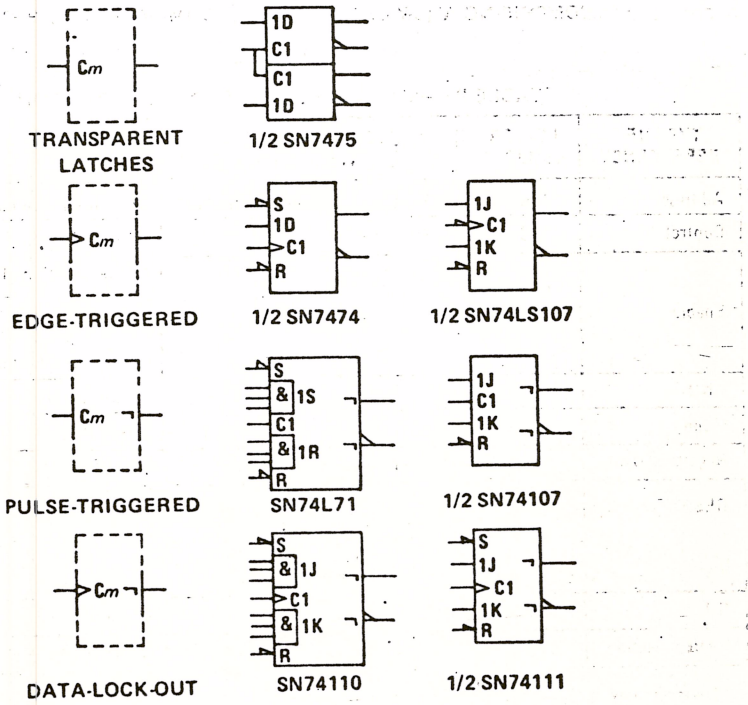


FIGURE 22 – FOUR TYPES OF BISTABLE CIRCUITS

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

6 CODERS

The general symbol for a coder or code converter is shown in Figure 23. X and Y may be replaced by appropriate indications of the code used to represent the information at the inputs and at the outputs, respectively.

FIGURE 23 – CODER GENERAL SYMBOL

Indication of code conversion is based on the following rule:

Depending on the input code, the internal logic states of the inputs determine an internal value. This value is reproduced by the internal logic states of the outputs, depending on the output code.

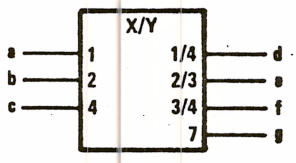
The indication of the relationships between the internal logic states of the inputs and the internal value is accomplished by:

1) labelling the inputs with numbers. In this case the internal value equals the sum of the weights associated with those inputs that stand at their internal 1-state, or by
2) replacing X by an appropriate indication of the input code and labelling the inputs with characters that refer to this code.

The relationships between the internal value and the internal logic states of the outputs are indicated by:

1) labelling each output with a list of numbers representing those internal values that lead to the internal 1-state of that output. These numbers shall be separated by solidi as in Figure 24. This labelling may also be applied when Y is replaced by a letter denoting a type of dependency (see Section 7). If a continuous range of internal values produces the internal 1 state of an output, this can be indicated by two numbers that are inclusively the beginning and the end of the range, with these two numbers separated by three dots, e.g., 4 . . . 9 = 4/5/6/7/8/9, or by
2) replacing Y by an appropriate indication of the output code and labelling the outputs with characters that refer to this code as in Figure 25.

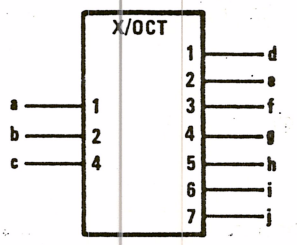
Alternatively, the general symbol may be used together with an appropriate reference to a table in which the relationship between the inputs and outputs is indicated. This is a recommended way to symbolize a PROM after it has been programmed.



TRUTH TABLE

INPUTS			OUTPUTS			
c	b	a	g	f	e	d
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	1	0
0	1	1	0	1	1	0
1	0	0	0	1	0	1
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	1	0	0	0

FIGURE 24 – AN X/Y CODE CONVERTER



TRUTH TABLE

INPUTS			OUTPUTS						
c	b	a	j	i	h	g	f	e	d
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	0
0	1	1	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	1	0	0	1	0	0	0	0
1	1	0	0	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0

FIGURE 25 – AN X/OCTAL CODE CONVERTER

7 USE OF A CODER TO PRODUCE AFFECTING INPUTS

It often occurs that a set of affecting inputs for dependency notation is produced by decoding the signals on certain inputs to an element. In such a case use can be made of the symbol for a coder as an embedded symbol. See Figure 26.

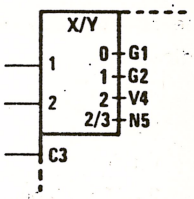


FIGURE 26 – PRODUCING VARIOUS TYPES OF DEPENDENCIES

If all affecting inputs produced by a coder are of the same type and their identifying numbers correspond with the numbers shown at the outputs of the coder, Y (in the qualifying symbol X/Y) may be replaced by the letter denoting the type of dependency. The indications of the affecting inputs should then be omitted. See Figure 27.

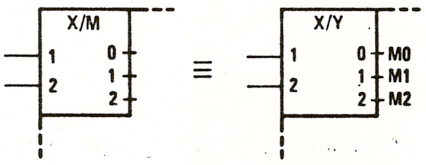


FIGURE 27 – PRODUCING ONE TYPE OF DEPENDENCY

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

8 USE OF BINARY GROUPING TO PRODUCE AFFECTING INPUTS

If all affecting inputs produced by a coder are of the same type and have consecutive identifying numbers not necessarily corresponding with the numbers that would have been shown at the outputs of the coder, use can be made of the binary grouping symbol (see 3.1). k external lines effectively generate 2^k internal inputs. The bracket is followed by the letter denoting the type of dependency followed by $\frac{m1}{m2}$. The m1 is to be replaced by the smallest identifying number and the m2 by the largest one, as shown in Figure 28.

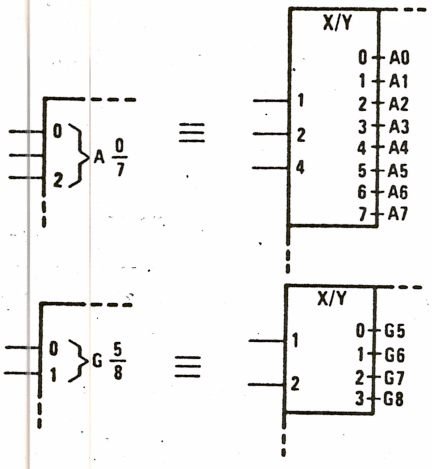


FIGURE 28 – USE OF THE BINARY GROUPING SYMBOL

9 SEQUENCE OF INPUT LABELS

If an input having a single functional effect is affected by other inputs, the qualifying symbol (if there is any) for that functional effect is preceded by the labels corresponding to the affecting inputs. The left-to-right order of these preceding labels is the order in which the effects or modifications must be applied. The affected input has no functional effect on the element if the logic state of any one of the affecting inputs, considered separately, would cause the affected input to have no effect, regardless of the logic states of other affecting inputs.

If an input has several different functional effects or has several different sets of affecting inputs, depending on the mode of action, the input may be shown as often as required. However, there are cases in which this method of presentation is not advantageous. In those cases the input may be shown once with the different sets of labels separated by solidi. See Figure 29. No meaning is attached to the order of these sets of labels. If one of the functional effects of an input is that of an unlabelled input of the element, a solidus will precede the first set of labels shown.

If all inputs of a combinational element are disabled (caused to have no effect on the function of the element), the internal logic states of the outputs of the element are not specified by the symbol. If all inputs of a sequential element are disabled, the content of this element is not changed and the outputs remain at their existing internal logic states.

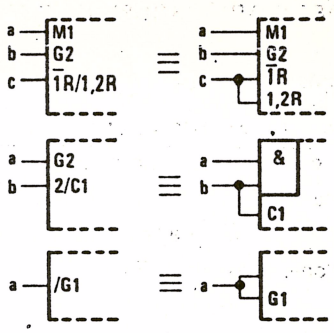


FIGURE 29 – INPUT LABELS

Labels may be factored using algebraic techniques.

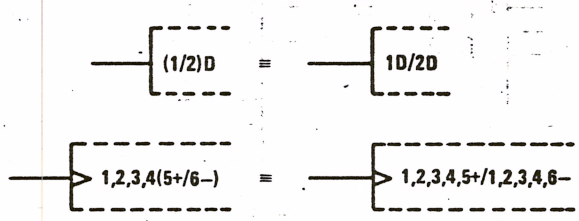


FIGURE 30 – FACTORING INPUT LABELS

10 SEQUENCE OF OUTPUT LABELS

If an output has a number of different labels, regardless of whether they are identifying number of affecting inputs or outputs or not, these labels are shown in the following order:

1) if the postponed output symbol has to be shown, this comes first, if necessary preceded by the indications of the inputs to which it must be applied;

2) followed by the labels indicating modifications of the internal logic state of the output, such that the left-to-right order of these labels corresponds with the order in which their effects must be applied;

3) followed by the label indicating the effect of the output on inputs and other outputs of the element.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

Symbols for open-circuit or three-state outputs, where applicable, are placed just inside the outside boundary of the symbol adjacent to the output line. See Figure 31.

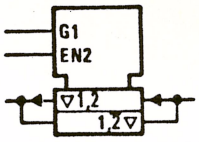


FIGURE 31 – PLACEMENT OF 3-STATE SYMBOLS

If an output needs several different sets of labels that represent alternative functions (e.g., depending on the mode of action), these sets may be shown on different output lines that must be connected outside the outline. However, there are cases in which this method of presentation is not advantageous. In those cases the output may be shown once with the different sets of labels separated by solidi. See Figure 32.

Two adjacent identifying numbers of affecting inputs in a set of labels that are not already separated by a nonnumeric character should be separated by a comma.

If a set of labels of an output not containing a solidus contains the identifying number of an affecting M*m* input standing at its internal 0 state, this set of labels has no effect on that output.

Labels may be factored using algebraic techniques.

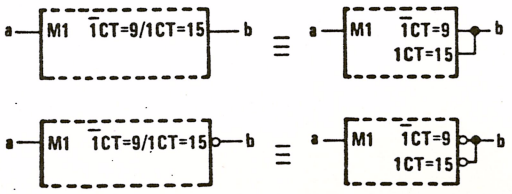


FIGURE 32 – OUTPUT LABELS

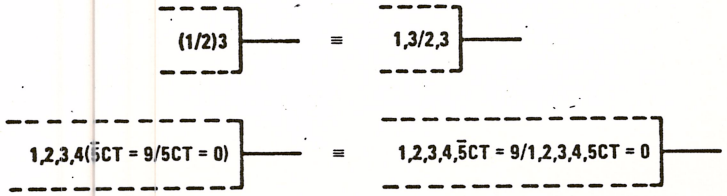


FIGURE 33 – FACTORING OUTPUT LABELS

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

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

IEEE Standards may be purchased from:

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

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

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