Auxiliary Function PROM (Cat. No. 1772-AF1) for the Mini–PLC–2/15 Controller

User Manual

Table of Contents

Chapter 1

Introduction

General

Installation of the Auxiliary Function (AF) PROM (cat. no. 1772-AF1) in your Mini-PLC-2/15 controller lets you expand its mathematical capabilities.

For simplification, throughout this manual we refer to the Auxiliary Function PROM (cat. no. 1772-AF1) as the AF1.

The AF1 can only be used with the series A Mini-PLC-2/15 processor module, firmware revision 11 or later (cat. no. 1772-LV). The AF4 can only be used with the series B Mini-PLC-2/15 processor module, firmware revision 4 or later. Programming the AF1 functions with either series Mini-PLC-2/15 processor module requires the Industrial Terminal (cat. no. 1770-T3).

The AF1 has a 2K (16 bit) word section to which you can transfer your program (for backup memory) and a 2K word section for higher mathematical functions. you can only transfer your program into the AF1 with the series B PLC-2/15 controller (Table 1.A). Series A Mini-PLC-2/15 Processor EPROM (publication 1770-915) describes program transfer to PROM. With the series A PLC-2/15 controller, program transfer to the AF1 is not possible.

Table 1.A
AF1 Response to Controller

Mini-PLC-2/15 Controller Series	User Program 2K Words		
	Read	Write	Erase [1]
A	Yes	No	No
B	Yes	Yes	No

[1]You can erase the 2K memory backup portion of the AF1 with ultraviolet light. However, the 2K section for higher mathematical functions would also be erased and all AF1 function capabilities lost. Once erased, the AF1 functions are irretrievable.

NOTE: The AF1 is sensitive to ultraviolet light, therefor when exposed to uv light, both the program and the auxiliary functions are erased. The AF1's transparent window is covered with the product label to avoid accidental alternation of memory from uv light sources. Do not remove this label.

Functions

The AF1 performs the following arithmetic functions:

- 6-digit add and subtract
- 6-digit multiply and divide
- Square rote
- Average
- Standard deviation
- BCD to binary conversion
- Binary to BCD conversion

Applications

These arithmetic functions have applications in various industries such as food processing, machine tool work and material handling. Applications in these industries could be weighing, blending, batch processing, scaling, positioning, test stands, and heat treating. The square root function is frequently used for flow measurement and mining applications. The average function can be used for averaging thermocouple inputs or other process variables. Standard deviation and averaging have applications in trend analysis and report generation.

Manual's Purpose

This manual shows you how to install and program the AF1 in your Mini-PLC-2/15 controller.

Audience

We assume that you are familiar with programming and operation of the Mini-PLC- 2/15 and the Industrial Terminal (cat. no. 1770-T3). If this is not the case, refer to the appropriate publications or see our Publications Index (publication SD499).

WARNING: Use only Allen-Bradley authorized programming devices to program Allen-Bradley programmable controllers. using unauthorized programming devices may result in unexpected operation, possibly causing equipment damage and/or injury to personnel.

Chapter 2

Installation

General

During AF1 installation, take special care not to bend or contaminate the pins. Bent or dirty pins can prevent proper AF1 programming and use. The AF1's transparent window is covered with the product label to avoid accidental alteration of memory from uv light sources. Do not remove this label. Store the AF1 in its shipping container when not installed in a Mini–PLC–2/15 processor.

Installation/Removal Handling Precautions

The AF1 can be damaged during routine handling if proper precautions are not taken to reduce static electricity discharges.

Recommended precautions include:

- Handle the AF1 by the case without touching its pins.
- Use a static free work station.
- Wear a conductive wrist strap which has a minimum 200k ohms resistance and is connected to earth ground.
- Ground tools prior to contacting the AF1.
- Connect static–free work station to ground through a minimum 200k ohm resistance.
- Control the relative humidity of the installation area–ideal conditions are 40% to 60% relative humidity.

The following is a list of things that should not be done:

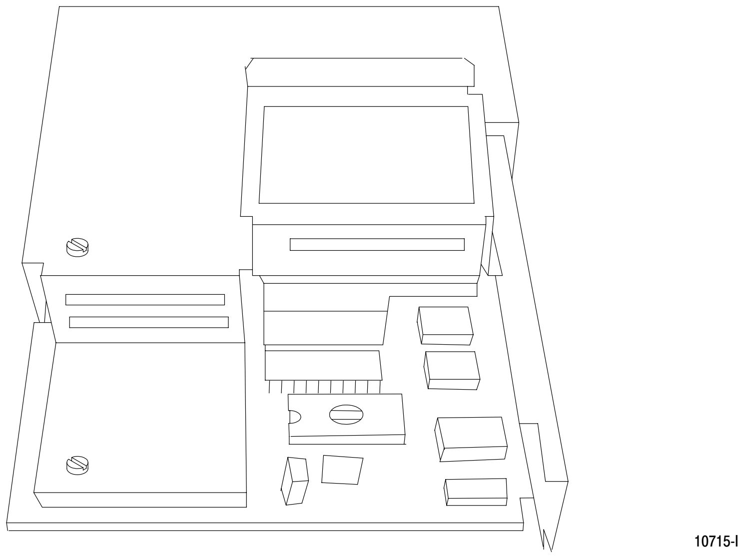
- Do not handle styrofoam, plastic, or cellophane–covered articles such as combs, cigarette packages, and candy immediately prior to handling an AF1.
- Do not hand the AF1 to someone who is not antistatic protected.
- Do not install the AF1 in areas which might contaminate or foul the pins of the AF1 device.
- Do not handle the AF1 by its pins.
- Do not slide the AF1 across any surface.
- Do not place the AF1 in a non–conductive plastic bag.

When these precautions are followed, the potential difference between the AF1 pins is reduced thereby reducing the problems associated with static discharges.

Installation

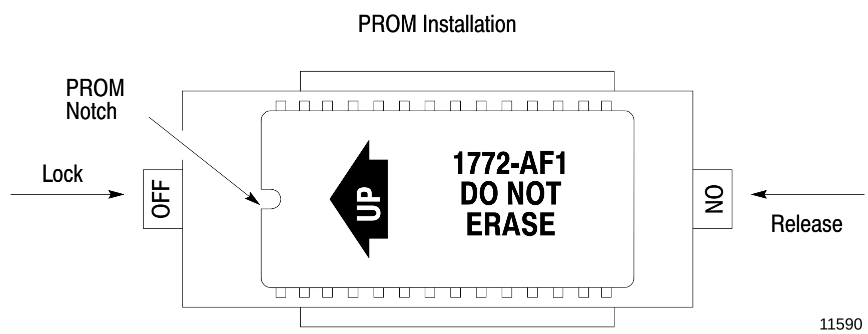
The AF1 fits into a 28–pin ZIF (zero insertion force) socket, which is located under a hinged door at the lower side of the Mini–PLC–2/15 processor (Figure 2.1).

Figure 2.1
PROM Socket



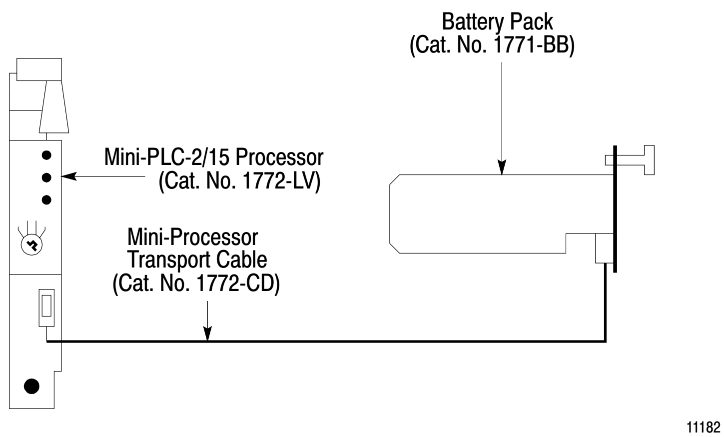
On the underside of the PROM door is a label that illustrates PROM installation. The notch on the AF1 PROM, when installed, must correspond to the notch shown on the label. Figure 2.2 shows a properly installed AF1.

Figure 2.2
AF1 Installed



To access the PROM socket, remove the Mini–PLC–2/15 processor module from the I/O chassis. If you desire to maintain processor memory contents, connect an external battery pack (Figure 2.3) to the processor with the Mini–Processor Transport Cable (cat. no. 1772–CD) prior to removing the module from the chassis.

Figure 2.3
External Battery Backup



To install the AF1, perform the following steps (Figure 2.2)

1. Turn the mode select switch to PROG.

2. Remove AC power from the I/O chassis power supply.

3. Remove the processor module from the I/O chassis.

4. Check all AF1 pins to ensure they are not bent or dirty.

5. Loosen the screw and lift the PROM door.

6. Push the ON tab in to unlock the socket.

7. Position the AF1 as shown in Figure 2.2 Be sure the notch on your AF1 PROM faces the OFF tab.

8. Line up the AF1 as shown in Figure 2.2 and seat in the socket. Be sure the pins are aligned as they bend easily.

9. Lock the AF1 in place by pushing the OFF tab in.

10. Close the PROM door and tighten the screw.

Removal

To remove the AF1, perform the following steps:

1. Turn the mode select switch to PROG.

2. To maintain processor memory contents connect an external battery pack to the processor with the mini–processor transport cable (Figure 2.3).

3. Remove AC power from the I/O chassis power supply.

4. Remove the processor module from the I/O chassis.

5. Loosen the screw, lift up the PROM door, and push the ON tab in to unlock the socket (Figure 2.2).

6. Carefully remove the AF1 and store it in its shipping container.

Chapter 3

Programming

General

You access the AF1 by pressing [SHIFT][EAF] (execute auxiliary function) or [SHIFT] [SCT] on the keyboard of your Industrial Terminal (cat. no. 1770-T3). The instruction is an output instruction and may be preceded on a rung by input instructions. Once you enter the function, the block diagram of Figure 3.1 appears on the CRT. To program a specific mathematics function, you would enter the appropriate function number (Table 3.A). If you enter a non-existent function number, the following occurs:

When the processor attempts to execute a function number which does not exist on the AF1, the response of the processor depends upon whether the keyswitch is in the RUN or RUN/PROGRAM position.

The response are:

In the RUN position, the processor stops running and the CRT displays PROCESSOR FAULT and CHANGE PROCESSOR TO PROGRAM MODE. The processor and memory LEDs illuminate. After you change processor operation to program mode the LEDs turn off the CRT displays MODE SELECTION menu and PLC-2 RUN TIME ERROR,PRESS 11 TO CONTINUE. When you press 11 the CRT displays and intensifies the rung containing the illegal opcode and states ILLEGAL OPCODE INTENSIFIED INSTRUCTION LINKED WITH CAUSE OF ERROR.

In the RUN/PROGRAM position, the processor stops running and the CRT displays MODE SELECTION menu and PLC-2 RUN TIME ERROR, PRESS 11 TO CONTINUE. When you press 11 the CRT displays and intensifies the rung containing the illegal opcode and states ILLEGAL OPCODE INTENSIFIED INSTRUCTION LINKED WITH CAUSE OF ERROR.

Table 3.A
AF1 Function Numbers

Function Number	Mathematical Operation
01	Add
02	Subtract
03	Multiply
04	Divide
05	Square root
06	Average
07	Standard deviation
13	BCD to binary conversion
14	Binary to BCD conversion

You enter an existent function number and then enter data and result addresses (we will explain this in detail later). The processor then places a number in the data address.

AF1 Function Sequence

When the Mini-PLC-2/15 controller encounters an AF2 function during program execution and the rung is true, the processor performs the following steps:

1. Saves its present position in the user program.

2. The interlock system (see Avoiding Excessive AF1 Execution Times) grants access to the AF1 function.

3. Reads the operand's data stored in the data address that you entered.

4. Reads the result address which you entered.

5. Obtains the location of the mathematical routine requested by the function number.

6. Executes the routine in the AF1 area. (See section for excessive execution time.)

7. Writes the results at the result address in the data table.

8. Returns program execution to the next instruction in the user's program after the AF1 function is completed. (See section for excessive execution time.)

9. Readies itself for the next AF1 operation.

AF1 Automatic Checks

To guard against improper program execution, automatic check routines are incorporated in the AF1. The processor uses these routines to prevent the following:

- Executing AF1 functions having invalid function addresses
- Spending so much time executing AF1 functions that the controller neglects its main program and I/O scans

Invalid Function Addresses

Valid AF1 function addresses include the I/O image table and the data table (except word 027). Specifically, valid addresses are from 010 to 026, from 030 to 077, and from 110 to the end of the data table. Result addresses must not reside in the input image table.

When a user programmed function has an invalid address, the response of the processor depends upon whether the keyswitch is in the RUN or RUN/PROGRAM position.

The responses are:

In the RUN position, the processor stops running and the CRT displays PROCESSOR FAULT and CHANGE PROCESSOR TO PROGRAM MODE. The processor and memory LEDs illuminate. After you change processor operation to program mode the LEDs turn off and the CRT displays MODE SELECTION menu and PLC-2 RUN TIME ERROR, PRESS 11 TO CONTINUE. When you press 11 the CRT displays and intensifies the rung containing the illegal address and states ILLEGAL ADDRESS INTENSIFIED INSTRUCTION LINKED WITH CAUSE OF ERROR.

In the RUN/PROGRAM position, the processor stops running and the CRT displays MODE SELECTION menu and PLC-2 RUN TIME ERROR, PRESS 11 TO CONTINUE. When you press 11 the CRT displays and intensifies the rung containing the illegal address and states ILLEGAL ADDRESS INTENSIFIED INSTRUCTION LINKED WITH CAUSE OF ERROR.

Avoiding Excessive AF1 Execution Times

Table 3.B lists execution times for AF1 functions. To avoid excessive AF1 function execution times, an interlock system is designed into the AF1. This system automatically checks and does the following:

- Permits no AF1 function to run longer than 6ms without returning processor scan to the processor.

- During a program scan each true AF1 function rung which can be completed in a single scan will be completed as it is encountered. However, upon encountering a true AF1 function rung which requires multiple program scan to complete, all other true AF1 function rungs will be "locked out" until sufficient program scans complete the active AF1 function rung.

- Once started, it completes an AF1 function prior to starting the next AF1 function encountered in the user program which has a true rung condition.
- Limits the number of enabled AF1 functions in a program to 50. You may include more functions but you must ensure that no more than 50 are enabled at one time. This requirement only applies where you have programmed a function that requires more than one scan to complete.

Table 3.B
Execution Time [1]

Function	Avg. Time	Worst Time	Number or Scans
Addition	1.22ms	1.27ms	1
Subtraction	1.22ms	1.27ms	1
Multiplication	4.99ms	5.28ms	1
Division	16.17ms	21.76ms	4
Square Root	6.08ms	7.11ms	1
Average	12.33ms +0.29ms per value	-	4 + N / 4 [2]
Standard Deviation	94.16ms +2.09ms per value	-	22 + 5N / 4 [2]
BCD to Binary	0.89ms	-	1
Binary to BCD	0.84ms	-	1

[1] These times are calculated for a single AF1 function. Overhead for AF1 lock maintenance and multiple rungs through the ladder program to complete some function are included
[2] N=number of values whose average or standard deviation is sought.

The time listed in Table 3.B includes:

- Overhead for AF1 PROM interlock system
- One run through the portion of the AF1 specified by the particular function

To obtain the time required from activation of the input that makes the rung containing the AF1 PROM function true until the correct answer for the function is in the data table, you must add the following times to the values in Table 3.B:

- Input delay time (from specification for specific input)
- One program scan time and one I/O scan time multiplied by the number of scans specified in Table 3.B.

Methods for determining these times are presented in Mini-PLC-2/15, series B, Programmable Controller Programming and Operations Manual (publication 1772-804).

Programming Specific Mathematical Functions

In this section we explain the following for each of the AF1 functions:

- What it is
- How to enter it in your program
- Its format in the data table

  a. word arrangement

  b. digit location

- Sample entry and display rungs. Although there are several techniques to enter this data, we use get instructions.
- Error messages. If an AF1 function has special error message responses to specific illegal programming procedures, we state these responses.

Status Bits

The most significant four bits of the most significant word of the result data area are reserved for status bits. These bits have the following meanings:

Enable = bit 17
Sign = bit 16
Done = bit 15
Error bit = bit 14

The enable bit is set at the start of an AF1 function and reset upon completion.

The sign bit, if set, indicates a negative value.

The done bit is rest at the start of an AF1 function and set upon completion.

The error bit is a general error flag that indicates overflow and invalid operand or result errors. Individual functions determine the actual state of this bit.

Throughout this manual, unused status bits are shown blank for the following reasons:

- Whether the content of an unused status bit in an input word is 0 or 1 is irrelevant as such bits are ignored in AF1 function execution.
- The AF1 reset unused status bits in result words. For simplicity these bits are left blank.

Accuracy

The accuracy of all function results on the Af1 is typically ±1 in the least significant digit.

AF1 Addition Function

An AF1 addition function operates on two 6-digit BCD numbers and presents the result in a third 6-digit BCD number

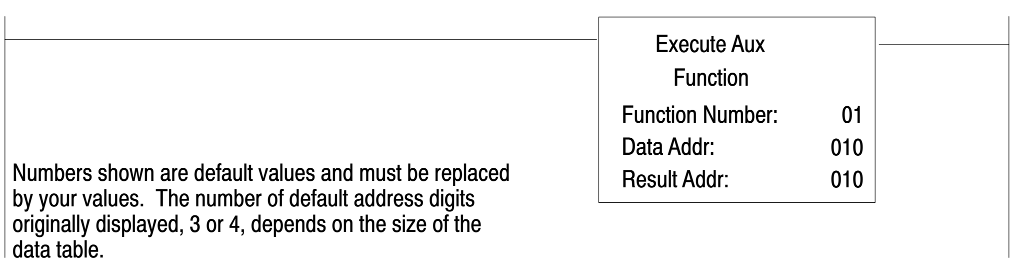
(±XXX XXX.) + (±XXX XXX.) = ±XXX XXX.

How to Enter an AF1 Addition Function

To program an AF1 addition function, perform the following steps:

1. Press [SHIFT][EAF] or [SHIFT][SCT] on the keyboard of your industrial terminal. Figure 3.1 appears on the CRT.

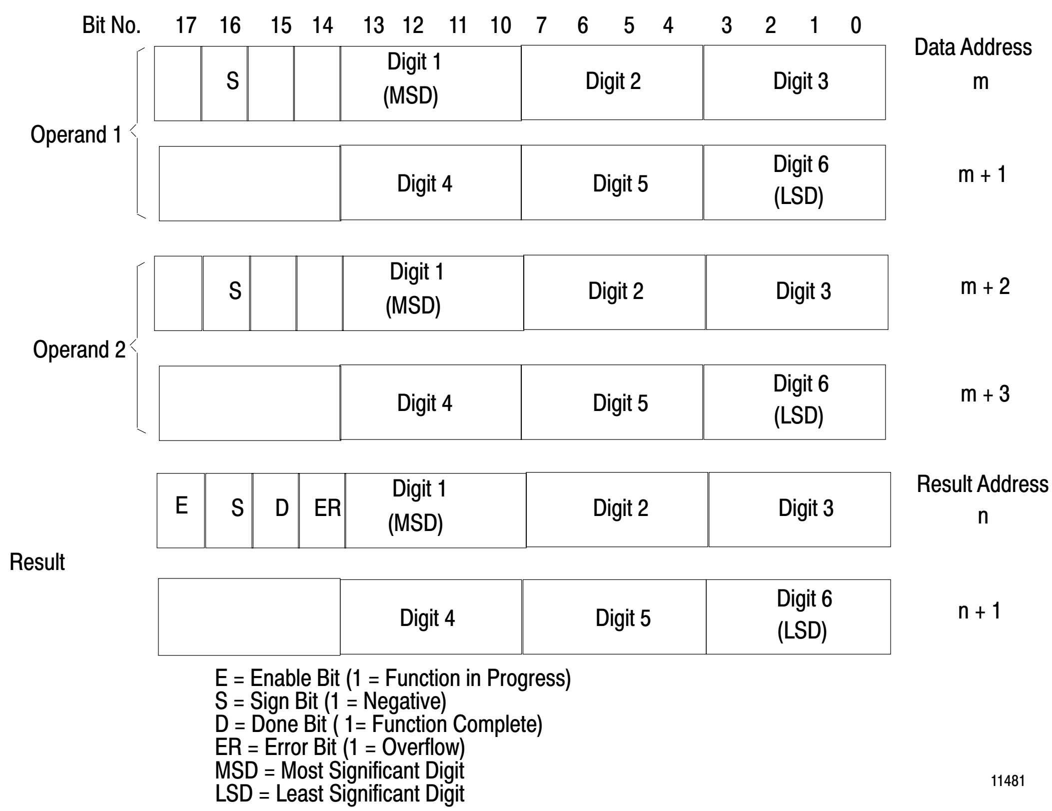
Figure 3.1
Execute Auxiliary Function Format



2. Enter 01, the function number for AF1 addition.

This entry identifies that the function entered is to perform an AF1 addition and that the processor use the data table format shown in Figure 3.2 when executed. Operands 1 and 2 represent the two 6-digit numbers you wish to add. The six digits of operand 1 are represented in BCD by the groups of bits labeled digit 1 through 6. Digit 1 and digit 6 are the most significant and the least significant digits respectively. This digit labeling system also applies to operand 2 and the result.

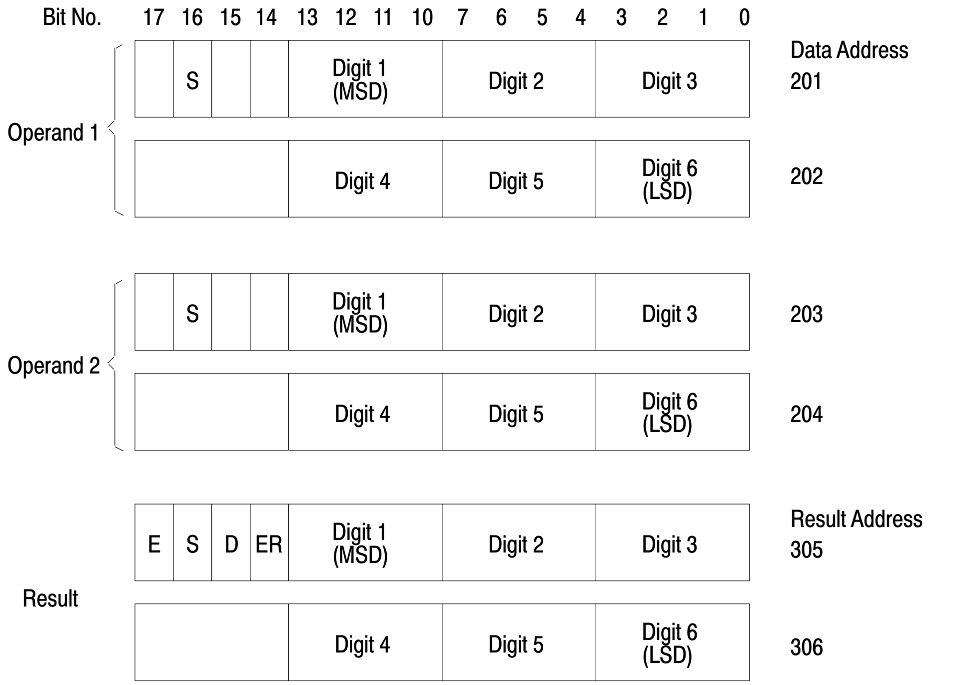
Figure 3.2
General AF1 Addition Function Word and Digit Format



3. Enter a data address and a result address.

If we select a data address of 201 and a result address of 305, the AF1 establishes the data table format shown in Figure 3.3. Be careful not to select data and result addresses so close together that the addresses of the operands following the data address overlap your result address. The data address eventually contains three digits of operand 1. The AF1 reserves the next three higher addresses for digits 4 through 6 of operand 1 and digits 1 through 6 of operand 2. The result address contains the most significant three digits of the result and the next higher address contains the least significant three digits.

Figure 3.3
AF1 Addition Function Format After Address Entry



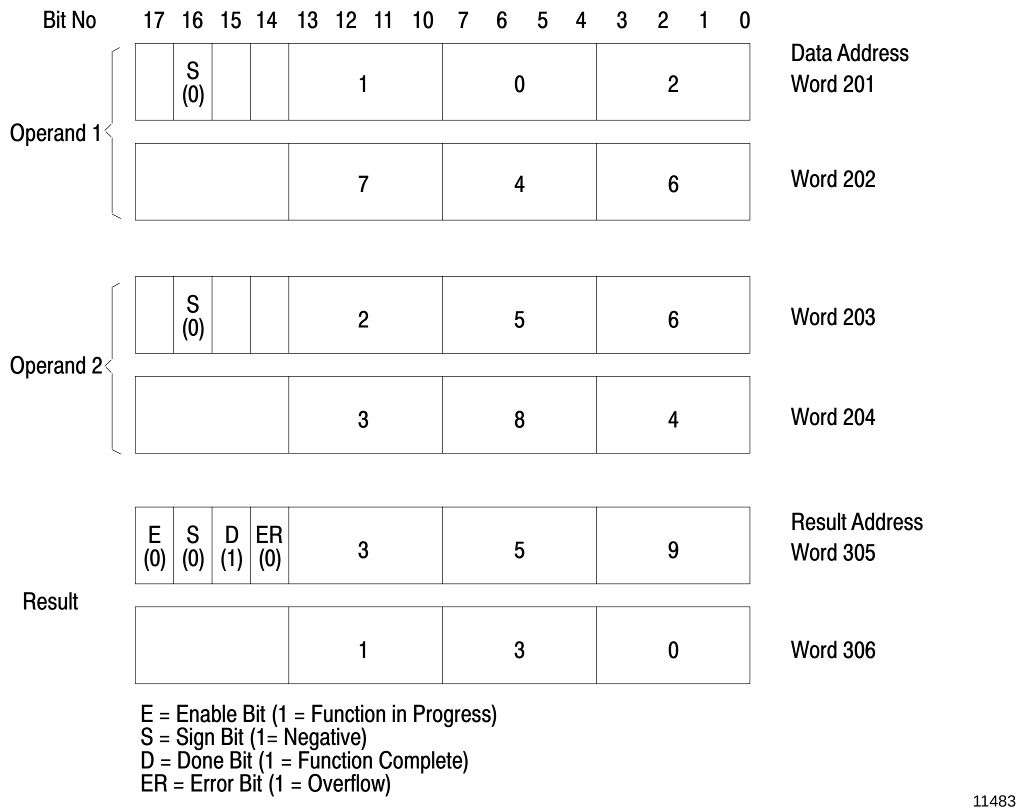
E = Enable Bit (1 = Function in Progress)
S = Sign Bit (1= Negative)
D = Done Bit (1 = Function Complete)
ER = Error Bit (1 = Overflow)
MSD = Most Significant Digit
LSD = Least Significant Digit

11482

4. Enter values for operands 1 and 2.

You can enter these values from the keyboard of your industrial terminal or through ladder diagram functions. Entry of operand 1 = 102746 and operand 2=256384 produces the result 359130 when the addition function executes. Figure 3.4 shows how the result is stored.

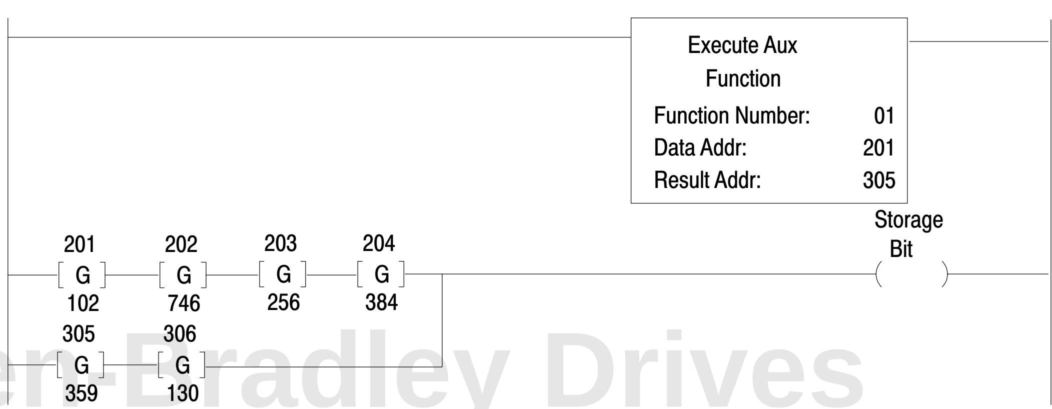
Figure 3.4
AF1 Addition Function Format After Execution



Entry and Display of Input and Result Values

Figure 3.5 shows one method for inserting input values and displaying input values and results of AF1 addition computations. Although there are several techniques for accomplishing this, we chose get instructions. The first rung requests an AF1 addition. The second rung shows the two operands in its top branch and the resultant sum in its lower branch.

Figure 3.5
AF1 Addition Function Input and Result Display Rungs



Error Message

If the resultant sum has more than six integers, the error bit (bit 14) is set indicating overflow.

AF1 Subtraction Function

An AF1 subtraction function operates on two 6-digit BCD numbers and presents the result in a third 6-digit BCD number.

$$(\pm XXX\ XXX.) - (\pm XXX\ XXX.) = \pm XXX\ XXX.$$

How to Enter an AF1 Subtraction Function

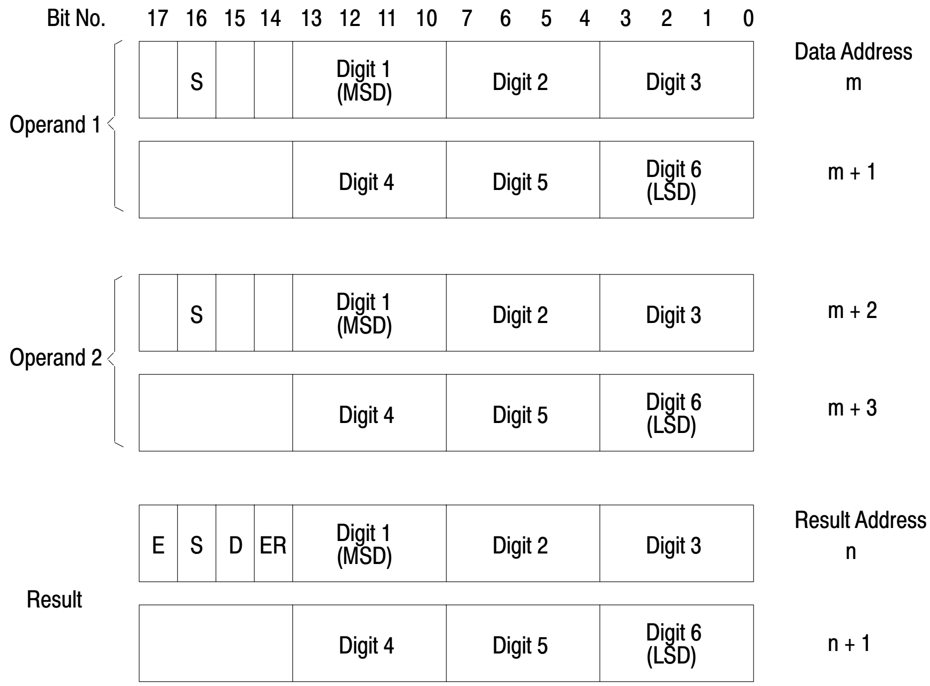
To program an AF1 subtraction function, perform the following steps:

1. Press [SHIFT] [EAF] or [SHIFT] [SCT] on the keyboard of your industrial terminal. Figure 3.1 appears on the CRT.

2. Enter 02, the function number for AF1 subtraction.

This entry identifies that the function entered is to perform an AF1 subtraction and that the processor use the data table format shown in Figure 3.6 when executed. Operands 1 and 2 represent the two 6-digit numbers whose difference you want to find. The six digits of operand 1 are represented in BCD by the group of bits labeled digit 1 through 6. Digit 1 and digit 6 are respectively. This digit labeling system also applies to operand 2 and the result.

Figure 3.6
General AF1 Subtraction Function Word and Digit Format



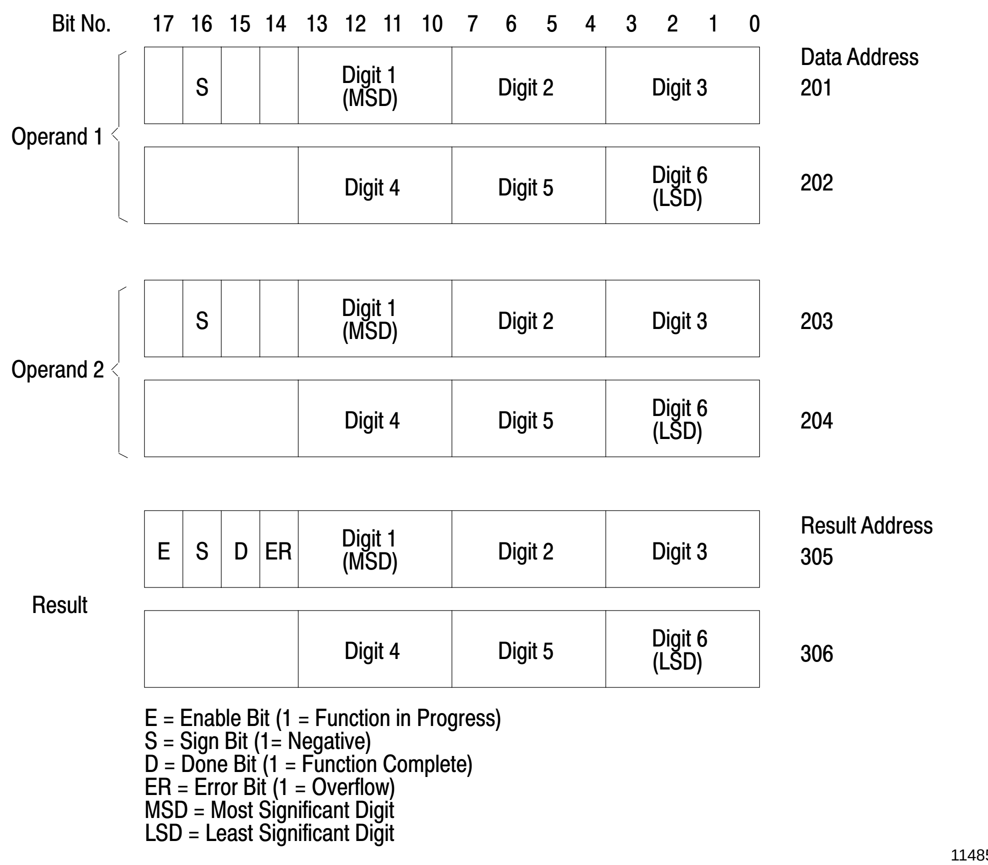
E = Enable Bit (1 = Function in Progress)
S = Sign Bit (1= Negative)
D = Done Bit (1 = Function Complete)
ER = Error Bit (1 = Overflow)
MSD = Most Significant Digit
LSD = Least Significant Digit

11484

3. Enter a data address and result address.

If we select a data address of 201 and a result address of 305, the AF1 establishes the data table format shown in Figure 3.7. The data address eventually contains three digits of operand 1. The AF1 reserves the next three higher addresses for digits 4 through 6 of operand 1, and digits 1 through 6 of operand 2. The result address contains the most significant three digits of the result and the next higher address contains the least significant three digits.

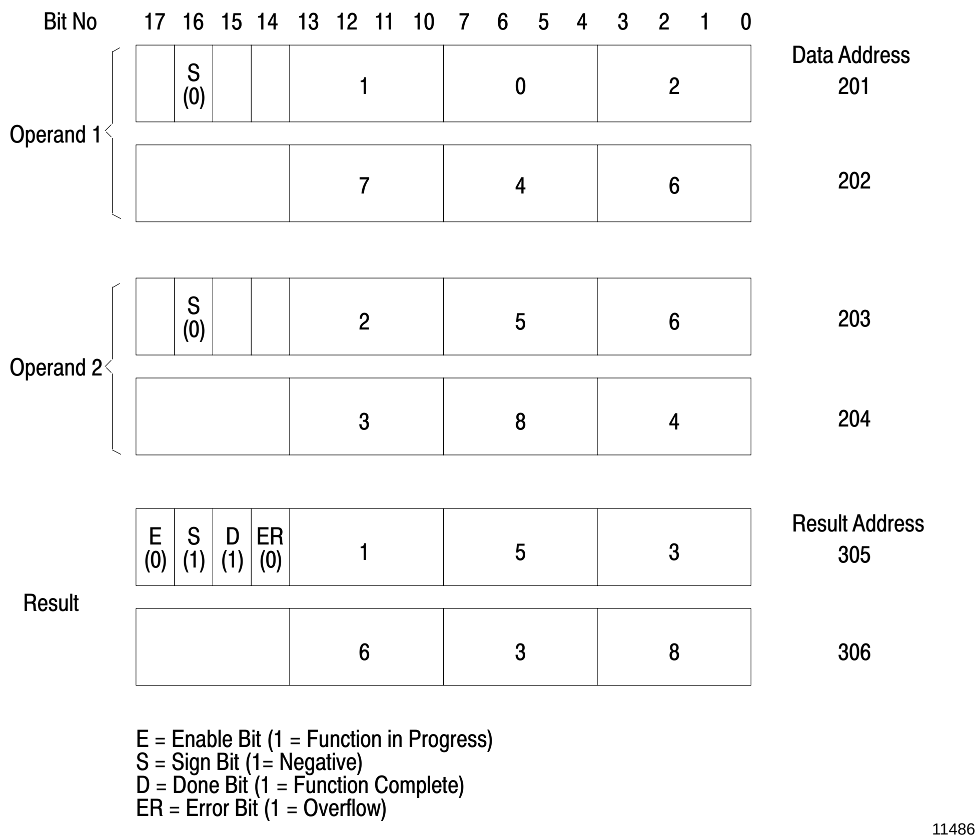
Figure 3.7
AF1 Subtraction Function Format After Address Entry



4. Enter values for operands 1 and 2.

You can enter these values from the keyboard of your industrial terminal or through ladder diagram functions. Entry of operand 1 = 102746 and operand 2 - 256384 produces the result -153638 when the subtraction function executes. Figure 3.8 shows how the result is stored.

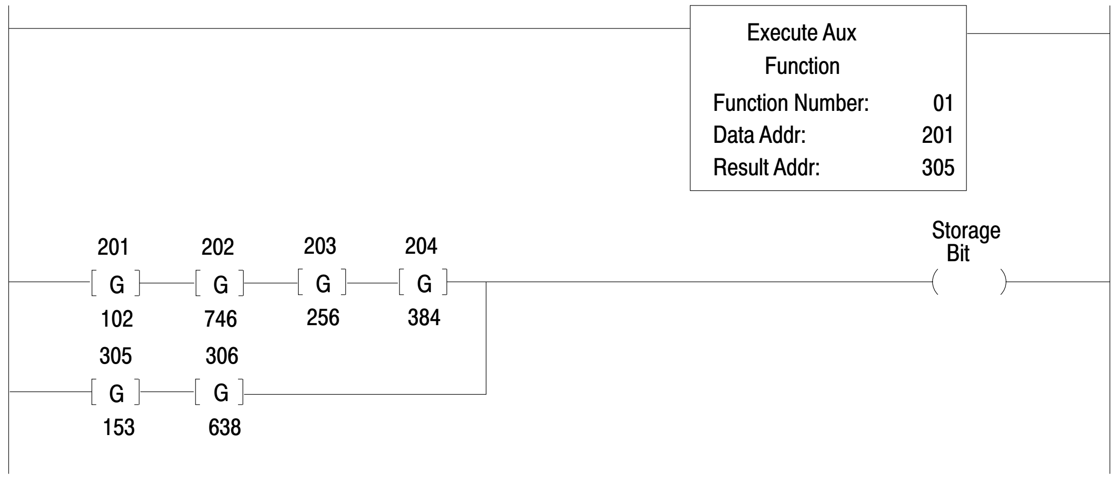
Figure 3.8
AF1 Subtraction Function Format After Execution



Entry and Display of Input and Result Values

Figure 3.9 shows one method for inserting input values and displaying input values and results of AF1 subtraction computations. Although there are several techniques for accomplishing this, we chose get instructions. The first rung requests an AF1 subtraction. The second rung shows the two operands in its top branch and the resultant difference in its lower branch.

Figure 3.9
AF1 Subtraction Function Input and Result Display Rungs



Error Message

If the result has more than six integers, the error bit (bit 14) is set indicating overflow.

AF1 Multiplication Function

An AF1 multiplication function operates on two 6-digit BCD numbers and presents the results in a 12-digit BCD number.

$$(\pm XXX\ XXX.)\ X\ (\pm\ XXX\ XXX.) = \pm XXX\ XXX\ XXX\ XXX.$$

How to Enter an AF1 Multiplication Function

To program an AF1 multiplication function, perform the following steps:

1. Press [SHIFT][EAF] or [SHIFT][SCT] on the keyboard of your industrial terminal. Figure 3.1 appears on the CRT.

2. Enter 03, the function number for AF1 multiplication.

This entry identifies that the function entered is to perform an AF1 multiplication and that the processor use the data table format shown in Figure 3.10 when executed. Operands 1 and 2 represent two 6-digit numbers whose product you want to find. The six digits of operand 1 are represented in BCD by groups of bits labeled digit 1 through 6. Digit 1 and 6 are most significant and least significant digits respectively. Operand 2 and the 12 digits of the result are labeled similarly.

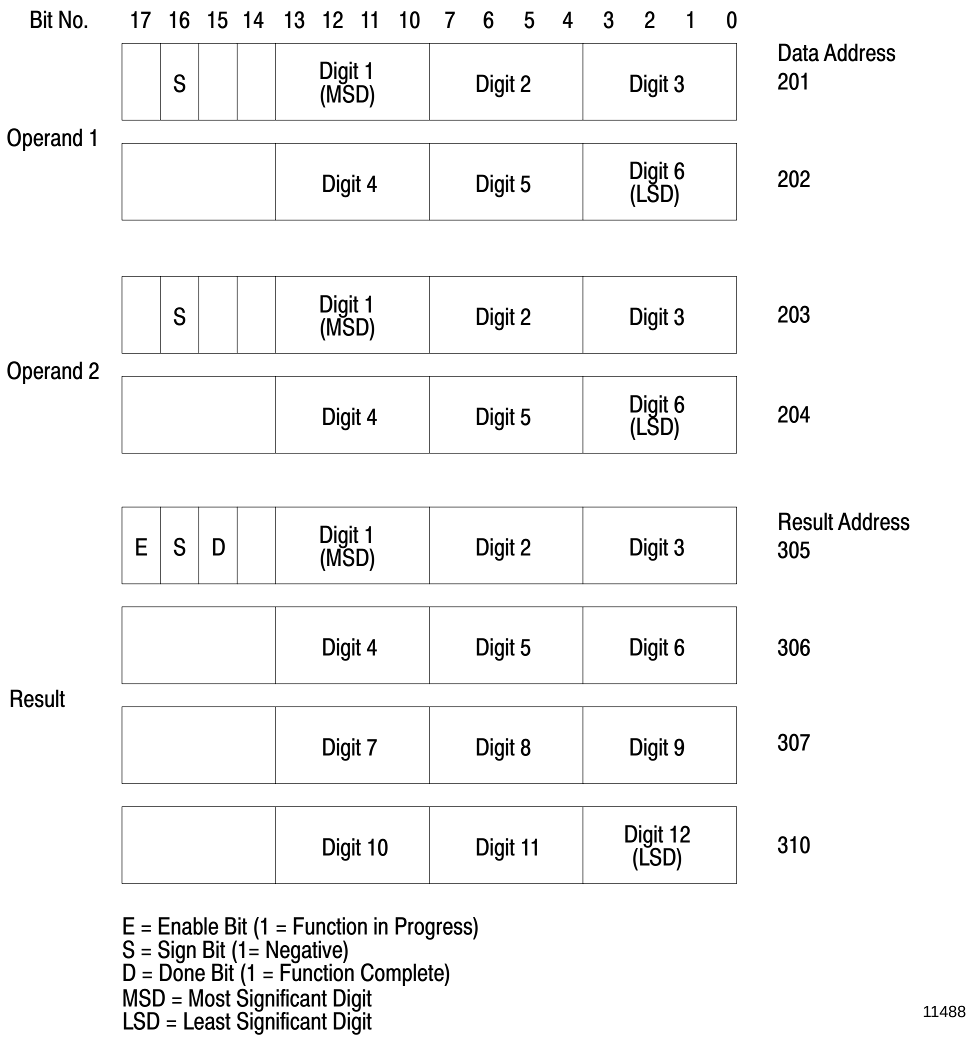
Figure 3.10
General AF1 Multiplication Function Word and Digit Format

Bit No.	17	16	15	14	13 12 11 10	7 6 5 4	3 2 1 0	
Operand 1		S			Digit 1 (MSD)	Digit 2	Digit 3	Data Address m
					Digit 4	Digit 5	Digit 6 (LSD)	m + 1
Operand 2		S			Digit 1 (MSD)	Digit 2	Digit 3	m + 2
					Digit 4	Digit 5	Digit 6 (LSD)	m + 3
Result	E	S	D		Digit 1 (MSD)	Digit 2	Digit 3	Result Address n
					Digit 4	Digit 5	Digit 6	n + 1
					Digit 7	Digit 8	Digit 9	n + 2
					Digit 10	Digit 11	Digit 12 (LSD)	n + 3

E = Enable Bit (1 = Function in Progress)
S = Sign Bit (1= Negative)
D = Done Bit (1 = Function Complete)
MSD = Most Significant Digit
LSD = Least Significant Digit

11487

Figure 3.11
AF1 Multiplication Function Format After Address Entry



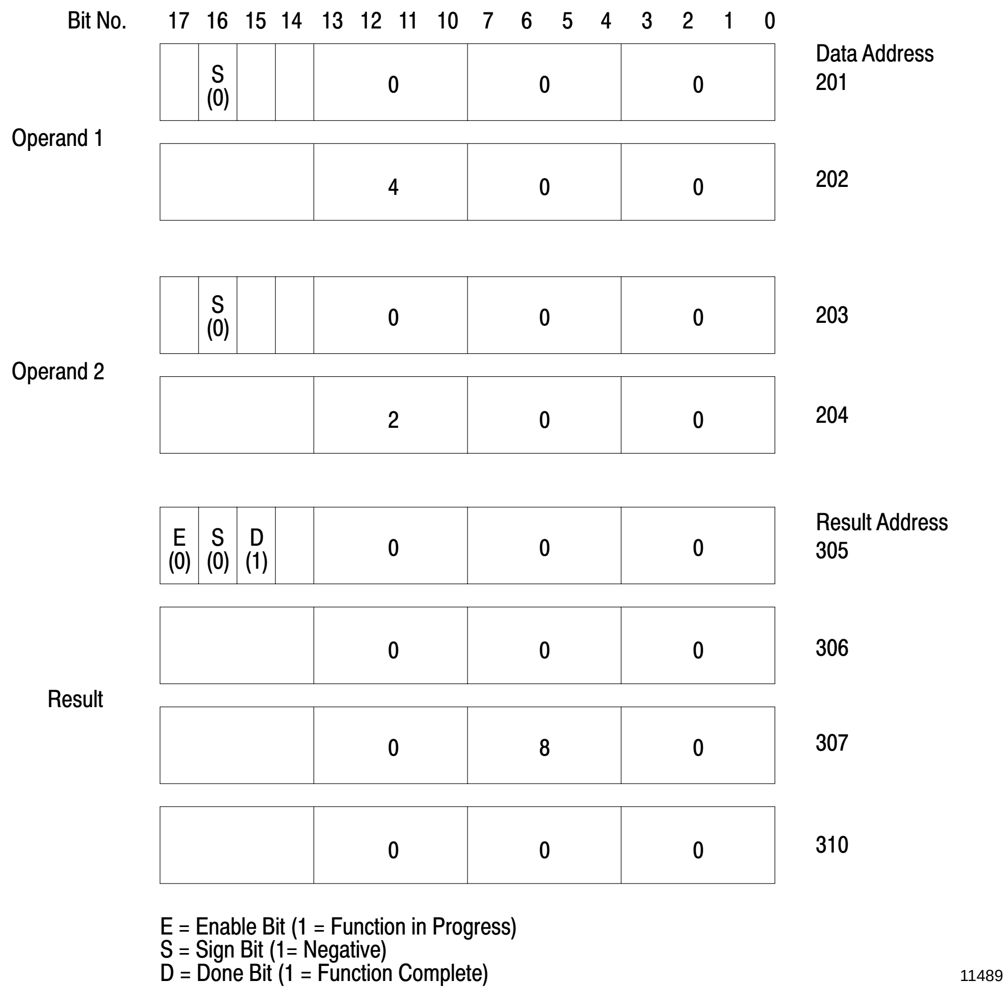
3. Enter a data address and a result address.

If we enter a data address of 201 and a result address of 305, the AF1 establishes the data table format shown in Figure 3.11. The data address eventually contains the most significant three digits of operand 1. The AF1 reserves the next three higher addresses for the least significant three digits of operand 1 and the six digits of operand 2. The result address contains the most significant three digits of the result. The AF1 reserves the next three higher addresses for the remaining nine digits of the result.

4. Enter values for operands 1 and 2.

You can enter these values from the keyboard of your industrial terminal or through ladder diagram functions. Entry of operand 1 = 000400 and operand 2 = 000200 produces the result 000000080000 (Figure 3.12).

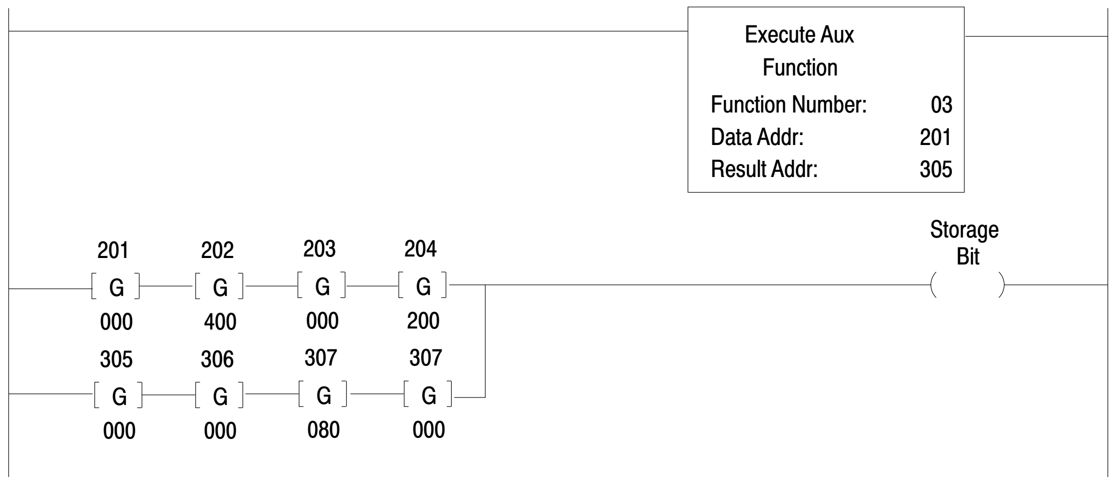
Figure 3.12
AF1 Multiplication Function Format After Execution



Entry and Display of Input and Result Values

Figure 3.13 shows one method you can use to enter values for operands 1 and 2 and for displaying the results of an AF1 multiplication. Although there are several techniques for accomplishing this, we chose get instructions. The first rung requests an AF1 multiplication. The top branch of the second rung shows the two 6-digit operands while the lower branch shows the 12 digit product.

Figure 3.13
AF2 Multiplication Function Input and Result Display Rungs



AF1 Division Function

An AF1 division function operates on two 6-digit BCD numbers and presents the results in a 12-digit BCD number.

$$(\pm XXX\ XXX.) : (\pm XXX\ XXX.) = \pm XXX\ XXX.XXX\ XXX$$

How to Enter an AF1 Division Function

To program an AF1 division function, perform the following steps:

1. Press [SHIFT] [EAF] or [SHIFT] [SCT] on the keyboard of your industrial terminal. Figure 3.1 appears on the CRT.

2. Enter 04, the function number for AF1 division.

This entry identifies that the function entered is to perform an AF division and that the processor use the data table format shown in Figure 3.14 when executed. Operands 1 and 2 represent two 6-digit numbers whose quotient you wish to find. The six digits of operand 1 are represented in BCD by groups of bits labeled digit 1 through 6. Digit 1 and 6 are the most significant and least significant digits respectively. Operand 2 and the 12 digits of the result are labeled similarly.

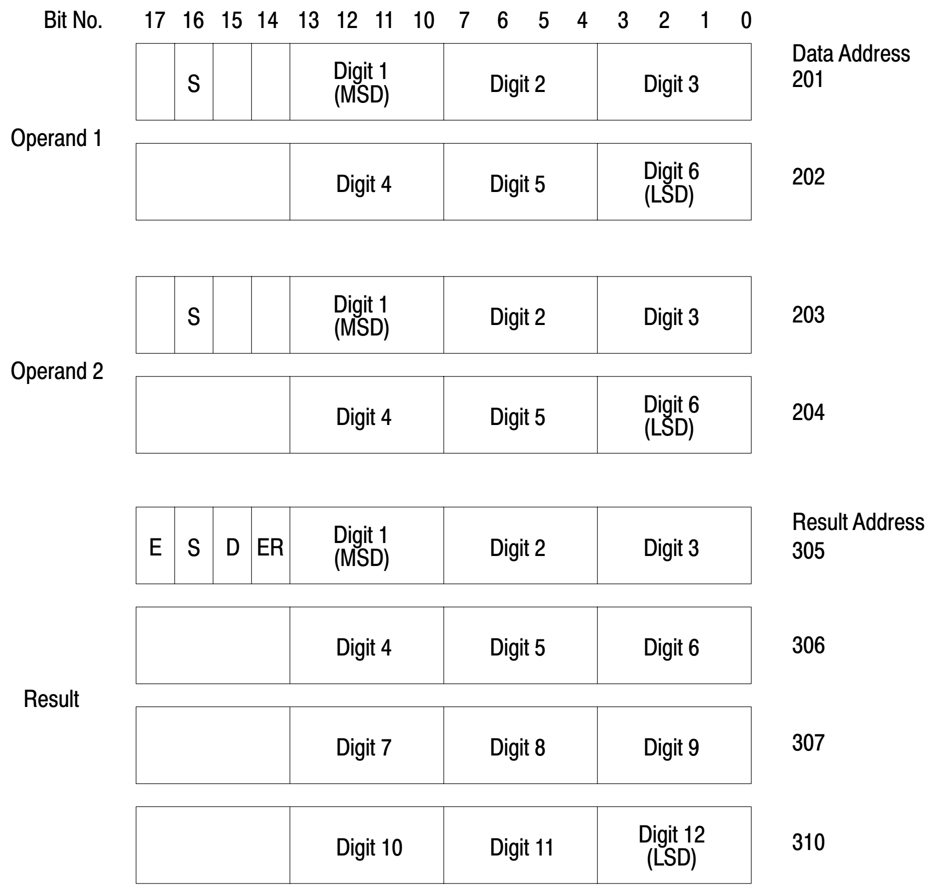
Figure 3.14
General AF1 Division Function Word and Digit Format

Bit No.	17	16	15	14	13 12 11 10	7 6 5 4	3 2 1 0	
Operand 1		S			Digit 1 (MSD)	Digit 2	Digit 3	Data Address m
					Digit 4	Digit 5	Digit 6 (LSD)	m + 1
Operand 2		S			Digit 1 (MSD)	Digit 2	Digit 3	m + 2
					Digit 4	Digit 5	Digit 6 (LSD)	m + 3
Result	E	S	D	ER	Digit 1 (MSD)	Digit 2	Digit 3	Result Address n
					Digit 4	Digit 5	Digit 6	n + 1
					Digit 7	Digit 8	Digit 9	n + 2
					Digit 10	Digit 11	Digit 12 (LSD)	n + 3

E = Enable Bit (1 = Function in Progress)
S = Sign Bit (1= Negative)
D = Done Bit (1 = Function Complete)
ER = Illegal Operand (Divide by Zero)
MSD = Most Significant Digit
LSD = Least Significant Digit

11490

Figure 3.15
AF1 Division Function Format After Address Entry



E = Enable Bit (1 = Function in Progress)
S = Sign Bit (1= Negative)
D = Done Bit (1 = Function Complete)
ER = Illegal Operand (Divide by Zero)
MSD = Most Significant Digit
LSD = Least Significant Digit

11491

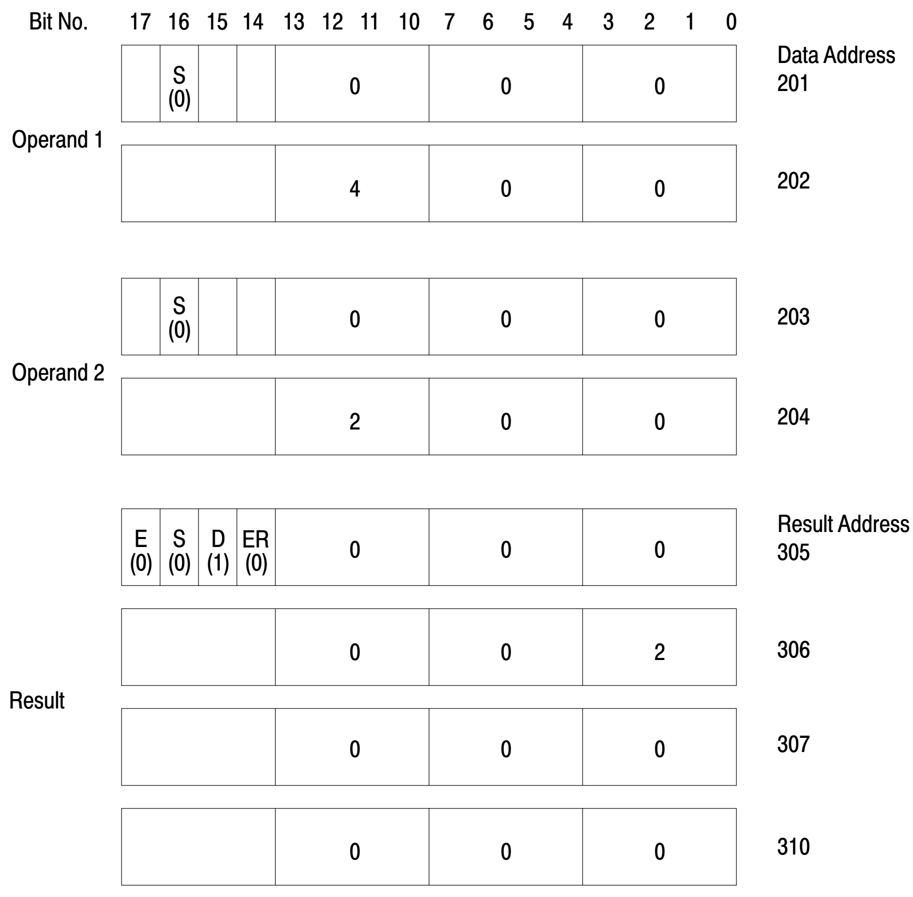
3. Enter a data address and a result address.

If we enter a data address of 201 and a result address of 305, the AF1 establishes the data table format shown in Figure 3.15. The data address eventually contains the most significant three digits of operand 1. The AF1 reserves the next three higher addresses for the least significant three digits of operand 1 and the six digits of operand 2. The result address contains the most significant three digits of the result. The AF1 reserves the next three higher addresses for the remaining nine digits of the result.

4. Enter values for operands 1 and 2.

You can enter these numbers from the keyboard of your industrial terminal or through ladder diagram functions. Entry of operand 1 = 000400 and operand 2 = 000200 produces the result 000002.000000 (Figure 3.16).

Figure 3.16
AF1 Division Function Format After Execution

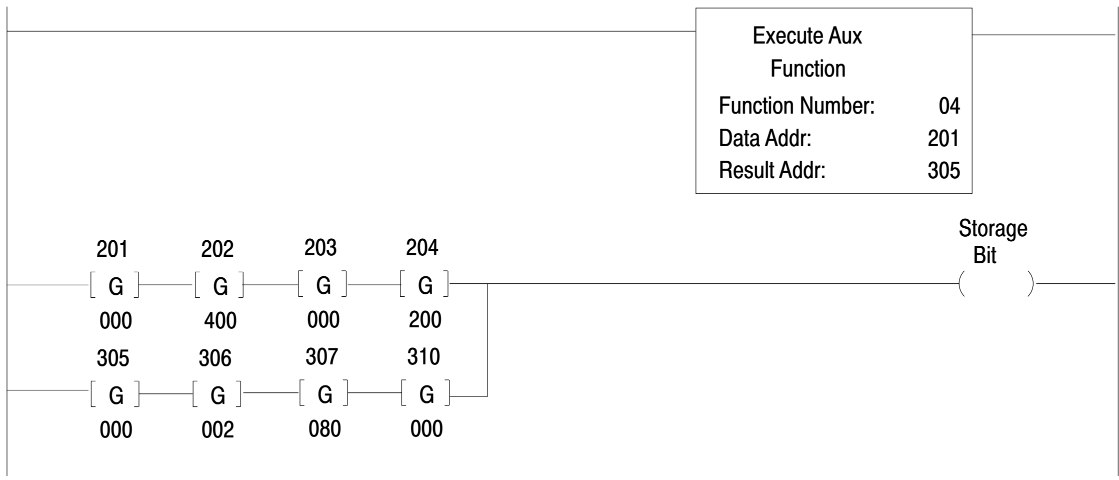


E = Enable Bit (1 = Function in Progress)
S = Sign Bit (1= Negative)
D = Done Bit (1 = Function Complete)
ER = Illegal Operand (Divide by Zero)

Entry and Display of Input and Result Values

Figure 3.17 shows one method you can use to enter values for operands 1 and 2 and for displaying the results of an AF1 division. Although there are several techniques for accomplishing this, we chose get instructions. The first rung requests an AF1 division function. The top branch of the second rung shows the two 6-digit operands while the lower branch shows the 12 digit quotient.

Figure 3.17
AF1 Division Function Input and Result Display Rungs



Error Message

If you divide by zero, the error bit (bit 14) is set and the result reads zero.

AF1 Square Root Function

The AF1 square root function operates on a 3-digit BCD integer and presents the result in a 6-digit BCD number composed of three integer digits and 3 decimal digits. The first integer digit and the last decimal digit (digit 1 and digit 6) are always zero.

$$(\pm XXX.)^{1/2} = \pm XXX.XXX$$

A decimal point is implied between digits 3 and 4. the function calculates the square root of the absolute input value and sets the sign bit in the output to the sign of the input. The result is accurate to $\pm$.01.

How to Enter an AF1 Square Root Function

To program an AF1 square root function, perform the following steps:

1. Press [SHIFT] [EAF] or [SHIFT][SCT] on the keyboard of your industrial terminal. Figure 3.1 appears on the CRT.

2. Enter 05, the function number for AF1 square root.

This entry identifies that the function entered is to perform an AF1 square root and that the processor use the data table format shown in Figure 3.18 when executed. The three digits of the operand (the number whose square root you desire) are represented in BCD by the groups of bits labeled digit 1 through 3. The six digits of the result are labeled digit 1 through 6. Digit 1 and digit 6 in the result are always zero.

Figure 3.18
General AF1 Square Root Function Word and Digit Format

Bit No.	17	16	15	14	13 12 11 10	7 6 5 4	3 2 1 0	
Operand		S			Digit 1 (MSD)	Digit 2	Digit 3 (LSD)	Data Address m
Result	E	S	D		Digit 1 (MSD) (Always = 0)	Digit 2	Digit 3	Result Address n
					Digit 4	Digit 5	Digit 6 (LSD) (Always = 0)	n + 1

E = Enable Bit (1 = Function in Progress)
S = Sign Bit (1= Negative)
D = Done Bit (1 = Function Complete)
MSD = Most Significant Digit
LSD = Least Significant Digit

11582

3. Enter a data address and a result address.

If we enter a data address of 200 and a result address of 305, the AF1 establishes the data table format shown in Figure 3.19. The data address eventually contains the three digits of the number whose square root is sought. The result address (word 305) contains the first three digits (integer part) of the result. The three decimal digits of the result are stored in the next higher address, word 306. The implied decimal point is between digits 3 and 4.

Figure 3.19
AF1 Square Root Function Format After Address Entry

Bit No.	17	16	15	14	13 12 11 10	7 6 5 4	3 2 1 0	
Operand		S			Digit 1 (MSD)	Digit 2	Digit 3 (LSD)	Data Address 200
Result	E	S	D		Digit 1 (Always = 0)	Digit 2 (MSD)	Digit 3	Result Address 305
					Digit 4	Digit 5	Digit 6 (LSD) (Always = 0)	306

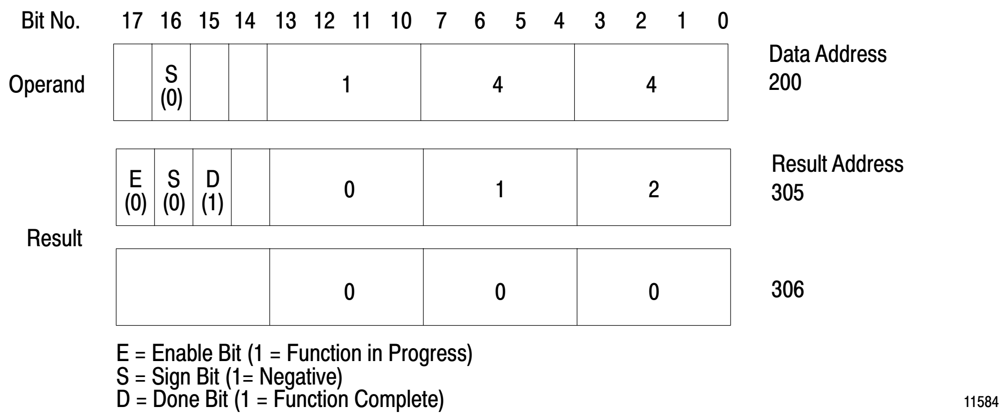
E = Enable Bit (1 = Function in Progress)
S = Sign Bit (1= Negative)
D = Done Bit (1 = Function Complete)
MSD = Most Significant Digit
LSD = Least Significant Digit

11583

4. Enter the value for the operand.

You can enter the value from the keyboard of your industrial terminal or through ladder diagram functions. Entry of operand = 144 produces the result 12 when the square root function executes. Figure 3.20 shows how the result is stored. The result is accurate to $\pm$0.01.

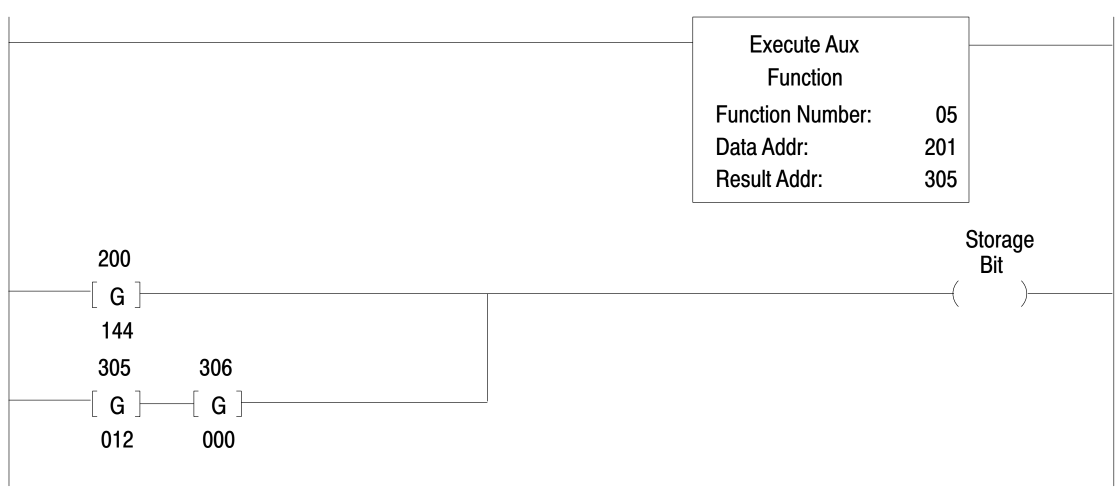
Figure 3.20
AF1 Square Root Function Format After Execution



Entry and Display of Input and Result Values

Figure 3.21 shows one method for inserting the input value and displaying the input value and result of an AF1 square root. Although there are several techniques for accomplishing this, we chose get instructions. The first rung requests an AF1 square root function. The second rung shows the operand 144 in word 200 and the resultant square root 012.000 in words 305 and 306.

Figure 3.21
AF1 Square Root Function Input and Result Display Rungs



AF1 Average Function

The AF1 average function determines the average of a group of N three digit integers. The numbers are in BCD format.

$$\frac{x_1x_1x_1. + x_2x_2x_2x_2. + x_3x_2x_2. + \ldots x_Nx_Nx_Nx_N.}{N} = xxx.xxx$$

The result is a 6 digit number composed of a 3-digit integer and a 3-digit decimal fraction. The maximum number of values you can average is 999 or is limited by the data table area available.

How to Enter an AF1 Average Function

To program an AF1 average function, perform the following steps:

1. Press [SHIFT] [EAF] or [SHIFT] [SCT] on the keyboard of your industrial terminal. Figure 3.1 appears on the CRT.

2. Enter 06, the function number for the AF1 average function.

This entry identifies that the function entered is to perform an AF1 average and that the processor use the data table format shown in Figure 3.22 when executed. The three digits showing the number of inputs and the three digits of each input value are represented in BCD by groups of bits labeled digit 1 through 3. The 6-digits of the results are labeled digit 1 through 6.

Figure 3.22
General AF1 Average Function Word and Digit Format

Bit No.	17	16	15	14	13 12 11 10	7 6 5 4	3 2 1 0	
Number of Inputs					Digit 1 (MSD)	Digit 2	Digit 3 (LSD)	Data Address m
Input 1		S			Digit 1 (MSD)	Digit 2	Digit 3 (LSD)	m + 1
Input 2		S			Digit 1	Digit 2	Digit 3	m + 2
Input 3		S			Digit 1	Digit 2	Digit 3	m + 3
Input N		S			Digit 1 (MSD)	Digit 2	Digit 3 (LSD)	m + N
Result	E	S	D	ER	Digit 1 (MSD)	Digit 2	Digit 3	Result Address n
					Digit 4	Digit 5	Digit 6 (LSD)	n + 1

E = Enable Bit (1 = Function in Progress)
S = Sign Bit (1= Negative)
D = Done Bit (1 = Function Complete)
ER = Error Bit (Set if number of readings = 0)
MSD = Most Significant Digit
LSD = Least Significant Digit
N = Number of Readings Being Averaged

11585

3. Enter a data address and store in that address the number of values to average. Then enter an address for the result.

Let's choose a data address of 200 and the number of values to average of 003. Also, let's choose 305 as the address for the result. Entering these values results in the data table format shown in Figure 3.23. Once you enter the data address, the addresses of the remaining input words are selected automatically by the AF1.

Figure 3.23
AF1 Average Function Format After Address Entry

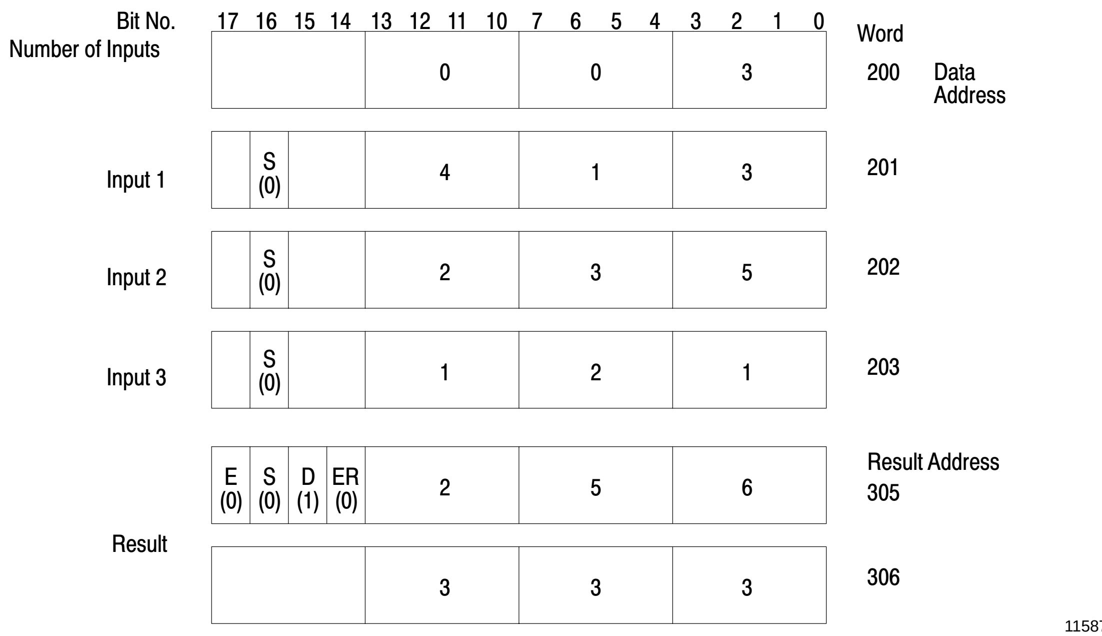
Bit No.	17	16	15	14	13 12 11 10	7 6 5 4	3 2 1 0	
Number of Inputs					0	0	3	Data Address 200
Input 1		S			Digit 1 (MSD)	Digit 2	Digit 3 (LSD)	201
Input 2		S			Digit 1	Digit 2	Digit 3	202
Input 3		S			Digit 1	Digit 2	Digit 3	203
Result	E	S	D	ER	Digit 1 (MSD)	Digit 2	Digit 3	Result Address 305
					Digit 4	Digit 5	Digit 6 (LSD)	306

E = Enable Bit (1 = Function in Progress)
S = Sign Bit (1 = Negative)
D = Done Bit (1 = Function Complete)
ER = Error Bit (Set if number of readings = 0)
MSD = Most Significant Digit
LSD = Least Significant Digit

11586

4. Enter the values of the numbers you wish to average in the input word addresses. You can enter these numbers from the keyboard of your industrial terminal or through ladder diagram functions. if we choose to enter values of +413, +235, and +121 into input word addresses 201, 202, and 203 respectively, we obtain the result shown in Figure 3.24 when the average function executes. The result appears in result words 305 and 306 as 256.333.

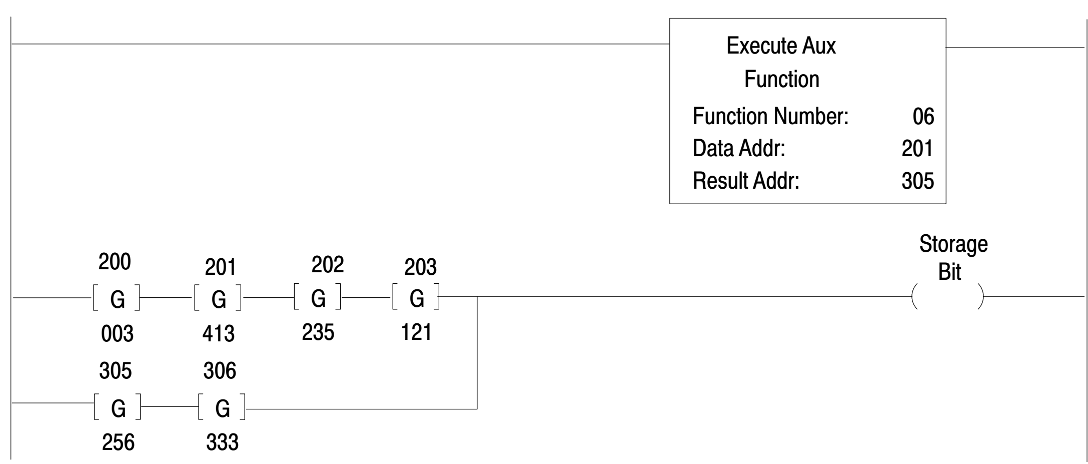
Figure 3.24
AF1 Average Function Format After Execution



Entry and Display of Input and Result Values

Figure 3.25 shows one method you can use to enter values and display results of an AF1 average function. although there are several techniques for accomplishing this, we chose get instructions. The first rung requests an AF1 average function. The top branch of the second rung shows the number of values to be averaged, 003, in word 200. It also shows the numbers to be averaged in words 201, 202, and 203. The lower branch of the second rung shows the resultant average as 256.333 in words 305 and 306.

Figure 3.25
AF1 Average Function Input and Result Display Rungs



Error Message

If you insert a zero for the number of values to be averaged, the error bit (bit 14) is set and the result reads zero.

AF1 Standard Deviation Function

The AF1 standard deviation function determines the standard deviation of 3-digit BCD numbers giving a 6-digit result with an implied decimal point after the third digit. The maximum number of values you can handle is 999 or is limited by the data table area available.

Standard deviation function uses the formula:

$$\left\{ \frac{\sum (x_i - x_{avg})^2}{n} \right\}^{1/2} = xxx.xxx$$

Where:

X_i = one of a group of N numbers whose standard deviation is being calculated

X_{avg} = the arithmetic average of N numbers

How to Enter an AF1 Standard Deviation Function

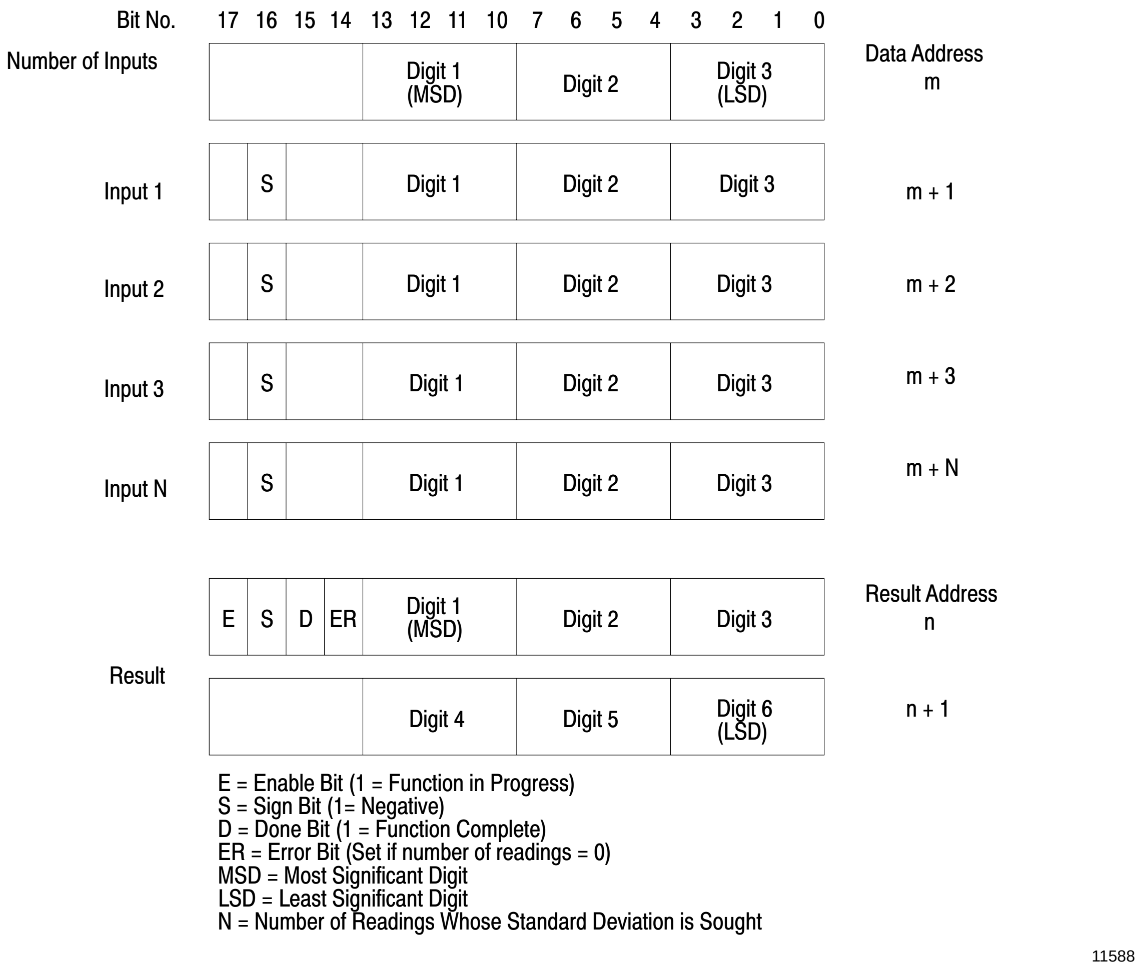
To program an AF1 standard deviation function, perform the following steps:

1. Press [SHIFT] [EAF] or [SHIFT] [SCT] on the keyboard of your industrial terminal. Figure 3.1 appears on the CRT.

2. Enter 07, the function number for AF1 standard deviation.

This entry identifies that the function entered is to perform an AF1 standard deviation and that the processor use the data table format shown in Figure 3.26 when executed. The three digits showing the number of inputs and the three digits of each input value are represented in BCD by groups of bits labeled digit 1 through 3. The six digits of the result are labeled digit 1 through 6.

Figure 3.26
General AF1 Standard Deviation Function Word and Digit Format



3. Enter a data address and store in that address the number of values whose standard deviation you wish to determine.

Let's choose a data address of 200 and the number of values whose standard deviation we want of 003. Also, let's choose 305 as our result address. Entering these values results in the data table format shown in Figure 3.27. once you enter the data address, the addresses of the remaining input words are selected automatically by the AF1.

Figure 3.27
General AF1 Deviation Function Format After Address Entry

Bit No.	17	16	15	14	13 12 11 10	7 6 5 4	3 2 1 0	
Number of Inputs					0	0	3	Data Address 200
Input 1		S			Digit 1 (MSD)	Digit 2	Digit 3 (LSD)	201
Input 2		S			Digit 1	Digit 2	Digit 3	202
Input 3		S			Digit 1	Digit 2	Digit 3	203
Result	E	S	D	ER	Digit 1 (MSD)	Digit 2	Digit 3	Result Address 305
					Digit 4	Digit 5	Digit 6 (LSD)	306

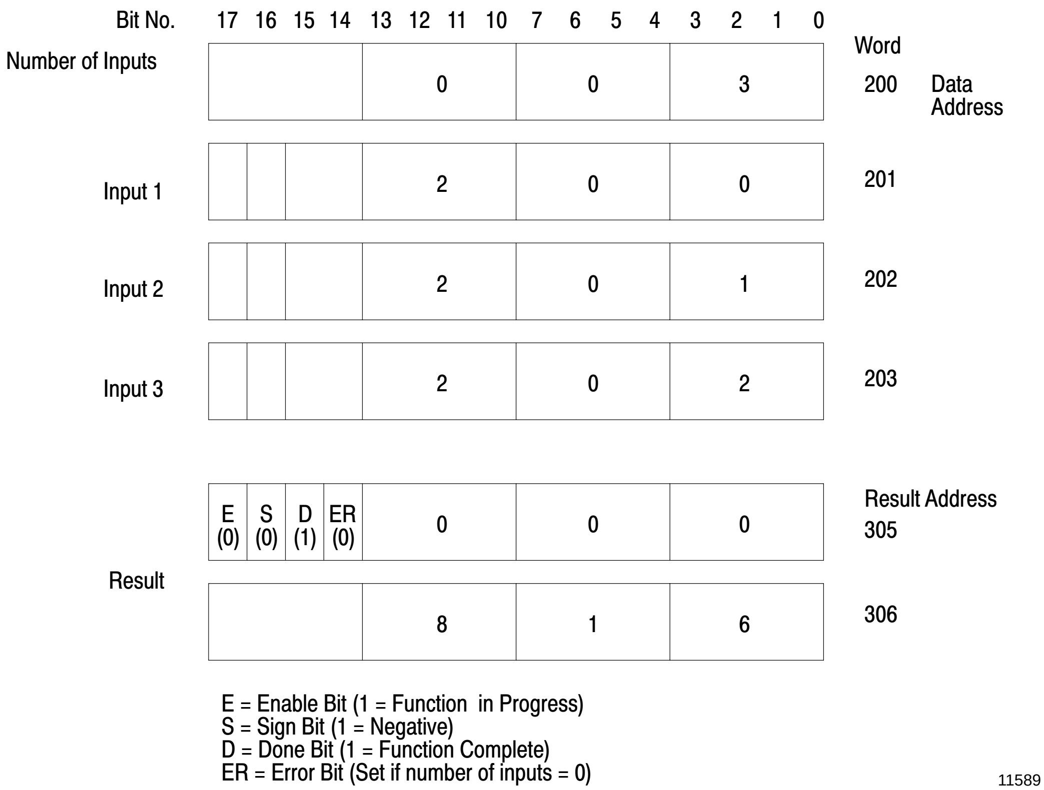
E = Enable Bit (1 = Function in Progress)
S = Sign Bit (1 = Negative)
D = Done Bit (1 = Function Complete)
ER = Error Bit (Set if number of readings = 0)
MSD = Most Significant Digit
LSD = Least Significant Digit

11586

4. Enter the values of the numbers whose standard deviation is desired in the input word addresses.

You can enter these numbers from the keyboard of your industrial terminal or through ladder diagram functions. If we choose to enter values of 200, 201, and 202 into input word addresses 201, 202, and 203 respectively, we obtain the result shown in Figure 3.28 when the standard deviation function executes. The result appears in words 305 and 306 as 000.816. The result is truncated but is accurate to $\pm$0.001.

Figure 3.28
AF1 Standard Deviation Function Format After Execution



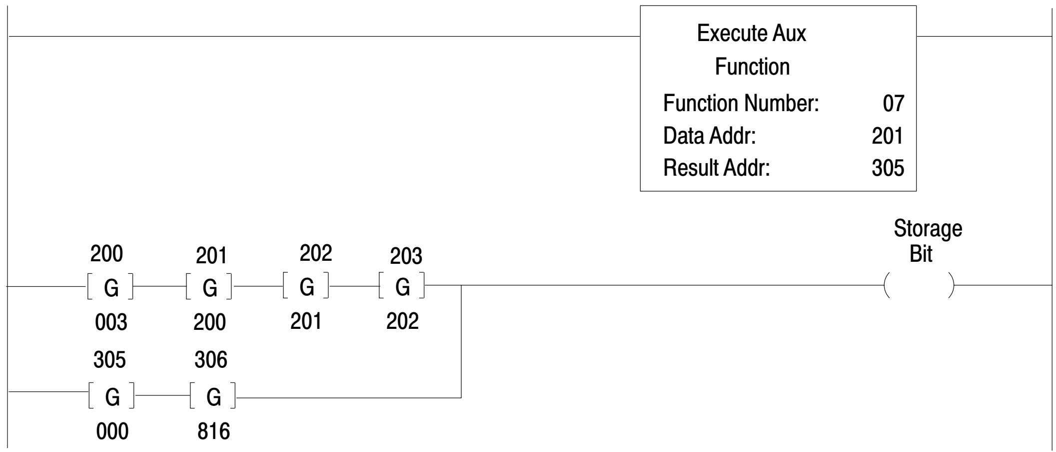
Entry and Display of Input and Result Values

Figure 3.29 shows one method you can use to enter values and display results of an AF1 standard deviation function. Although there are several techniques for accomplishing this, we chose the instructions. The first rung requests an AF1 standard deviation function. The top branch of the second rung shows the number of values whose standard deviation is sought, 003, in word 200. it also shows the three numbers involved in words 201, 202, and 203. The lower branch of the second rung shows the resulting standard deviation as 000.816 in words 305 and 306.

Error Message

If you insert a zero for the number of values whose standard deviation is sought, the error bit (bit 14) is set and the result reads zero.

Figure 3.29
AF1 Standard Deviation Function Input and Result Display Rungs



AF1 BCD to Binary Conversion Function

The AF1 BCD (binary coded decimal) to binary conversion function converts a BCD number (from 0 to 4095) into a 12-bit binary number.

How to Enter an AF1 BCD to Binary Conversion Function

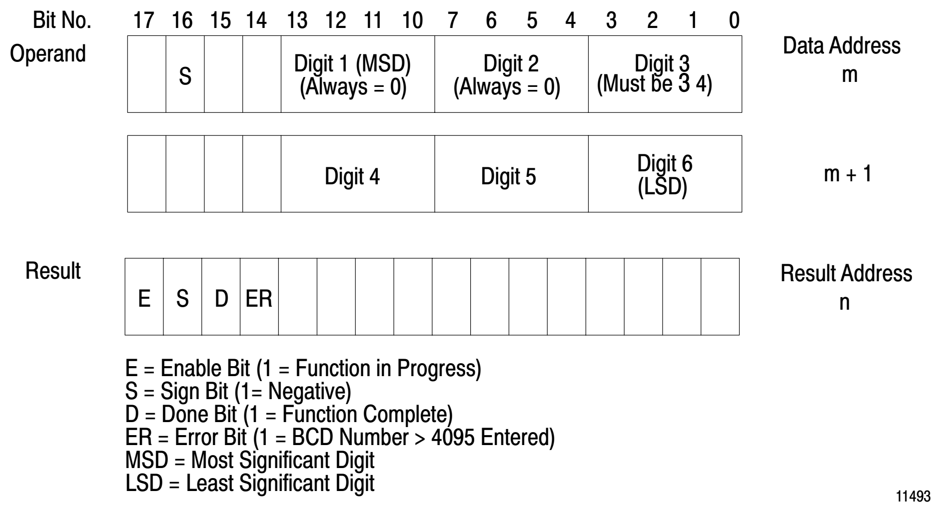
To program an AF1 BCD to binary conversion function, perform the following steps:

1. Press [SHIFT] [EAF] or [SHIFT] [SCT] on the keyboard of your industrial terminal. Figure 3.1 appears on the CRT.

2. Enter 13, the function number for AF1 BCD to binary conversion.

This entry identifies that the function entered is to perform an AF1 BCD to binary conversion and that the processor use the data table format shown in Figure 3.30 when executed.

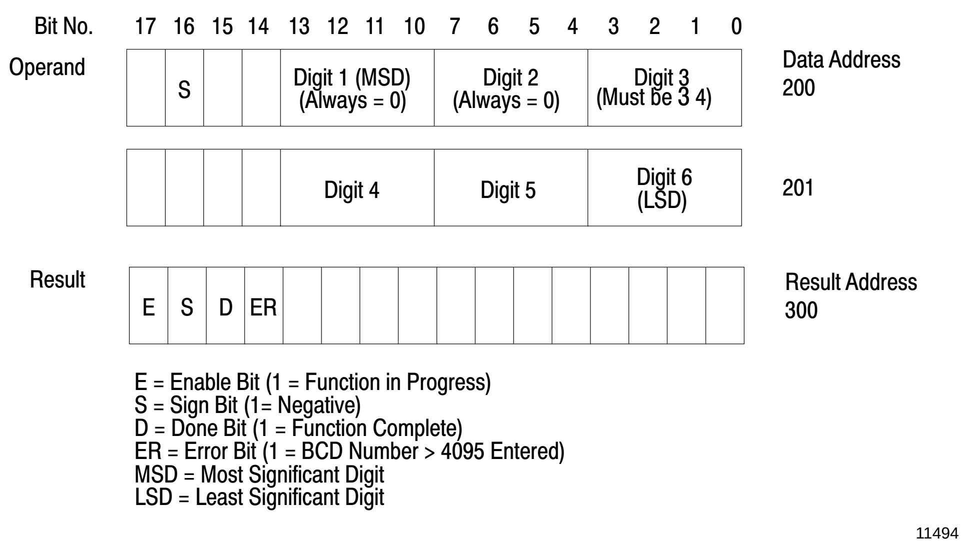
Figure 3.30
General AF1 BCD to Binary Conversion Function Word and Digit Format



3. Enter a data address.

If we choose a data address of 200 and a result address of 300, the data table format is as shown in Figure 3.31. The most significant three digits of the operand (the BCD number we want to convert to binary) reside in the data address word 200 and the least significant three digits reside in the next higher address, 201. The first two digits are always zero and the third digit must not exceed four. The number, converted to binary format, is stored in bits 0 through 13 in the result address, word 300.

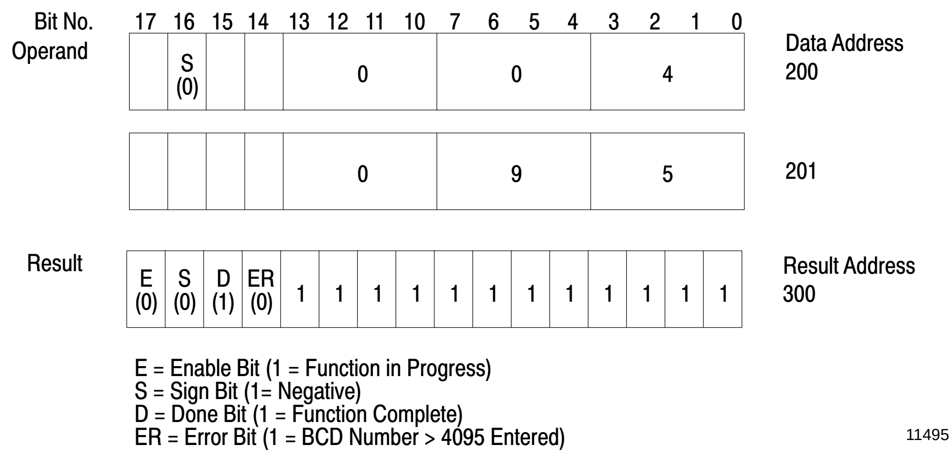
Figure 3.31
AF1 BCD to Binary Conversion Function Format After Address Entry



4. Enter the operand.

You can enter the operand from the keyboard of your industrial terminal or through ladder diagram functions. If we choose to enter 4095, the largest BCD number that we can convert to a 12 bit binary number, we obtain the data tale configuration shown in Figure 3.32.

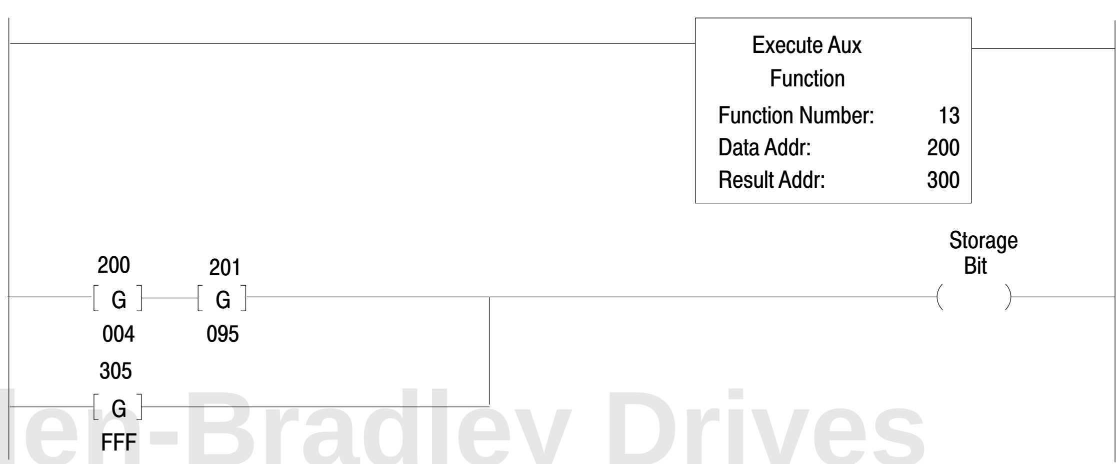
Figure 3.32
AF1 BCD to Binary Conversion Function Format After Execution



Entry and Display of Input and Result Values

Figure 3.33 shows one method for inserting input values and displaying inputs and results of an AF1 BCD to binary conversion function. Although there are other methods for accomplishing this, we chose get instructions. The first rung requests an AF1 BCD to binary conversion. The top branch of the second rung shows the BCD number that is to be converted (004095) in words 200 and 201. The bottom branch shows in the hexadecimal notation FFF (bits 0 through 13 in word 300 are set as shown in Figure 3.32).

Figure 3.33
AF1 BCD to Binary Conversion Function Input and Display Rungs



Error Message

If you enter a BCD number larger than 4095, the error bit (bit 14) is set and the result reads zero.

AF1 Binary to BCD Conversion Function

The AF1 binary to BCD conversion function converts a 12-bit binary number to a BCD number (from 0 to 4095).

How to Enter an AF1 Binary to BCD Conversion Function

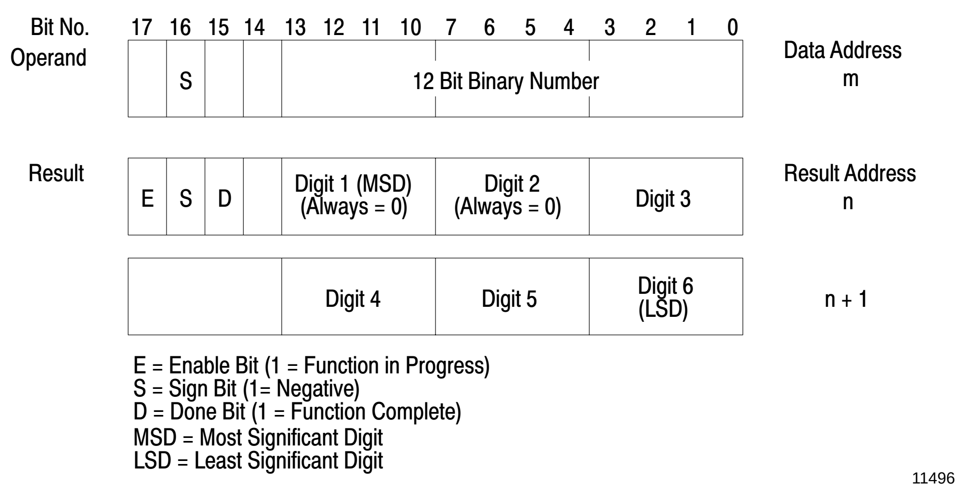
To program an AF1 binary to BCD conversion function, perform the following steps:

1. Press [SHIFT] [EAF] or [SHIFT] [SCT] on the keyboard of your industrial terminal. Figure 3.1 appears on the CRT.

2. Enter 14, the function number for the AF1 binary to BCD conversion function.

This entry identifies that the function entered is to perform an AF1 binary to BCD conversion and that the processor use the data table format shown in Figure 3.34 when executed.

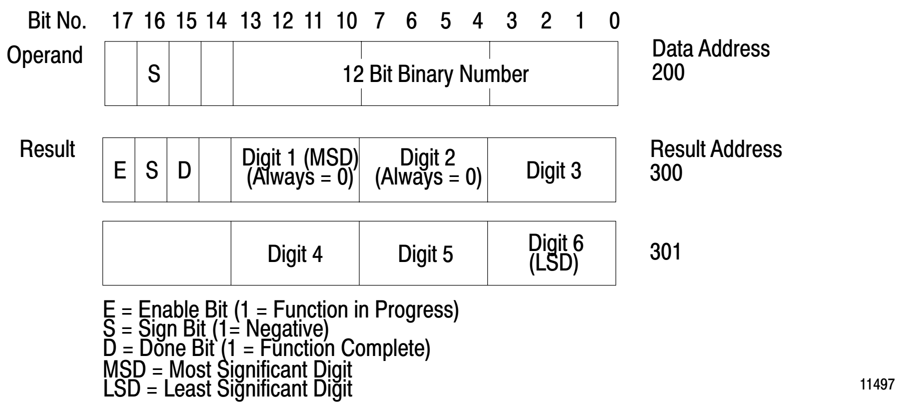
Figure 3.34
General AF1 Binary to BCD Conversion Function Word and Digit Format



3. Enter a data address and a result address.

If we choose a data address of 200 and a result address of 300, the data table format is as shown in Figure 3.35. Bits 0 through 13 of word 200 are reserved for the operand (the 12-bit binary number we want to convert to BCD). The result address, 300, contains the most significant three digits of the resulting BCD number. The least significant three digits reside in the next higher address, 301. The first two digits of the BCD number are always zero and the third digit can not exceed four.

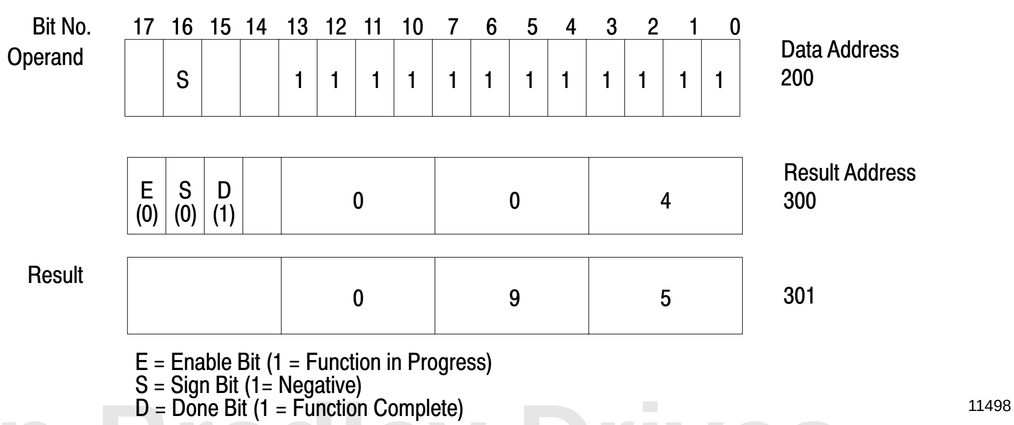
Figure 3.35
AF1 Binary to BCD Conversion Function Format After Address Entry



4. Enter the operand.

You can enter the operand from the keyboard of your industrial terminal or through ladder diagram functions. If we choose to set bits 0 through 13 in word 200, that is, insert the largest possible binary number in 12 bits, we obtain 4095 for the corresponding BCD number (Figure 3.36). The ones in bits 0 through 13 or word 200 indicate that each bit is set.

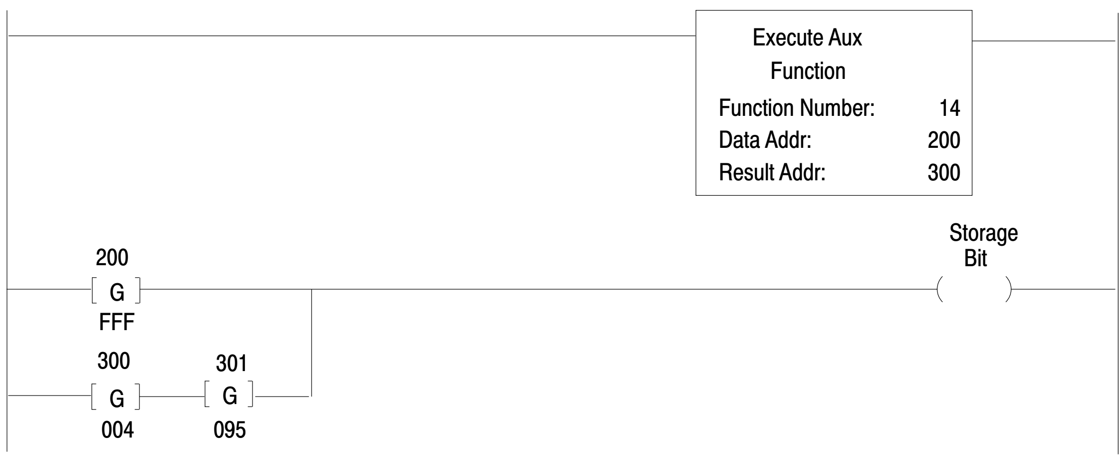
Figure 3.36
AF1 Binary to BCD Conversion Function Format After Execution



Entry and Display of Input and Result Values

Figure 3.37 shows one method for inserting input values and displaying inputs and results of an AF1 binary to BCD conversion function. Although there are other methods for accomplishing this, we chose get instructions. The first rung requests an AF1 binary to BCD conversion function. The top branch of the second rung shows the binary number (in the hexadecimal notation FFF) that we want converted to BCD. In this example, the binary number is the largest possible, with bits 0 through 13 or word 200 set as shown in figure 3.36. The lower branch shows the resulting BCD number, 004095, in words 300 and 301.

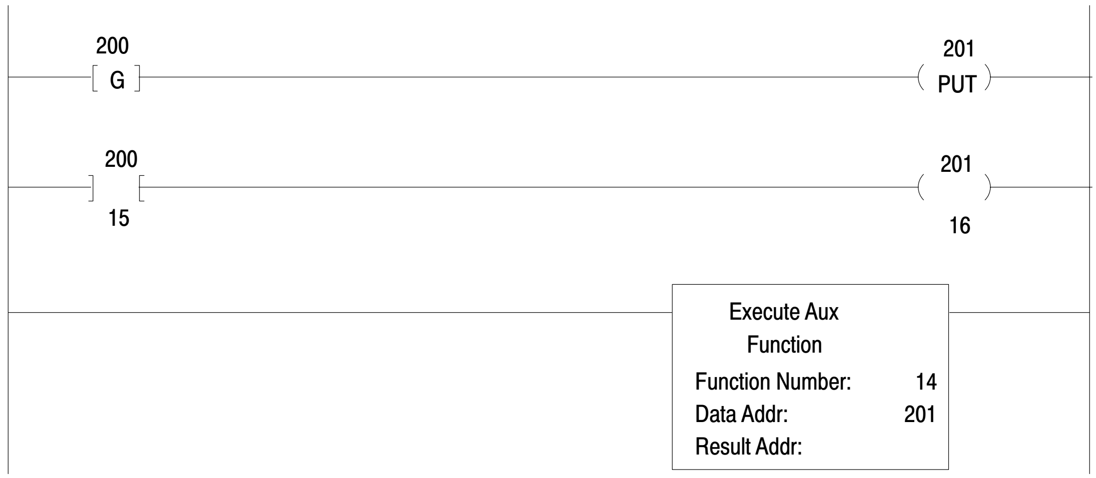
Figure 3.37
AF1 Binary to BCD Conversion Function Input and Result Display Rungs



Sign Bits

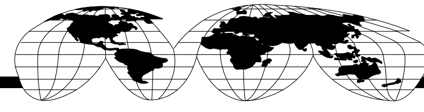
If you encounter any binary data where the sign bit is not in bit 16, you must move the sign bit into bit 16 of an auxiliary data table word prior to doing a binary to BCD conversion. If, for example, the sign bit of your module is bit 15 in word 200, the rungs in Figure 3.38 permit you to make a binary to BCD conversion. The first rung puts word 200 data into word 201. Rung two sets bit 16 in word 201 if bit 15 in word 200 is set. The AF1 binary to BCD conversion function in rung three then uses word 201 as its data address.

Figure 3.38
Transfer of Sign Bit



ALLEN-BRADLEY
A ROCKWELL INTERNATIONAL COMPANY

Allen-Bradley has been helping its customers improve productivity and quality for 90 years. A-B designs, manufactures and supports a broad range of control and automation products worldwide. They include logic processors, power and motion control devices, man-machine interfaces and sensors. Allen-Bradley is a subsidiary of Rockwell International, one of the world's leading technology companies.



With major offices worldwide.

Algeria • Argentina • Australia • Austria • Bahrain • Belgium • Brazil • Bulgaria • Canada • Chile • China, PRC • Colombia • Costa Rica • Croatia • Cyprus • Czech Republic • Denmark • Ecuador • Egypt • El Salvador • Finland • France • Germany • Greece • Guatemala • Honduras • Hong Kong • Hungary • Iceland • India • Indonesia • Israel • Italy • Jamaica • Japan • Jordan • Korea • Kuwait • Lebanon • Malaysia • Mexico • New Zealand • Norway • Oman • Pakistan • Peru • Philippines • Poland • Portugal • Puerto Rico • Qatar • Romania • Russia–CIS • Saudi Arabia • Singapore • Slovakia • Slovenia • South Africa, Republic • Spain • Switzerland • Taiwan • Thailand • The Netherlands • Turkey • United Arab Emirates • United Kingdom • United States • Uruguay • Venezuela • Yugoslavia

World Headquarters, Allen-Bradley, 1201 South Second Street, Milwaukee, WI 53204 USA, Tel: (1) 414 382-2000 Fax: (1) 414 382-4444

Publication 1772-6.5.1 – March 1984

P/N 955094–44