



## ALLEN-BRADLEY BULLETIN 1336 IMPACT FUNCTION BLOCK OVERVIEW

APPLICATION NOTE # 1336E - 4

August 27, 1997

---

---

### PURPOSE

The purpose of this document is to provide guidelines for implementing the function block in the Bulletin 1336E AC Drive. This document is to be used as a suggestion only. Users must ensure that installations meet applicable codes and are suitable for the existing conditions.

The Bulletin 1336E User Manual should be used as a reference to ensure that proper wire selection, routing and fusing guidelines are followed.

### WHAT THIS NOTE CONTAINS

This note contains descriptions and possible uses for the function block that are incorporated into the 1336E drive.

### INTENDED AUDIENCE

This application note is intended to be used by personnel familiar with the hardware components and programming procedure necessary to operate the Bulletin 1336E.

### WHERE IT IS USED

The diagrams, parameter settings and auxiliary hardware used in this application note are designed to address specific issues in many different applications. Some changes by the User may be necessary to apply the concepts of this document to a specific application.

### TERMS AND DEFINITIONS

[ ] - indicates a parameter name.

Function Block - Is a group of parameters that dynamically interact to perform a math, logical, counter, or timer task as determined by the user.


Function Input - Is an unfiltered value used by a function block.

Function Mask - Is a filter for the function input values.

Function Evaluation - Is the type of filter used to mask a function input.

Function Output - Is the result of the function block "program" that is used to adjust/control the desired operation for a specific application.

Link - A link is a software connection between two parameters that lets one parameter receive information from another.

 - This represents a **source** which is a link parameter that provides the information.

 - This represents a **destination** which is a link parameter receiving the information.

## DESCRIPTION

Function blocks are used to perform internal logical, mathematical, timer, or counter features. A “block” of parameters (198-214) define the task performance. Inputs are used to manipulate/interact with each other to accomplish a “function” (addition/multiplication/time/count). Since this is a “block” of parameters, only one of the functions can be programmed. Figure 3 shows an overview of all the function block parameters.

To configure the function block parameters, we must understand the purpose of each parameter type . Each parameter is a 16-bit word.

### A. Function In 1,2,3

These three parameters (198, 201, 204) reflect a process or operational state of the machine that need to be acted upon. Parameters can be bit-coded to indicate a status such as, at speed/running/enabled and many more. It may be necessary to only act upon the state of a single bit.

### B. Function Mask/Val 1,2,3

These three parameters (199, 202, 205) are filters for the Function 1,2,3 inputs. The mask is used to select only the desired bits for evaluation.

### C. Function Eval Sel 1,2,3

These three parameters (200, 203, 206) define how the mask will evaluate the value in the “Function Input” word. A typical evaluation is to AND the inputs; only the bits that are true in the mask and the input are set to a one.

Refer to figure 1 for a simplified diagram of the parameters that are used to develop “In1”.

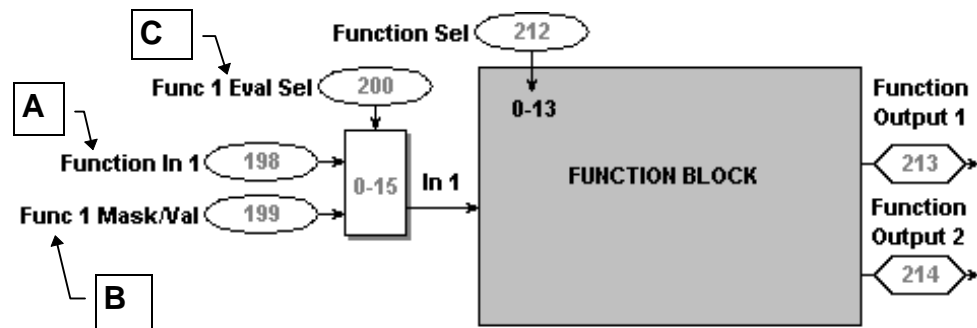


Figure 1

### D. Function In 4-8

These parameters (207-211) are used to set the delay time for a timer, the increment/decrement value for a counter the input range for a scale, or may be used as an output value for a state machine. The mask function does not apply to these parameters.

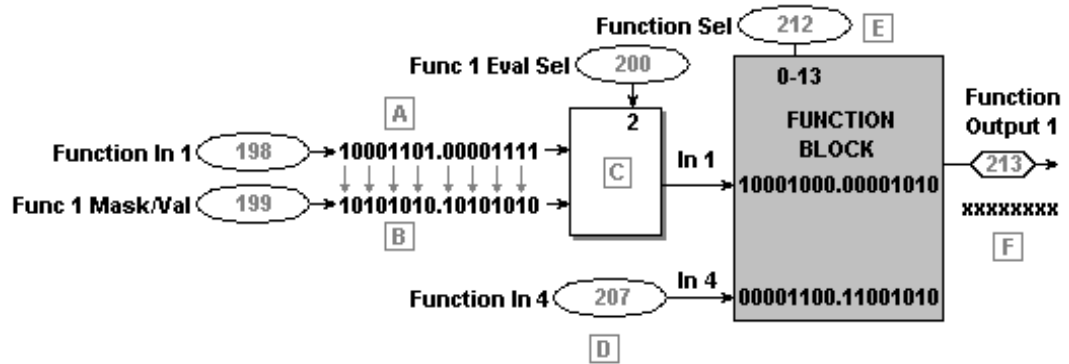
### E. Function Sel

This parameter (212) defines the function performed by the “block” of parameters (timer/counter/math/max-min/state machine/scale).

**F. Function Output 1 & 2**

These parameters (213, 214) are the resultant output of the block. Some of the functions use only output #1, while others will use both.

Since the parameters described in section A, B, C, D are used to perform the same type of operation, we will use the “function #1” and “function #4” input parameters for an example. Refer to figure 2.



**Figure 2**

The example shows how (A)[Function In 1] and (B)[Func 1 Mask/Val] are compared. A value of 2 in (C)[Func 1 Eval Sel] means only the bits that are “set” in the mask will be evaluated. The evaluation is a bitwise AND function (if both are “1”, then output a “1”). The result of the mask is shown as “In 1” inside of the function block.

The value of (D)[Function In 4] is passed directly in, no masking function is used. The operation of the function block is defined by the value (0-13) set in (E)[Function Sel]. The result of the function block operation (F) is shown in [Function Output 1]. This can be either one word (16-bits), or two words (32-bits).

Many permutations can be configured with the function block parameters, but only one function may be programmed. For this reason it is important to understand the basic operation of each type of function block.

**DEFINING THE FUNCTION BLOCK**

**TIMER/DELAY**

The timer/delay function emulates the action of a mechanical on/off delay timer. The function inputs are routed to circuits that perform logical functions (AND,NAND,OR, and NOR). Additional noise immunity can be achieved by using a time value in both the on and off delay circuits. Refer to application note #1336E - 6 for more detailed information.

**STATE MACHINE**

The state machine block is a function generator. The logical true/false states of the function input parameters are used to generate one of four different outputs. Refer to application note #1336E - 7 for more detailed information.

<u>In1 timed</u>	<u>In2</u>	<u>Out1</u>
False	False	In3
True	False	In6
False	True	In7
True	True	In8

**ADD/SUBTRACT**

This block adds the values of two function input parameters. A subtract may be performed if one signal is negative in polarity. Refer to application note #1336E - 8 for more detailed information.

**MAXIMUM/MINIMUM**

The logical true/false state of an input is used as a selector switch. When this input is true, the maximum value of two input signals is routed to the output. When false, the minimum value becomes the output. Refer to application note #1336E - 9 for more detailed information.

**UP/DOWN COUNTER**

This block performs a count up or count down function. It is edge triggered with adjustable gain (increment/decrement). The counter can be reset , and the output can be a single or a double word. Refer to application note #1336E - 10 for more detailed information.

**MULTIPLY/DIVIDE**

This block multiplies two inputs and divides them by a third input. The output, which is a double word, can be standard or per unit math as determined by an input signal. Refer to application note #1336E - 11 for more detailed information.

**SCALE**

The scale block performs offset, scale and clamp functions. Input and output ranges are specified by the function input parameters. The output of this block is a double word. Refer to application note #1336E - 12 for more detailed information.

Figure 3 shows an overview of the available blocks and the parameters associated with each block. Not all parameters are used for every block.

Table 1 shows a listing of the mask evaluations available for function inputs 1-3.

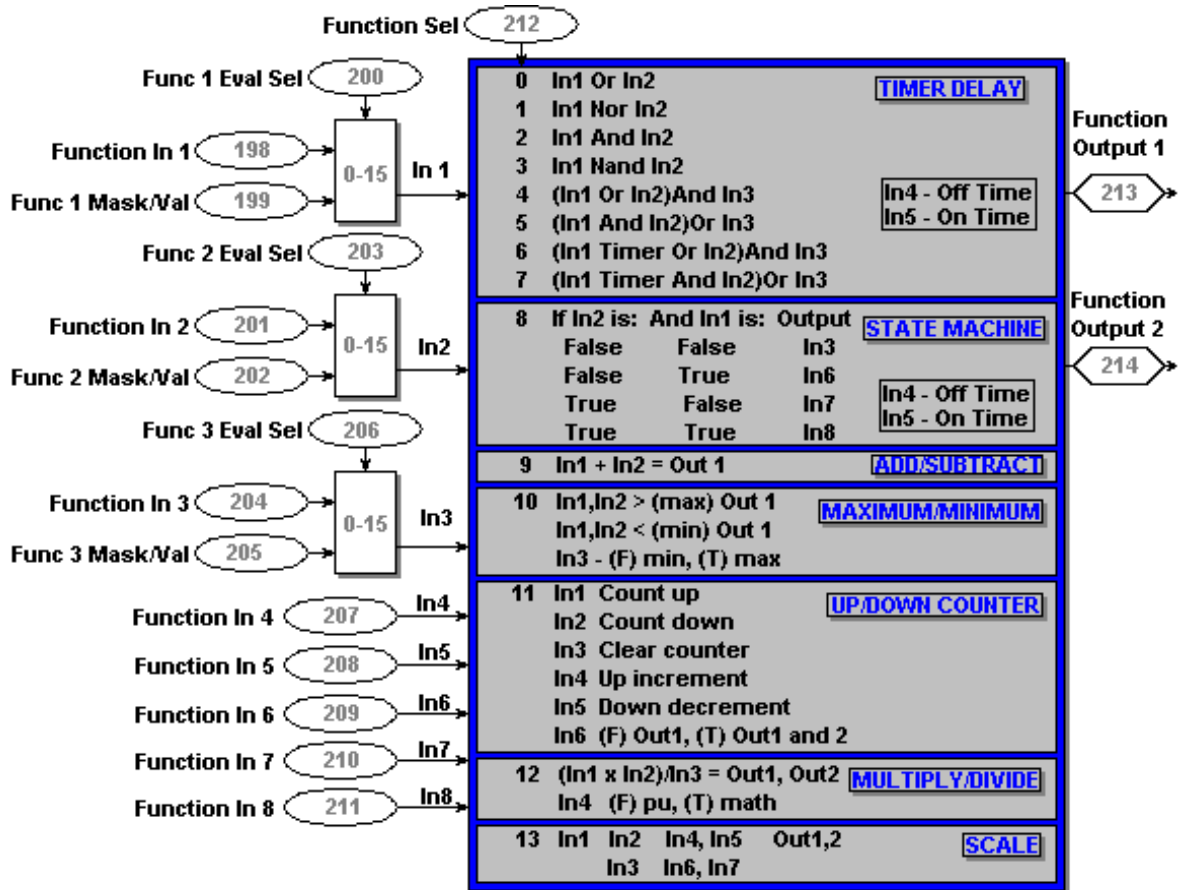


Figure 3

Table 1

VALUE	EVALUATION (1 = on = true = set = closed) (0 = off = false = reset = open)
0	Pass the value directly through the function block
1	Mask the value (logical AND the input value with a value)
2	Send a true value when <b>all</b> bits that are set in the mask are <b>on</b> in the input value
3	Send a true value when <b>all</b> bits that are set in the mask are <b>off</b> in the input value
4	Send a true value when <b>any</b> bits that are set in the mask are <b>on</b> in the input value
5	Send a true value when <b>any</b> bits that are set in the mask are <b>off</b> in the input value
6	Send a true value when the input value is equal to the value of the mask
7	Send a true value when the input value is not equal to the value of the mask
8	Send a true value when the signed input value is < the value of the mask
9	Send a true value when the signed input value is < or = to the value of the mask
10	Send a true value when the signed input value is > the value of the mask
11	Send a true value when the signed input value is > or = to the value of the mask
12	Send a true value when the unsigned input value is < the value of the mask
13	Send a true value when the unsigned input value is < or = to the value of the mask
14	Send a true value when the unsigned input value is > the value of the mask
15	Send a true value when the unsigned input value is > or = to the value of the mask