Coders, code converters

1 Relationships between inputs and outputs of coders

1.1 Indication of input and output codes in the general qualifying symbol

This method of indicating code conversion is based on the following rule:

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

The relationships between the internal logic states of the inputs and the internal value shall be indicated in one of the following ways:

– label the inputs with numbers, in which case the internal value equals the sum of the numbers associated with those inputs that stand at their internal 1-states; or

– replace X by an appropriate designation of the input code and label the inputs with characters that refer to this code.

The relationships between the internal value and the internal logic states of the outputs shall be indicated in one of the following ways:

– label each output with a list of numbers representing the internal values that lead to the internal 1-state of that output. These numbers shall be separated by solidi. This method may also be applied when Y is replaced by a letter denoting a type of dependency (see also A00288). 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, for example, 4 ... 9 = 4/5/6/7/8/9; or

– replace Y by an appropriate indication of the output code and label the outputs with characters that refer to this code.

For illustrations, see A00296_Illustration_a_EN.pdf below.

NOTE - Alternatively, the general qualifying symbol BIN/6 may be used instead of X/Y. See 1.1.1 and 1.1 2.

If X or Y is replaced by an indication of a specific code, further rules apply.

In the following text, the codes are subdivided into three categories:

– summing codes,

– direct-indication codes, and

– identification codes.

1 Summing Codes

With these codes, like "X", there is an internal numeric value that corresponds to the sum of the weights of the inputs [outputs] that stand at their internal 1-states.

The indication of the relationships between the internal logic states of the inputs [outputs] and the internal value shall be accomplished by replacing X [Y] of the qualifying symbol with an appropriate indication of the input [output] code and by labelling the inputs [outputs] with numbers indicating their individual weights.

The following summing codes are defined:

BIN Binary code

The number code in which the individual weights are all powers of 2. Inputs [outputs] shall be labelled either with decimal weights or with decimal exponents of the powers of 2.

BCD 8-4-2-1 Binary-coded decimal

The number code in which each digit in the decimal representation of a number is encoded as a binary number in 4 bits with the relative weights of 8, 4, 2, and 1.

For an example, see A00296_Example_a_EN.pdf below.

Inputs [outputs] shall be labelled with decimal weights, for example 1, 2, 4, 8, 10, 20, etc.

NOTE - For inputs, the behaviour of the element is unspecified by the symbol if the internal value produced by any set of four inputs exceeds 9 (x10^n). For outputs, the behaviour of the element is unspecified by the symbol if the internal value requires more digits than are provided at the outputs.

X-3 Excess-three code

The BCD code in which the internal value of each 4 inputs [outputs] is 3 (X10^n) less than the sum of those inputs [outputs]. See note to BCD.

For illustrations, see A00296_Illustration_b_EN.pdf below.

For invalid BCD codes, that is, those that would produce an internal value greater than 9, the resulting output states are not specified by this symbol. If the general qualifying symbol were BIN/Y, then the symbol would show that all outputs stand at the internal 0-state for internal values greater than 9.

2CMPL Twos complement code

The n-bit number code (x_{n-1}, ..., x_0) representing a number y in the range $-2^{k+n-1} \leq y \leq 2^{k+n-1} - 2^k$. (For integers, k = 0. For fixed-point fractions, k is negative.)

The individual weights of x_0 through x_{n-2} are powers of 2(2^k through 2^{k+n-2}). The additional bit (x_{n-1},) indicates -2^{k+n-1}. The relationship between the values of the individual bits and y can be expressed by

$$y = -2^{k+n-1} x_{n-1}, + \sum 2^{k+i} x_i$$

A negative [positive] number is represented by 2^k plus the one's-complement (logic complement) of the corresponding positive [negative] number.

For an example, see A00296_Example_b_EN.pdf below.

Inputs [outputs] shall be labelled either with positive decimal weights or with exponents of the powers of 2 including the highest order (sign) bit.

For an illustration, see A00296_Illustration_c.pdf below.

1.1.2 Direct-indication codes

With these codes, like "Y", the relationship between the internal numeric value and the internal logic state of each input [output] shall be indicated by replacing X [Y] of the qualifying symbol with an appropriate indication of the input [output] code and by labelling each input with a number indicating the internal value produced, or by labelling each output with a list of numbers indicating those internal values that lead to the internal 1-state of that output. These numbers shall be separated by solidi.

If a continuous range of internal values produces the internal 1-state of an output, this may be indicated by two numbers that are inclusively the beginning and the end of the range, with these two numbers separated by three dots, for example:

4 ... 9 = 4/5/6/7/8/9.

The following codes are defined:

m **General code with m states (m shall be replaced by a number)**

A code in which m combinations of internai logic states are defined for inputs or possibly for outputs.

HPRI **Highest-priority input code**

An input code in which the input with the highest weight takes priority if more than one input stands at its internal 1-state. If no input stands at its internal 7-state, the internal value is zero.

DEC **Decimal code**

The code in which 10 inputs [outputs] exist and have the weights 0 through 9.

NOTE - If the input [output] with the weight of zero is omitted, the internal value of zero corresponds to all inputs [outputs] standing at their internal 0-states.

OCT **Octal code**

The code in which 8 inputs [outputs] exist and have the weights 0 through 7. See note to DEC.

HEX **Hexadecimal code**

The code in which 16 inputs [outputs] exist and have the weights 0 through 15. See note to DEC.

Except for HPRI, if these codes are used for inputs and more than one input stands at its internal l-state, the behaviour of the element is not specified by the symbol.

For illustrations, see A00296_Illustration_d_EN.pdf below.

1.1.3 Identification codes

With these codes there is no internal numeric value. Instead, each input [output] pattern identifies a symbol (for example the letter "E") or other object according to a named coding scheme. The equivalent of the internal numeric value is the symbol or object identified by the input [output] pattern. Examples of these codes are ISO Latin-1, ASCII, EBCDIC, and 7-segment. The relationship between the internal symbol or object and the internal logic state of each input [output] shall be indicated by replacing X [Y] of the qualifying symbol with an appropriate indication of the input [output] code and by labelling each input [output] with an appropriate indication of its bit position within the code.

If a code identifying a symbol is used in a coder together with a code that is associated with internal numeric values, the conversion to or from these codes is based on the symbolic decimal representation of those internal numeric values. If there is no symbolic representation for a value in the code, the behaviour of the element for that value is unspecified by the symbol for the element.

For an illustration, see A00296_Illustration_e.pdf below.

1.2 Use of coding tables

As an alternative to the use of the previously defined codes and labelling, the general qualifying symbol X/Y (or another, more appropriate, qualifying symbol) may be used together with an appropriate reference to a table (as, for example, in symbol S01621 (12-33-09)) in which the relationship between the inputs and outputs is indicated. The correspondence between inputs [outputs] and the columns in the table may be given in any convenient way, for example by using terminal designations. In this case, any internal labelling that might be confused with that arising in one of the other methods shall be avoided.

For an illustration, see A00296_Illustration_f_EN.pdf below.

2 Replacement of X and Y by indications other than designations of the input code or the output code

2.1 The internal value of a coder may also be produced by other means, for example by a counter whose content is the internal value, by a multi-position switch whose position produces the internal value, etc. In such cases, the X shall be replaced by an appropriate indication of the means involved.

For illustrations, see A00296_Illustration_g_EN.pdf below.

2.2 The internal value of a coder may also be represented by a visual display or be regarded as a value to become the content of an element or as a value on which a mathematical operation is performed. In such cases, the Y shall be replaced by the general qualifying symbol of the function involved.

For an illustration, see A00296_Illustration_h_EN.pdf below.

2.3 It may be necessary, especially if an internal register is involved, to specify both an input code and an output code in addition to the type of register found in between the inputs and outputs, for example, "BCD/CTRDIV1OO/BIN".

For an illustration, see A00296_Illustration_j_EN.pdf below.