****

Written by:

Toshiba International Corporation
Industrial Division
13131 W. Little York Road
Houston, TX 77041

Toshiba Corporation
1-1, Shibaura 1-Chome,
Minato-ku
Tokyo 105, Japan

Phone: 800-231-1412, 713-466-0277
Fax: 713-466-8773

Feel free to call the PLC Marketing department if you have any questions about this document, or need additional information about other Toshiba programmable controllers.

****

IMPORTANT INFORMATION

Misuse of this equipment can result in property damage or human injury. Because control system applications vary widely, you should satisfy yourself as to the acceptability of this equipment for your intended purpose. In no event will Toshiba Corporation be responsible or liable for either indirect or consequential damage or injury that may result from the use of this equipment. No patent liability is assumed by Toshiba Corporation with respect to the use of information, illustrations, circuits, equipment, or examples of applications in this publication.

Toshiba Corporation reserves the right to make changes and improvements to this publication and/or related products at any time without notice. No obligation shall be incurred, except as noted in this publication.

****

This manual is almost identical to the M20/40 and EX100 Users Manuals. It has been specially adapted for presentation in the text-based format that you see here. Additional explanations have been made in areas found difficult to understand in the printed manuals. Also, some of the wiring diagrams from the printed manuals have been abbreviated, due to the difficulty of translating graphics into ASCII text.

Topic numbers (example: 1.7 Instructions) are the same as topic numbers in the M20/40 Users Manual.

For referencing this, we suggest using MSDOS Edit, or a word processing program with a searching routine. You can then search for the topic number that you need to see. For graphical word processing programs such as MS Word for Windows, it is recommended that this manual be imported using a text (fixed size) font such as Roman 10cpi. This should preserve the ASCII formatting.

****

This manual has been prepared for first-time users of the M20/M40/EX100 (hereafter called "M/EX100") Programmable Controllers to enable a full understanding of the configuration of the equipment, and to enable the user to obtain the maximum benefits of the equipment.

Sections 1 and 2 outline the M/EX100 configuration. To fully understand the M/EX100, it is important to read these chapters carefully. Sections 3 and 4 describe the hardware used in designing external circuits and panels. Sections 5 to 9 are mainly concerned with software. Sections 10 and 11 describe the maintenance procedure for the M/EX100, to ensure safe operation and long service life.

****

Inside each section
The contents of this manual are as follows:

Section 1 Introduction
Introduces the features of the M/EX100, the names of its components, and describes handling precautions.

Section 2 System configuration
Describes the M/EX100 input and output configuration, and the equipment that constitutes the M/EX100.

Section 3 Specifications
Contains the external dimensions of the M/EX100 and input and output specifications.

Section 4 Installation and Wiring
Describes installation procedures and the wiring method.

Section 5 Operating the M/EX100
Describes the configuration of the internal memory and the method of operating the M/EX100.

Section 6 Input and Output Allocation
Explains the assignment of the input and output numbers.

Section 7 Instructions
Describes the procedures for starting the M/EX100 in detail.

Section 8 Basic Programming Procedures
Describes the procedures for starting the M/EX100, for executing a program, and other operating procedures.

Section 9 Special Functions
Describes the unique special functions of the M/EX100 and their use.

Section 10 Maintenance
Describes the precautions and maintenance procedures for ensuring reliable operation of the M/EX100.

Section 11 Troubleshooting
Lists the causes of typical problems and the items of the M/EX100's diagnostic check.

Appendices
List the execution time of each instruction, module part numbers, and I/O wiring diagrams.

Note and Caution symbols
Users of this manual should pay special attention to information preceded by the following symbols.

Note Calls the reader's attention to information considered important for full understanding of programming procedures and/or operation of the equipment.

Caution Calls the reader's attention to conditions or practices that could damage the equipment or render it temporarily inoperative.

Related manuals
The following related manuals are available.

Handy Programmer (HP) Operation Manual UM-EX25UB-E022
EX100 Computer Link User's Manual UM-EX100-E004
EX100 Position Control Module User's Manual
EX100 ASCII/Basic Module Operation Manual UM-EX100-E016
M20/40 User's Manual UM-EXM2040-E002
EX100 User's Manual UM-EX100-E001
PLC Primer by Country Squire: Call 903-364-2365

Terminology
The following is a list of abbreviations and acronyms used in the manual.

AWG American Wire Gage
ASCII American Standard Code for Information Interchange.
CPU Central Processing Unit
EEPROM Electrically Erasable Programmable Read Only Memory
EPROM Erasable Programmable Read Only Memory
H hexadecimal (when it appears in front of an alphanumeric string)
I/O Input/Output
LCD Liquid Crystal Display
LED Light Emitting Diode
M/EX100 M20 and M40 and EX100
ms millisecond
NEMA National Electrical Manufactures' Association
PC Programmable Controller
PROM Programmable Read Only Memory
RAM Random Access Memory
ROM Read Only Memory
uS Microsecond
Vac ac voltage
Vdc dc voltage

****

INDEX

5.4 Scanning

****

1. Introduction

1.1

Introducing the M20/40, and the EX100 The M20/40 is a compact, "shoe-box" type, high-performance programmable controller with a range of 20 to 168 input and output points.

The EX100 is a compact, modular type, high-performance controller with up to 480 I/O (depending on I/O type).

I/O points -- The M20 has 12 input and 8 output points. The M40 has 24 input and 16 output points. For I/O points expansion, two types of expansion units are available: 12 inputs/8 outputs and 24 inputs/16 outputs. The EX100's I/O modules can also be used with the M20/40 by connecting an expansion rack: 2-slot rack or 4-slot rack. One additional expansion unit or expansion rack can be connected to the M20/40. If the 4-slot expansion rack with four 32 point modules is connected to the M40, the M40 can control up to 168 points.

I/O points -- The EX100 is a modular PLC. Four types of racks are available: Expandable: 6 slot or 9 slot
Non-expandable: 6 slot or 9 slot
The maximum configuration uses two 9 slot racks, one CPU, two power supplies and 15 I/O modules. The two racks are connected by a cable.

Memory capacity -- Program memory capacity can be selected in either 4K steps or 3K steps by switch settings. If the 3K-step mode is selected, 1K words of data are stored in EEPROM. (The program is also stored in EEPROM).

Control functions -- In addition to the basic relay ladder functions, the M/EX100 provide functions such as data operations, arithmetic operations, various functions, etc. Furthermore, their analog control functions, positioning functions and data communication functions allow their application to a wide scope of control systems.

Series compatibility -- Programs from the EX100, M20/40, EX200B, EX250 and EX500 are compatible. Peripheral equipment can also be shared.

The M20 and M40 have all of the features of the EX100 with the enhanced CPU. This includes the real-time clock calendar, RS485 connection, 4K of memory, and the same instruction set. They also use the same communication protocol on both the programming and computer link (RS485) ports, so any software package written for one will work with the other. The difference between the M20/40 and EX100 is the number of I/O. The M20/40 are "shoebox" PLCs, while the EX100 is used for larger systems.

M/EX100 features Battery-free - The M/EX100 CPU has a standard built-in EEPROM, permitting operation without need of a battery. If the memory setting is 3K mode, the variable data can be written and/or read from the EEPROM, providing completely maintenance-free back-up operation.

The ability to write data to the EEPROM while the programming is in RUN is a powerful feature of these PLCs. This means that a battery is not needed even if data must be maintained during a power failure. EEPROM storage of data can be performed by the ladder logic automatically. This is an important feature for OEMs who ship their equipment with Toshiba programmable controllers in it, because it can eliminate the need for changing the battery every few years. The cost of the battery is also eliminated. See section 9.5 for an example circuit illustrating this.

High speed execution -- Quick response owing to high speed program execution of 0.9 uS per contact instruction and immediate I/O update instructions.

On-line program changing -- On-line (during RUN) program changes are possible, providing efficient program debugging simulation.

Clock-calendar -- The M20/40, and EX100 with enhanced CPU have the clock- calendar function (year, month, day, day of the week, hour, minute, second), which is a powerful tool for performing scheduled operations and batch processing. The built-in capacitor can maintain the clock- calendar for a week at 25 degC, and the optional battery can maintain it for the life of the battery.

Efficient data link network -- The M20/40 and EX100 with enhanced CPU can communicate with a computer via the RS-485 interface, and with other Toshiba PLCs and remote I/O stations via TOSLINE-30, allowing use of an efficient network environment.

Write-protect function -- The operation control switch on the M/EX100 can set to the protect (RUN-P) position. In this position, no program changes can be made.

Program protect feature -- By setting the last three digits of the program ID to "FFF", the program can be protected from unintended changes. Of course, after changing the ID back, it can then be modified.

Built-in 24 Vdc power supply -- The M/EX100 are equipped with a 24 Vdc power supply for external devices as well as for input signals.

Although the M/EX100 are designed to withstand severe environmental conditions, it is necessary to observe the following installation, wiring, and storage precautions.

Handling and operation Because the M/EX100 has an EEPROM as standard memory, it does not require a battery. When power to the M/EX100 is turned on, the program stored in the EEPROM is transferred to the RAM for execution. Therefore, if the program in the RAM is changed, it is necessary to write the updated program into the EEPROM BEFORE turning the power off. Otherwise, the updated program will be lost, and the old program will be restored. The optional battery is useful for maintaining the clock calendar's operation when the power is shut off for more than a week.

Environment Ambient temperature and humidity during operation: 0 to 55 degC (32 to 131 deg F), 20% to 90% RH During storage: -20 to 75 degC (-4 to 167 degF), 20% to 90% RH

Because the M/EX100 are sensitive electronic equipment, it is necessary to avoid abrupt changes in temperature and humidity. In particular, avoid sudden temperature changes that will cause condensation of water vapor.

Do not install the M/EX100 in locations subject to excessive shock or vibration.

Installation and Wiring

When wiring the M/EX100 and installing it on a control panel, be sure that fragments of wire or other metal scraps do not fall into the ventilation vents on the top of the unit. Proper grounding also is very important for safety and for the M/EX100 to operate reliably. Be sure to ground the unit correctly, by referring to Section 4 of this manual. Be sure that power is turned off when connecting or disconnecting the expansion cable. Also, be sure that power is turned off when mounting or removing an I/O module.

****

2. System Configuration

2.1 System Configuration

Peripherals:	Computer communications:
Handy Programmer EX25-UHP100	RS485 - Allows 16 PLCs to communicate on two twisted pairs with a computer
Data Access Panel EX25-UDP100	
Third party products List available on request	Programming port - Allows 1 PLC to communicate with a computer

PLC to PLC communications:
Using the Tosline 30 modules, up to 32 EX100's and M20/40's can exchange data using 32 Z registers as common memory. Tosline 30 is available using wire or fiber optic communications.

2.2 I/O expansion

The M20 has 12 input and 8 output points. The M40 has 24 input and 16 output points. I/O points can be expanded by connecting an expansion unit or an expansion rack.

The following combinations are available.

Model	I/O points	Base Unit	Expansion Unit
M20	20 (12/8)	M20	None
	40 (24/16)	M20	20 point exp
	84	M20	Rack 2-slot
	148	M20	Rack 4-slot
M40	40 (24/16)	M40	None
	60 (36/24)	M40	20 point exp
	80 (48/32)	M40	40 point exp
	104	M40	Rack 2-slot
	168	M40	Rack 4-slot

NOTE

(1) 20 point exp and 40 point exp respectively indicate a 20 point (12 inputs/8 outputs) expansion unit and a 40 point (24 inputs/16 outputs) expansion unit.

(2) Rack 2-slot and Rack 4-slot respectively indicate a 2-slot expansion rack and a 4 slot expansion rack.

(3) I/O points of the combinations with an expansion rack show the maximum points using 32 point I/O modules.

2.3a System components for M20/40

2.3.1a The basic M20/40 unit is available in two models: the M20 and the M40. And each model is available in three types, depending on the input type and the power supply voltage.

Model	Part No.	Input	Output	Power supply
M20	EX10M20DR5	12 pts, 24 Vdc	8 pts, Relay	100-240 Vac
	EX10M20DR6	12 pts, 24 Vdc	8 pts, Relay	24 Vdc
	EX10M20AR5	12 pts, 120 Vac	8 pts, Relay	100-240 Vac
M40	EX10M40DR5	24 pts, 24 Vdc	16 pts, Relay	100-240 Vac
	EX10M40DR6	24 pts, 24 Vdc	16 pts, Relay	24 Vdc
	EX10M40AR5	24 pts, 120 Vac	16 pts, Relay	100-240 Vac

Connections(M20/40): Please refer to a picture of the M20/40 as shown in the sales flyer or sales brochure.

1. Power supply terminals
   connect the power cable, see section 4.6

2. Ground terminal
   use grounding, see section 4.5

3. 24 Vdc output terminals
   supply 24 Vdc +/- 10% (0.4A max.) Allowable maximum output current is limited depending on the unit configuration and type of I/O modules. See section 2.6

4. Input terminals
   Connect input signal wires. See section 3.4.1 and 3.4.2.

5. Output terminals
   Connect output signal wires. See sections 3.4.1 and 4.3.2.

6. Computer link terminals
   Used for data linkage with a computer. For details of the computer link function, refer to the separate EX100 Computer Link User's Manual (UM-EX100-E004).

7. Programmer port
   Connects the programmer.

RUN CPU I/O 8. Status LEDs
Displays the status of the M20/40.

(Power)	Lit	Internal 5 Vdc normal
	Not Lit	Internal 5 Vdc abnormal

RUN	Lit	In operation (the RUN mode)
	Blinking	In the HOLD state
	Not Lit	In the HALT mode or ERROR mode

CPU	Lit	CPU normal
	Blinking	Program abnormal
	Not Lit	CPU abnormal
I/O	Lit	I/O normal
	Not Lit	I/O abnormal

9. Operation control switch
Controls the operation modes of the M/EX100.

HALT Program execution stopped. This is the position during normal programming.

RUN Program execution, HALT or RUN can be selected from the programmer. On-line (during RUN) program changes are possible. Writing the ladder logic program to EEPROM is not permitted in RUN mode (Although data can be written to EEPROM while in RUN - see section 9.5).

RUN-P Program execution. HALT or RUN can be selected from the programmer. Programming and EEPROM writing are inhibited.

For details of the operation control switch and operation mode, see section 5.3.

10. Computer link setting switches.

For EX100: Rotary switch: station number setting (0 to 15)

DIP switch settings for EX100 - see section 2.3b.3

DIP switch (For M20/40)

No.	Name	
1	BR2	Baud rate 9600 4800 2400 1200
		BR2 OFF OFF ON ON
2	BR1	BR1 OFF ON OFF ON
3	PEN	Parity setting
		Parity None Odd Even
4	PR	PEN OFF OFF ON ON
		PR OFF ON OFF ON
5	PRG/LINK	
6	3K/4K	Programmer/computer link selection (exclusive)
7	Not Used	OFF Programmer can be used
		ON Computer link can be used
8	Not Used	

Memory setting (see Section 5.2)
OFF 3K-step mode
ON 4K-step mode

11 Input status LEDs
Indicates the ON status of each input signal.

12 Output status LEDs
Indicates the ON status of each output signal.

13 Expansion connector
Connects expansion cable to the expansion unit or expansion rack.

2.3a.2 Expansion units (M20/40)
The following two types of expansion units are available.

Part No.	Input	Output	Power Supply
EX10E20DR	12 pts. 24 Vdc	8 pts. Relay	Supplied from basic unit
EX10E40DR	24 pts. 24 Vdc	16 pts. Relay	

NOTE A 30 cm expansion cable is supplied with the expansion unit.

1. Ground terminal
Use grounding, see section 4.5.

2. Input terminals
Connect input signal wires, see sections 3.4.3 and 3.4.4.

3. Output terminals
Connect output signal wires, see sections 3.4.3 and 3.4.4.

4. Input status LEDs
Indicates the ON status of each input signal.

5. Output status LEDs
Indicates the ON status of each output signal.

6. Expansion connector
Connects expansion cable from the basic unit. Either side expansion connector can be used according to unit installing layout.

2.3a.3 Expansion rack
The following two types of expansion racks are available. By using the expansion rack, the EX100's I/O modules can be used for the M20/40.

Part No.	Slot	Power supply
EX10EUB2	2 slots for I/O modules	Supplied from basic unit
EX10EUB4	4 slots for I/O modules	Supplied from basic unit

NOTE A 10 cm expansion cable is supplied with the expansion racks.

2.3a.4 Expansion cable for the M20/40

The expansion cable is used to connect the basic unit and the expansion unit or expansion rack. It is available in the following lengths.

Part No.	Length
EX10PEXPC1	10 cm
EX10PEXPC3	30 cm

2.3b Input and output configuration of the EX100

The rack of the EX100 is available in two sizes. The smaller one can accommodate a total of six modules, including the power supply and CPU, and the larger one can accommodate a total of nine modules. Furthermore, there are two types of each rack, the expandable and the nonexpandable type, making a total of four kinds. The expandable rack is equipped with an expansion connector on the left side. In the expansion configuration, both racks must be of the expandable type.

The minimum configuration is one six slot non-expandable rack with a P/S, CPU, and up to four I/O modules.

The maximum configuration is two nine slot expandable racks connected together. Two power supplies and one CPU are needed, leaving space for a total of 15 I/O modules.

2.3b.1 Rack for EX100

EX10-UBA1	6 slots	non-expandable
EX10-UBA2	9 slots	non-expandable
EX10-UBB1	6 slots	expandable
EX10-UBB2	9 slots	expandable

2.3b.2 Power supply for EX100

The power supply is mounted in the slot at the extreme left of the rack. The following three types of power supply modules are available, depending on the voltage required.

EX10-MPS51	100 to 120 Vac (+10 / -15%)
EX10-MPS61	200 to 240 Vac (+10 / -15%)
EX10-MPS31	24 Vdc (+20 / -15%)

Output rating
2.5 A max at 5 V for internal control
0.5 A max at 24 Vdc for powering external devices
Total for internal and external power supply: 15 W or less

External 24 Vdc output terminals
These terminals supply 24 Vdc (+- 10%, 0.5A max) to external devices, such as sensors, as well as to some of the input/output modules for power. See section 2.6 for power calculations.

RUN signal output terminals
Built-in NO contact that turns on (contact closes) when the EX100 is in the RUN state. It is rated for 240 Vac or 24 Vdc at 2 A maximum.

Also see section 4.5 for Installation and Wiring instructions.

2.3b.3 CPU module

One CPU module is mounted on the basic unit of the EX100. This module must be mounted in the slot next to the power supply module. Two types of CPU modules are available: the enhanced and the standard.

EX10-MPU11A Standard
EX10-MPU12A Enhanced

The enhanced CPU has the following additional features which the standard CPU does not have:
Built in RS485 connection. This allows up to 16 PLC's to communicate with a computer over two twisted pair.
Real time clock calendar which keeps track of hour, minute, second, day of the month, day of the week, month, and year.

CPU LED indicators and the RUN/HALT/RUNP modes:

RUN CPU I/O Status LEDs
Displays the status of the EX100.

(Power)	Lit	Internal 5 Vdc normal
	Not Lit	Internal 5 Vdc abnormal
RUN	Lit	In operation (the RUN mode)
	Blinking	In the HOLD state
	Not Lit	In the HALT mode or ERROR mode
CPU	Lit	CPU normal
	Blinking	Program abnormal
	Not Lit	CPU abnormal
I/O	Lit	I/O normal
	Not Lit	I/O abnormal

Operation control switch
Controls the operation modes of the EX100.

HALT Program execution stopped. This is the position during normal programming.

RUN Program execution, HALT or RUN can be selected from the programmer. On-line (during RUN) program changes are possible.

RUN-P Program execution. HALT or RUN can be selected from the programmer. Programming and EEPROM writing are inhibited.

For details of the operation control switch and operation mode, see section 5.3.

Computer link setting switches.

Rotary switch - station number setting (0 to 15)

Baud Rate DIP switches on enhanced EX100 CPU

BR2	BR1	Baud Rate
OFF	OFF	9600
OFF	ON	4800
ON	OFF	2400
ON	ON	1200

Parity DIP switches on EX100

PEN	PR	Parity
OFF	--	None
ON	OFF	Odd
ON	ON	Even

2.3.5 As listed below, various I/O modules are available for the M/EX100, allowing it to be used for a wide variety of applications.

Part No.	Type	Specifications
EX10-MDI31	dc/ac input	16 pts (16 pts common), 12 to 24 Vdc/Vac
EX10-MDI32	dc input	32 pts (8 pts common), 24 Vdc
EX10-MIN51	ac input	16 pts (16 pts common), 100 to 120 Vac
EX10-MIN61	ac input	16 pts (16 pts common), 200 to 240 Vac
EX10-MRO61	Relay output	12 points (4 pts common), 240 Vac, +10%/24 Vdc, +20% (Max.) 2A/point 4A / common (max.)
EX10-MRO62	Relay output	8 points (isolated), 240 Vac, +10%) 24 Vdc, +20% (max) 2A/point (max.)
EX10-MDO31	Transistor output	16 pts (16 pts / common), 5 to 24 Vdc
EX10-MDO32	Transistor output	32 points (8 pts / common), 5 to 24 Vdc
EX10-MAC61	Triac output	12 pts (4 pts common), 100 to 240 Vac. 0.5 A/point (max.), 0.6 A/SSR (max.)
EX10-MAI21	Analog input	4 channels, 1 to 5 V/4 to 20mA, 8-bit resolution
EX10-MAI22	Analog input	4 channels, 1 to 5 V/4 to 20 mA 12-bit resolution
EX10-MAI31	Analog input	4 channels, 0 to +10V, 8-bit resolution
EX10-MAI32	Analog input	4 channels, -10 to 10V. 12-bit resolution
EX10-MAO31	Analog output	2 channels, 1 to 5 V/ 4 to 20mA/ 0 to 10V 8-bit resolution
EX10-MAO22	Analog output	2 channels, 1 to 5 V/4 to 20mA 12-bit resolution
EX10-MAO32	Analog output	2 channels, -10 to +10 V, 12-bit resolution
EX10-MPI21	Pulse input	1 ch (two phase and zero marker), 5/12V 100 kps (max.), 24-bit counter
EX10-MMC11	Positioning	1 axis, 200 kpps (max.), 5 to 24 Vdc +/-999,999 pulses, 64 points data

EX10-MAS11	ASCII/BASIC	2 ports of RS-232C, BASIC-52, 32K Bytes EEPROM for user BASIC program
EX10-MLK11	TOSLINE-30 Twisted-pair	8/16/32 words data highway 187.5 kbps, 1 km max.
EX10-MLK12	TOSLINE-30 Optical fiber	8/16/32 words data highway 375 kbps, 2 km max.

NOTE For detailed specifications of the I/O modules, see section 3.4.5
Custom I/O modules are also available from third party suppliers.

2.3.6 Options

Battery

The M/EX100 has an EEPROM for storing the user program permanently. Moreover, the data in the retentive registers and the clock-calendar are backed up by a built-in capacitor (7 days at 25 degr. C). The optional battery is used when it is necessary to back up the data in the retentive registers or the clock-calendar in excess of the capacitor's back-up period.

Applicable battery for EX100
Part number (EX100): EX10-ACR2 (Toshiba)
Voltage: 3V
Capacity: 180 mAh
Recommended
replacement period: 1 year

Applicable battery for M20/40
Part number EX2040PBATT

Vacant slot cover

Optional covers are available for covering vacant slot on the rack.

Part number: EX10-ABP1

2.4 The M/EX100 supports two types of data link systems, the computer link and TOSLINE-30.

The computer link is a data transmission function between computer and the M/EX100, using the standard RS-485 interface. The data in the M20/M40, can be read and written by creating simple communication program on the computer. (Communication via the RS-232C is also possible using a conversion adapter.) Our ASCII communication protocol is published in the Computer Link Manual (UM-EX100-E004). A sample program has been written in BASIC that illustrates the information in this manual. Call your local distributor to obtain a copy of EXDATA.

Interface Conforms to RS-485

Transmission system Half-duplex 4 wire system

Synchronization system Start-stop system

Topology Party line (multi-drop)

Transmission speed 1200/2400/4800/9600 bps

Transmission distance 1 km maximum

No. of stations 16 maximum

Tosline 30 is a data highway which allows up to 16 controllers to share 32 registers of common memory. Using Tosline-30 the on/off status of an I/O point (or a 16 bit register value) can easily be transferred from one CPU to any of the other 15 (max) CPUs in the data highway.

Wire

Topology Party line (multi-drop)
Transmission speed 187.5 kbps

Transmission distance 1 km maximum (total)

No. of stations 16 maximum

Transmission capacity 8/16/32 words (cyclic)

Response speed 25 ms/32 words

Checking method Inverted double transmission

Fiber Optic

Topology Star

Transmission speed 375 kbps

Transmission distance 2 km maximum (stn-stn)

No. of stations 16 maximum

Transmission capacity 8/16/32 words (cyclic)

Response speed 19.2 ms/32 words

Checking method Inverted double transmission

NOTE Refer to the Tosline manual for details of the data link system.

2.5 Peripheral devices

The following peripheral devices are available for the M/EX100.

The handy programmer (HP-100)
The handy programmer is a compact, hand-held programmer, that can be used to program the M/EX100 using ladder diagrams.

The program storage module (PRM11)
The program storage module is an external memory dedicated to the M/EX100 program. By using this module, program

saving from the M/EX100 can be done without need of a programmer. Because the program storage module has an EEPROM, maintenance-free program storage and rapid saving/loading can be done.

Data access unit (DP100)
This provides access to all of the registers in the M20/40, and EX100. It provides a simple and easy-to-use method of monitoring and changing register values.

2.6 Internal current consumption

The M20/M40's power supply has the following output capacity.

Internal 5 Vdc 2.0 A (max.)
Internal 24 Vdc 0.16 A (max.)
External 24 Vdc 0.4 A (max.)
-- Total 16 W (max.)

The following table shows the maximum internal current consumption of each unit/module.

Unit	Internal 5 Vdc	Internal 24 Vdc
M20 basic unit	700 mA	40 mA
M40 basic unit	700 mA	60 mA
20 points expansion unit	200 mA	40 mA
40 points expansion unit	200 mA	80 mA

Module		Internal 5 Vdc	External 24 Vdc
DC INPUT	EX10-MDI31	15 mA	-
	EX10-MDI32	80 mA	-
AC INPUT	EX10-MIN51	15 mA	-
	EX10-MIN61	15 mA	-
Relay output	EX10-MRO61	50 mA	140 mA
	EX10-MRO62	40 mA	100 mA
Transistor output	EX10-MDO31	60 mA	35 mA
	EX10-MDO32	250 mA	100 mA
Triac output	EX10-MAC61	300 mA	-
Analog input	EX10-MAI21	50 mA	-
	EX10-MAI22	50 mA	50 mA
	EX10-MAI31	50 mA	50 mA
	EX10-MAI32	50 mA	50 mA
Analog output	EX10-MAO31	70 mA	90 mA
	EX10-MAO22	170 mA	90 mA
	EX10-MAO32	170 mA	90 mA

Pulse input	EX10-MPI21	80 mA	-
Position control	EX10-MMC11	200 mA	100 mA
ASCII/BASIC	EX10-MAS11	800 mA	-
TOSLINE-30	EX10-MLK11	250 mA	-
	EX10-MLK12	250 mA	-

Usable maximum current of the external 24 Vdc output can be determined as follows for the M20/40. Ratings for the EX100 power supply are slightly different, so for calculating EX100 power consumption modify the below examples using the data for the EX100 power supply.

I24 = [16- (5 x I5i + 24 x I24i)]/24

I24e: Usable current of external 24 Vde (A)
I5i: Total current consumption of internal 5 Vdc (A)
I24i: Total current consumption of internal 24 Vdc (A)

NOTE: (1) Maximum output current is limited by output capacity ratings.
(internal 5V: 2A, internal 24V:0.16A, external 24V: 0.4A)

(2) External 24 Vdc is isolated from internal 5/24 Vdc

(Example with a single M20)

M20 I5i = 0.7 (A)
I24e=0.48A (from formula)
I24i=0.04 (A)

Therefore, allowable maximum current of external 24 Vdc is determined as 0.48A for the M20/40.

(Example with an M40 connected to a 40 point expansion unit)

M40 I5i-0.7 + 0.2 (A)
I24e-0.32 A (from formula)
40 exp I24i=0.08 + 0.08 (A)

Allowable maximum current of external 24 Vdc is 0.32A.

(Example with an M20 connected to a two slot rack with two EX10-MAC61 modules in the rack)

M20 15i=0.7 + 2 x 0.3 (A)
124e=0.35 A (from formula)
124i=0.04 (A)

****

3. Specifications

3.1 General specifications

Item	Specifications
Power supply voltage M20/40	(1) 100 to 240 Vac (+10/-15%), 50/60Hz (+/-5%) (2) 24 Vdc (+20/-15%)
Power supply voltage EX100	(1) 100 to 120 Vac (+10/-15%), 50/60Hz (+/-5%) (2) 24 Vdc (+20/-15%) (3) 200 to 240 Vac (+10%/-15%),50/60Hz (+/-5%)
Power consumption	50 VA or less (ac power supply) 22 W or less (dc power supply)
Retentive power fault	10 ms or less
Insulation resistance	10 MOhm or more between power terminals and case
Withstand voltage	1500 Vac, 1 min. between power terminals and case
Ambient temperature	Operating temperature 0 to 55C (32 to 131F) Storage temperature -20 to 75C (-4 to 167F)
Ambient humidity	20 to 90% RH, no condensation
Noise immunity	1000 Vp-p/1 uS, NEMA ICS3-304
Vibration immunity	16.7 Hz - 3 mm p-p (X, Y, and Z directions)
Shock immunity	10 g, 3 times (X, Y, and Z directions)
Weight	M20/20 exp: 1.5kg (3.3lb) M40/40 exp: 2.0kg (4.4lb)
Weight	6 slot unit with I/O's: 2.8 kg (6.1 lb) 9 slot unit with I/O's: 4.0 kg (8.8 lb)

3.2 External dimensions (note that M20/40 can be DIN rail mounted)

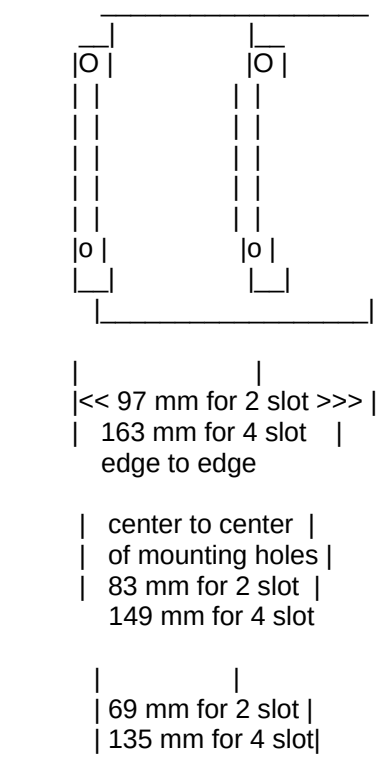
M20 basic unit / 20 point expansion unit

240 mm long (228 mm from center to center of mounting holes)
125 mm wide (108 mm from center to center of mounting holes)
79 mm high

M40 basic unit / 40 point expansion unit

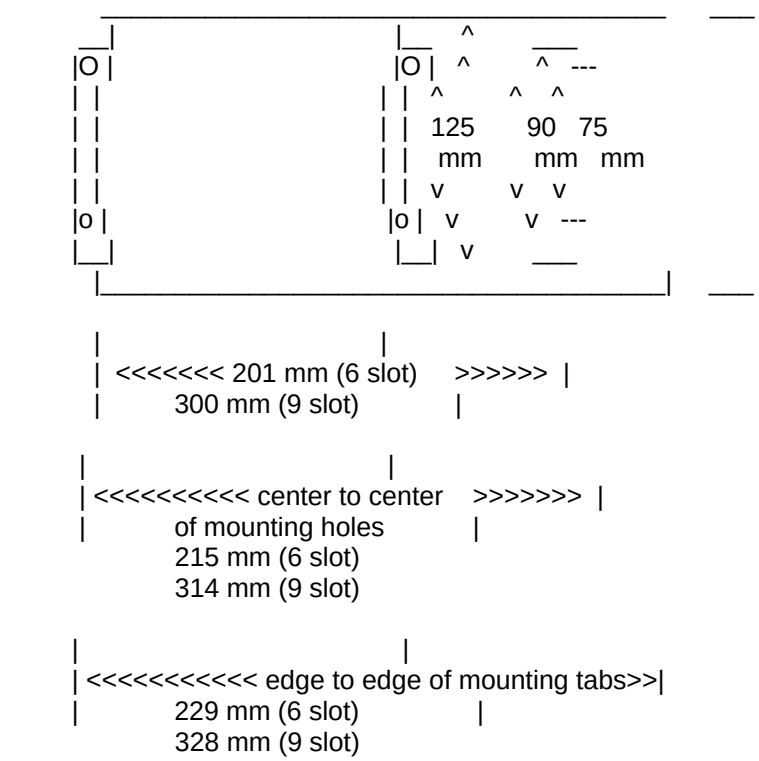
320 mm long (308 mm from center to center of mounting holes)
same width as M20
same height as M20
DIN rail mount is also on unit.

2 and 4 slot expansion racks for M20/40



For 2 and 4 slot racks, DIN rail attachment is not provided.

EX100 Dimensions



The racks are 106.5 mm deep without I/O in them.

With I/O, and without terminal strips, the racks are 115 mm deep.
With I/O, and with terminal strips, the racks are 143 mm deep.

3.3 Functional specifications

Item		Specifications
Control method		Stored program continuous cyclic scan
I/O update		Batch I/O (immediate I/O instruction available)
Program language		Ladder diagram with function block
Memory	Program capacity	3K steps or 4K steps, switch selection. (For 3K steps, 1K word of data is stored in the EEPROM)
	Memory type	EEPROM (Transferred to the RAM on power up)
Instructions		15 basic types, 67 functional types
Execution speed		0.9 uS/contact instruction 110 uS/ 16 bit addition
No. of I/O points		Discrete I/O: 480 points (EX100) Register I/O: 60 registers (1 register = 16 pts)
Internal relays/ registers	Data register	1536 registers (1 register = 16 bits)
	Timer register	120 (0.1 s), 8 (0.01 s) Set value range: 0 to 32767
	Counter register	96, Set value range: 0 to 65535
	Auxiliary relay	960 points/60 registers (area shared)
	Link relay	512 points /32 registers (area shared)
	Special relay	Link status, timing clock, special functions, self diagnosis, and others (64 points total)
	Retentive memory	Data registers, timer registers, counter registers, and auxiliary relays can be partitioned into retentive and non-retentive memory.
Clock-calendar function		Year, month, day, day of the week, hours, minutes, seconds.
Data Link	Computer link	RS-485, 16 stations maximum, 1km maximum
	Link between PCs	TOSLINE-30, twisted pair/optical fiber
	Remote I/O	cable.
Self diagnosis		Memory, I/O bus, program, I/O response, scan time, transmission, and watchdog timer checks
Self monitoring		Error table, scan time

RAM back-up	Built-in capacitor: 7 days/25dC (77dF) Optional battery: 5 years/25dC (EX100) 2 years/25dC (M20/40)
Programming tool	HP100, Programming software
Maintenance tool	DP100 or third party operator interfaces

3.4 I/O specifications

3.4.1 M20 basic unit

INPUT

Item	DC input EX10M20DR5 EX10M20DR6	AC input EX10M20AR5
Input voltage	24 Vdc, +10/-15%	100-120 Vac, +10/-15% (50-60Hz)
Minimum ON voltage	15.0 Vdc	70 Vac
Maximum OFF voltage	5.0 Vdc	25 Vac
Input current	7 mA (24 Vdc) (typ.)	8mA (100Vac,50Hz) typ.
No. of input points	12 points (12 points/common)	12 points (12/common)
ON delay	10ms or less	20ms or less
OFF delay	10ms or less	20ms or less
Withstand voltage	1500 Vac, 1 minute	1500 Vac, 1 minute

OUTPUT

Item	Relay output EX10M20DR5 EX10M20DR6	EX10M20AR5
Load power	24 Vdc, +20% (max.)	140 Vac, +10% (max.)
Maximum load	2A/ pt resistive load, 1A/ pt inductive 2A /2pts common	
Minimum load	50 mW (5V or more)	
No. of output points	8 points (6 x isolated, 1 x 2 points/common)	
ON delay	10 ms or less	
OFF delay	15 ms or less	
Leakage current when OFF	0 ms	
Withstand voltage	1500 Vac. 1 minute	

Power supply 24 Vdc/240Vac (max.)

NOTE Expected relay life: 100,000 operations (electrical)

20 million operations (mechanical)

3.4.2 M40 basic unit

Item	DC input EX10M40DR5 EX10M40DR6	AC input EX10M40AR5 EX10M40AR5
Input voltage	24 Vdc, +10/-15%	100-120 Vac, +10/-15% (50-60Hz)
Minimum ON voltage	15.0 Vdc	70 Vac
Maximum OFF voltage	5.0 Vdc	25 Vac
Input current	7 mA (24 Vdc) (typ.)	8mA (100Vac,50Hz) typ.
No. of input points	24 points (24 points/common)	24 points (24/common)
ON delay	10ms or less	20ms or less
OFF delay	10ms or less	20ms or less
Withstand voltage	1500 Vac, 1 minute	1500 Vac, 1 minute

OUTPUTS	Relay output EX10M40DR5, EX10M40DR6, EX10M40AR5
Load power	24 Vdc, +20% (max.)/240 Vac, +10% (max.)
Maximum load	2A / pt resistive load, 1A/pt inductive 2A /2pts common, 4A/ 4 pts common
Minimum load	50 mW (5V or more)
No. of output points	16 points (6 x isolated, 1 x 2 points/common 2 x 4 points/common
ON delay	10 ms or less
OFF delay	15 ms or less
Leakage current when OFF	0 mA
Withstand voltage	1500 Vac. 1 minute

Power supply 24 Vdc (240 Vac (max.)

NOTE Expected relay life: 100,000 operations (electrical)
20 million operations (mechanical)

3.4.3 20 point expansion unit

INPUTS

DC input
EX10E20DR

Input voltage	24 Vdc, +10/-15%
Minimum ON voltage	15.0 Vdc
Maximum OFF voltage	5.0 Vdc
Input current	7 mA (24 Vdc) (Typical)
No. of input points	12 points (12 points/common)
ON delay	10 ms or less
OFF delay	15 ms or less
Withstand voltage	1500 Vac. 1 minute

OUTPUT Relay output
EX10E20DR

Load power	24 Vdc, +20% (max.) 240 Vac, +10% (max.)
Maximum load	2A / pt resistive load, 1A/pt inductive 2A /2pts common
Minimum load	50 mW (5V or more)
No. of output points	8 points (6 x isolated 1 x 2 points /common)
ON delay	10 ms or less
OFF delay	15 ms or less
Leakage current when OFF	0 mA
Withstand voltage	1500 Vac, 1 minute

power supply 24 Vdc/240 Vac (max.)

NOTE Expected relay life: 100,000 operations (electrical)
20 million operations (mechanical)

3.4.4 40 point expansion unit

	DC input EX10E40DR
INPUTS	
Input voltage	24 Vdc, +10/-15%
Minimum ON voltage	15.0 Vdc
Maximum OFF voltage	5.0 Vdc
Input current	7 mA (24 Vdc) (Typ.)
No. of input points	24 points (24 points/common)
ON delay	10 ms or less
OFF delay	10 ms or less
Withstand voltage	10 ms or less

	Relay output EX10E40DR
OUTPUTS	
Load power	24 Vdc, +20% (max.)/240 Vac. + 10% (max.)
Maximum load	2A/ pt resistive load, 1A / pt inductive
Minimum load	50 mW (5V or more)
No. of output points	16 points (6 x isolated, 1 x 2 points/common 2 x 4 points/common
ON delay	10 ms or less
OFF delay	15 ms or less
Leakage current when OFF	0 mA
Withstand voltage	1500 Vac, 1 minute

Power supply 24 Vdc/240 Vac (max.)

NOTE: Expected relay life: 100,000 operations (electrical)
20 million operations (mechanical)

3.4.5 I/O modules

16-point DC/AC input

Item	DI31 (EX10-MDI31)
Input voltage range	12 to 24 V, +10/-15%, dc/ac (50-60Hz)
Minimum ON voltage	9.6 V
Maximum OFF voltage	3.6 V (0.7 mA or less)
Input current	8 mA (24V (typ.)
No. of input points	16 points (16 points/common)
On delay Mode N	10 ms or less (dc)/20 ms or less (ac)
Mode H	1.5 ms or less (dc)
Off delay Mode N	10 ms or less (dc)/15 ms or less (ac)
Mode H	1.5 ms or less (dc)
Withstand voltage	1500 Vac, 1 minute
Current consumption	15 mA (5 Vdc) or less

32-point DC input

Item		DI32 (EX10-MDI32)
Input voltage		24 Vdc, +10/-15%
Minimum ON voltage		18.0 V
Maximum OFF voltage		6.0 V
Input current		5 mA (24 Vdc) (typ.)
No. of input points		32 points
ON delay	Mode N	10 ms or less
	Mode H	1. 5 ms or less
OFF delay	Mode N	10 ms or less
	Mode H	1.5 ms or less
External connection		2 x 24-pin connector
Common	No. of commons	4
System	Input points	8 points/common
	common	
	Common polarity	Non
Withstand voltage		1500 Vac/ 1 minute
Current consumption		80 mA (4 Vdc) or less

NOTE Cable side connectors (soldering type) are attached as standard.

16-point AC input

Item	IN51 (EX10-MIN51)	IN61 (EX10-MIN61)
Input voltage range (sine wave)	100 to 120 Vac, +10/-15% (50 to 60Hz)	200 to 240Vac, +10/-15% (50 to 60HZ)
Minimum ON voltage	80 Vac	160 Vac
Maximum OFF voltage	30 Vac (2 mA or less)	60 Vac (2mA or less)
Input current	7 mA (100V, 50Hz) (typ.)	6 mA (200V/50Hz) (typ.)
No. of input points	16 points (16 points/common	16 points (16 points/common)
ON delay	20 ms or less	20 ms or less
OFF delay	15 ms or less	15 ms or less
Withstand voltage	1500 Vac, 1 minute	1500 Vac, 1 minute
Current consumption	15 mA (5 Vdc) or less	15 mA (5 Vdc) or less

12-point relay output

Item	RO61 (EX10-MRO61)
Load power	24 Vdc, +20% (max.)/240 Vac, +10% (max.)
Maximum load	2A /pt resistive load, 1A/pt inductive
Minimum load	50 mW (5 V or more)
No. of output points	12 points (4 points/common)
ON delay	10 ms or less
OFF delay	15 ms or less
Leakage current when OFF	0 mA
Withstand voltage	1500 Vac, 1 minute
Current consumption	50 mA (5 Vdc) or less
External power required	24 Vdc +/-10% -140mA (all pts ON), 10mA/pt

8-point isolated relay output

Item	RO62 (EX10-MRO62)
Load	24 Vdc, + 20% (max.)/240 Vac, +10% (max.)
Maximum load	2 A/ pt resistive load, 1A/ pt inductive
Minimum load	50 mW (5 V or more)
No. of output points	8 points (isolated)
ON delay	10 ms or less
OFF delay	15 ms or less
Leakage current when OFF	0 mA
Withstand voltage	1500 Vac, 1 minute
Current consumption	40 mA (5 Vdc) or less
External power required	24 Vdc +/- 10% -100 mA (all pts ON), 10 mA /pt

NOTE Expected relay life: 100,000 operations (electrical)
20 million operations (mechanical)

16-point transistor output

Item	DO31 (EX10-MDO31)
Load power	5 to 24 Vdc, +10/-5%
ON output current	1 A/point (load power 7 V or more) 0.3A/point (load power 7 V or less) 1.2A/4 point (transistor array)
ON resistance	1.5 ohm or less
No. of output points	16 points (16 points/common, o common)
ON delay	1 ms or less
OFF delay	1 ms or less
Leakage current when OFF	0.1 mA or less
Withstand voltage	1500 Vac, 1 minutes
Current consumption	60 mA (5 Vdc) or less

32-point transistor output

Item	(DO32) (EX10-MDO32)
Load power	5 to 24 Vdc, +10/-5%
ON output current	100 mA/point (load power 24 V) 20 mA/point (load power 24 V) 800 mA/common
ON saturated voltage	0.4 V or less
No. of output points	32 points
Output method	current sinking
ON delay	1 ms or less
OFF delay	2 ms or less (typ.)
Leakage current when OFF	0.1 mA or less
External connection	2 x 24-pin connector
Common No. of commons	4
Output pts per common	8 points per common
System common polarity	o common
Withstand voltage	1500 Vdc, 1 minute
Built-in fuses	4 x 2 /common
Current consumption	250 mA (5 Vdc) or less

NOTE Cable side connectors (soldering type) are attached as standard, and are the same as the connectors on the MDI32.

12-point triac output

Item	AC61 (EX10-MAC61)
Load power	100 to 240 Vac, +10/-15% (50 to 60 HZ sine wave)
ON output current	0.5 A /point, 0.6A /SSR
ON saturated voltage	1.5 V or less (0.3 A load)
No. of output points	12 points (4 points/common)
ON delay	1 ms or less
OFF delay	1/2 cycle of load power + 1 ms or less
Leakage current when OFF	1.2 mA (100 Vac) or less, 3 mA (240 Vac) or less
Withstand voltage	1500 Vac, 1 minute
Current consumption	300 mA (5 Vdc) or less (all points ON), 20 mA/pt

4-channel analog input (8-bit)

Item	AI21 (EX10-MAI21)	AI31 (EX10-MA131)
Input range	1 to 5V or 4 to 20mA	0 to 10V
Input impedance	1 to 5V; 500 Kohm or more	500 Kohm more
No. of input points	4 channels, N common	4 channels, N common
Resolution	1 to 5V: 0 to 250 4 to 20 mA: 0 to 250	0 to 10V: 0 to 250
Overall accuracy	+/- 1% (FS)	+/- 1% (FS)
Conversion cycle	Approx. 1 ms	Approx. 1 ms
Wire breakage detection	Yes, for 4 to 20 mA	-
External power failure detection	Yes	Yes
Withstand voltage	1500 Vac, 1 minute	1500 Vac, 1 minute
Current consumption	50 mA (5 Vdc) or less	50 mA (5 Vdc) or less
External power required	12 to 24 Vdc,+/-10%-50mA	12 to 24 Vdc, +/-10% -50mA

Analog to digital conversion formulas:

(a) 4 to 20 mA: D=15.625 A-62.5
(b) 1 to 5V: D=62.5 A-62.5
(c) 0 to 10: D=25A

D: Digital value
A: Analog value

Data format

bits 0 through 7 (8 bits) data ("D" below)
0 to 250 (H00 to HFA)

bit F: Bit for detecting trouble in external wiring ("B" below)
0: Normal
1: Abnormal If all the data bits are 0, the current input cable is open (4 to 20 mA only). If all the data bits are 1, the external power supply is off.

bits 8 through E: Always 0

	B	*	*	*	*	*	*	*	D	D	D	D	D	D	D	D
XW:	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

Register assignment: XW(n) Ch1
XW(n+1) Ch2
XW(n+2) Ch3
XW(n+3) Ch4

4-channel analog input (12-bit)

Item	AI22 (EX10-MAI22)	AI32 (EX10-MAI32)
Input range	1 to 5V or 4 to 20mA	-10 to +10V
Input impedance	1 to 5V: 1 M-Ohm or more, 40 to 20 mA: 250 ohm	1 M-Ohm or more
No. of input points	4 channels, N common	4 channels, N common
Resolution	1 to 5V: 0 to 4000 4 to 20 mA: 0 to 4000	-10 to +10V: -2000 to 2000
Overall accuracy	+/- 0.5% FS/25 degr.C +/-1% FS/ 0 to 55 degr. C	+/-0.5% FS/25 degr. C +/-1% FS/0 to 55 degr. C
Conversion Cycle	Approx. 9.6ms/4 channels	Approx. 9.6ms/4 channels
Wire breakage detection	Yes, for 4 to 20mA	-
External power failure detection	Yes	No
Withstand voltage	1500 Vac, 1 minute	1500 Vac, 1 minute
Current consumption	50 mA (5 Vdc) or less	50 mA (5 Vdc), or less
External power required	24 Vdc, +/-10%-50 mA	24 Vdc, +/-10%-50 mA

Analog to digital conversion formulas:

(a) 4 to 20 mA: D=250 A-1000
(b) 1 to 5V: D=1000A-1000
(c) +- 10V: D=200A

D: Digital value
A: Analog value

Data format

bits 0 through B (12 bits)
0 to 4000

bit F: Bit for detecting trouble in external wiring
0: Normal
1: Abnormal If all the data bits are 0, the current input cable is open (4 to 20 mA only). If all the data bits are 1, the external power supply is off.

bits B through F: sign bits for +-10V scale

bits C,D,E are always equal to 0 in 4-20mA or 1-5V scales

bit F is for wire breakage detection in 1-5V and 4-20mA scale
0: Normal
1: Abnormal

for +- 10V scale:
- 2000 to 2000 (HF830 to H07D0)
Two's complement is used if the value is negative

XW: F E D C B A 9 8 7 6 5 4 3 2 1 0

Register assignment: XW(n) Ch1
XW(n+1) Ch2
XW(n+2) Ch3
XW(n+3) Ch4

2-channel analog output (8-bit)

Item	AO31 (EX10-MAO31)
Output range	0 to 10 V, 1 to 5 V, 4 to 20 mA
Load impedance	5 V full-scale terminal: 5 Kohm or more 10 V full-scale terminal 10 Kohm or more 20 mA full-scale terminal: 600 Kohm or less
No. of output channels	2, voltage and current paired (N common)
Resolution	0 to 250 (full scale)
Overall accuracy	+/- 1% (FS)
Conversion cycle	Approx. 1 ms
External power failure detection	No
Withstand voltage	1500 Vac, 1 minute
Current consumption	70 mA (5 Vdc) or less
External power required	24 Vdc, +/- 10% - 90 mA

1 to 5V/4 to 20mA
is set at pre-shipment
inspection. Refer to
the circuit configuration
for other settings

Digital to analog conversion formulas:

(a) 4 to 20 mA: A = 0.064 (D + 4)
(b) 1 to 5V: A = 0.016 (D + 1) A: Analog value
(c) 0 to 10 V: A = 0.04 D D: Digital value

Data format

D: Data bit (8 bits)
0 to 250 (H00 to HFA)
bits 0 through 7 are used for data
bits 8 through F are not used

Register assignment YW(n) CH1
YW(n+1) CH2

If the immediate output instruction (FUN097) is used, two registers (both channels) should be specified as immediate output registers.

2-channel analog output (12-bit)

Item	AO22 (EX10-MAO22)	AO32 (EX10-MAO32)
Output range	1 to 5 Vdc or 4 to 20 mA	-10 to +10Vdc
Load impedance	1 to 5 V: 5 Kohm or more 4 to 20 mA: 600 ohm or less	5 Kohm or more
No. of output channels	2 channels, (N common) (voltage and current paired)	2 channels (N common)
Resolution	1 to 5 V: 0 to 4000	-10 to +10V -2000 to +2000
Overall accuracy	+/-0.5% FS/25 degr C +/-1% FS/0 to 55 degr C	+/-0.5% FS/25 degr C +/-1% FS/0 to 55 degr C
Conversion cycle	Approx. 1 ms	Approx. 1 ms
External power failure detection	No	No
Withstand voltage	1500 Vac, 1 minute	1500 Vac, 1 minute
Current consumption	170 mA (5 Vdc) or less	170 mA (5 Vdc) or less
External power required	24 Vdc +/-10% -90 mA	24 Vdc, +/-10% -90 mA

Digital to analog conversion formulas:

(a) 4 to 20 mA: A = 0.004 (D + 4)
(b) 1 to 5 V: A = 0.001 (D + 1) (A: Analog value)
(c) +/- 10 V A = 0.005 D (D: Digital value)

Data format

4 to 20 mA/ 1 to 5V

Data bit (12 bit)
0 to 4000 (H0000 to H00FA)
bits 0 through B are used for data; with +- 10V scale, bit B is the sign bit
bits C through F are not used
0: Positive
1: Negative

Data bit (11 bits)
-2000 to 2000 (HF830) to H07D0)
Two's complement if negative

If the immediate output instruction (FUN097) is used, two registers (both channels) should be specified as immediate output registers.

1-channel pulse input

Item		PI-21 (EX10-MPI21)
Input voltage	A, B, M	12V, +10/-20% (12V setting)/5V, +10/-20% (5V setting)
	EXT	12 to 24 Vdc, +10/-15%
Minimum ON voltage	A, B, M	9V (12V setting) / 3.5V (5 V setting)
	EXT	9.6 V
Maximum OFF voltage	A, B, M	2V (12V setting) / 1V (5V setting)
	EXT	3.6V
Input current	A,B,M	12V-7.5mA (12V setting/5 V-10 mA (5V setting)
	EXT	24V-10 mA, 12V-5mA
Input channel		1 channel (phase A, B, M, and EXT)
Count speed		100 kpps (max.) (pulse width 4 uS or more)
Counter		24-bit binary
Pulse mode	Quadrature	Phase A, B (90 degree phase shift), up/down
	Up/down	Phase A: count up phase B: count down
Counter mode	Normal	Always count enable
	Hold	Both M and EXT ON: count enable Either M or EXT OFF: count value held
Counter reset		Count value is reset to zero at the moment when both M and EXT are turned ON
EXT ON/OFF delay		5 ms or less
Withstand voltage		1500 Vac, 1 minute
Current consumption		80 mA (5 Vdc) or less

The input voltage of A,B, and M are set to 12V, and the counter mode is set to quadrature prior to shipping.

Data format

SSSSSSSSSDDDDDDDDDDDDDDDDDDDDDDD S=sign bits (all are the same value),
FEDCBA9876543210FEDCBA9876543210 D=data bits.
|_____||_____|
XW(n) XW(n+1)

24 bits counter - 0 to 16,777,215 (HOOFFFFF)

NOTE: Bits 8 to F of XWn are same as bit 7 of XWn.
Normally, bits 8 to F should be masked by user program as follows.
(e.g. storing the count value into D0100 - D0101)

Pulse mode
<Quadrature>
Count up when phase B leads phase A
Count down when phase A leads phase B

<Up/Down>
Count up on every rising edge of pulses on A
Count down on every rising edge of pulses on B

Counter mode
<Normal>
When phase A/B is up, then the counter counts up.
When phase A/B is down, then the counter counts down.
When the phase M is on, then the counter immediately resets.

<Hold>
When phase A/B is up, then the counter counts up.
When phase A/B is down, then the counter counts down.
Counter only counts when phase M is high.
Counter resets on the rising edge of phase M.
Counter "holds" when phase M is low.

NOTE: If the immediate input instruction (FUN096) is used, two registers should be specified as immediate input registers.

Normally, bits 8 to F of XWn are masked by user program as follows:

```
|--[XW(n) AND .H00FF ->> D0100]---
|
|--[XW(n+1) W->>W D0101]---
|
```

Motion control

Item	MC11 (EX10-MMC11)
No. of control axes	1 axis
Control units	Pulse / inch / mm etc.
Control range	+/- 999,999 units
Point data capacity	64 points
Absolute max. speed	200 kpps
Operation speed	Origin return speed, max. speed, min. speed
Acceleration/deceleration system	Automatic trapezoidal/triangular system
Acceleration/deceleration rate	0 to 26 s
Backlash compensation	0 to 1000 pulses
Zero position offset	+/- 999,999 units
Dwell time	0 to 99 s
Registers	X + Y 4 W (64 bits) 4 registers (2XW,2YW)
Parameter storage	EEPROM
Input voltage	12/24 Vdc (Z-phase; 5/12/24 Vdc)
Input current	10 mA (24 V)
Input ON/OFF voltage	9.6 V / 3.2 V
ON/OFF delay	5 ms (Z-phase; 1 ms)
Output Pulse output (switch setting) Mode	1) CW/CCW, error counter clear 2) Pulse/direction, error counter clear
Output method	Open collector (5-24 Vdc, 50 mA)
ON/OFF delay	2 uS
RUN output Output method	Open collector (5-24 Vdc, 50mA)
RUN output Operation	ON during normal operation
Current consumption Internal	200 mA - 5 Vdc 400 mA - 5 Vdc
External	100 mA - 12/24 Vdc

Connector pin arrangement

	A	B	
RUN output (0 V)	1	1	RUN output (5-24 Vdc)
CW/Pulse output 0 V)	2	2	CW/Pulse output (5-24 Vdc)
CCW/Direction output (0 V)	3	3	CCW/Direction output (5-24 Vdc)
Error counter clear output (0V)	4	4	Error counter clear output (5-24 Vdc)
Z-phase input (0 V)	5	5	Z-phase input (5 Vdc)
Z-phase input (0 V)	6	6	Z-phase input (12/24 Vdc)
Origin marker input (0 V)	7	7	Origin marker input (12/24 Vdc)
Hold input	8	8	Hold input (12/24 Vdc)
Emergency stop input	9	9	Emergency stop input (12/24 Vdc)
CW limit input	10	10	CW limit input (12/24 Vdc)
CCW limit input	11	11	CCW limit input (12/24 Vdc)

External power (0 V)	12	12	External power (12/24 Vdc)

For additional information refer to the manual on the MC11, the UM-EX100-E004

****

4. Installation and Wiring

4.1 Operating environment

Do not install the M/EX100 in the following locations:

- Where the ambient temperature drops below 0 degr. C (32 degr. F) or exceeds 55 degr. C (131 degr. F).
- Where there is condensation due to sudden temperature changes.
- In locations subject to vibration that exceeds tolerance
- In locations subject to shock that exceed tolerance
- Where there are corrosive or flammable gases.
- In locations subject to dust, machining debris or other particles
- In locations exposed to direct sunlight.

Observe the following precautions when installing enclosures in which the M/EX100 will be mounted:

- Provide the maximum possible distance between high-voltage cables or high power panels. This distance must be at least 200 mm (8 in).
- If installing the enclosures in the vicinity of high-frequency equipment, be sure to correctly ground the enclosures.
- When sharing the channel base with other panels, check for leakage current from the other panels or equipment.

4.2 Installing the unit

NOTE The M20/M40 and the expansion units come equipped with a bracket at the rear for mounting on a 35 mm DIN rail.

Installation precautions:

- Because the M/EX100 is not dust-proof, install it in a dust-proof enclosure.
- Do not install the unit directly above equipment that generates a large amount of heat, such as a heater, transformer, or large-capacity resistor.
- Allow at least 70 mm (2.8 in) on all sides of the unit for ventilation.
- For safely during maintenance and operation, install the unit as far as is possible from high-voltage or power equipment. Alternatively, keep the unit separate using a steel plate or similar separator.
- If installing the unit near high-voltage or power equipment, the grounding requires special attention. See Section 4.5.
- Be sure to install the unit vertically.

4.3 Mounting the modules

The I/O modules are mounted in the slots of the expansion rack.

1. Install I/O modules starting from the left slot, taking care to securely insert them into the slots of the rack.

2. Insert the modules fully until the front panels of the modules are locked in the rack.

CAUTION For safety, be sure to turn off power to the M/EX100 before installing or removing a module.
After installing the modules, secure the unit to keep the modules vertical. Also, be sure that the modules remain vertical if transporting the unit.

4.4 Connecting the expansion unit/rack

One expansion unit (20 exp/40 exp) or expansion rack (2-slot/4-slot) can be connected to the basic unit of the M20/40. One 6 or 9 slot expandable EX100 rack can be connected as an expansion rack to a base EX100 having an expandable rack.

A 30 cm cable is supplied with the expansion unit for the M20/40. A 10 cm cable is optionally available depending on the configuration.

For the EX100, the racks do not come with a cable; be sure to purchase the expansion cable in additional to the expansion rack.

To connect an expansion unit/rack, remove the expansion connector cover(s), then securely connect an expansion cable to the expansion connectors.

4.5 Grounding

The optimum method for grounding electronic equipment is to ground it separately from other high-power systems, and to ground more than one units of electronic equipment with a single-point ground.
The M/EX100 has a noise-proof design, sufficient to withstand industrial operating conditions. Although the M/EX100 itself can resist noise (signal interference) without being grounded, for safety and reliability, grounding is recommended.

4.5.1 Check points for grounding

Check the grounding against the following criteria.

1. The case containing the electronic equipment must not become a path for a ground current. A high-frequency current is particularly harmful.
2. Equalize the ground potentials when more than two units of electronic equipment are to be connected. Ground them at a single point.
3. Do not connect the ground of the EX100 to that of high-power systems.
4. Do not use a ground that has an unstable impedance, such as painted

screws, or grounds subject to vibration.

4.5.2 The GND terminal is provided on the M/EX100 basic unit and the expansion units for grounding purposes. In case of the expansion racks for the M20/40, the rack mounting screw is used for this purpose.

. 2mm2 (14AWG) or larger wire should be used as grounding wire.
  (GND terminal screw: M3.5)
. 100 ohms or less to earth is recommended.

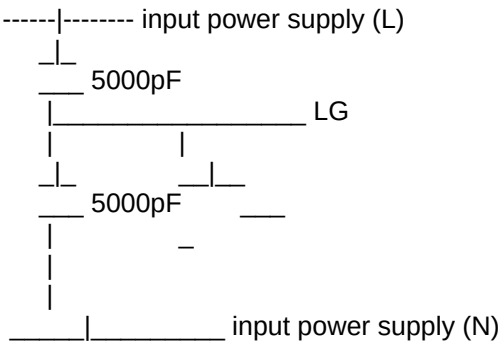
4.6 Wiring the power supply

Wire the external power supply to the M/EX100 power supply terminals (L and N).

For EX100 power supply:

Normally, the LG and FG terminals are shorted. However, if the leakage current through LG is a problem, or if the ground for the power supply system is provided separately, the LG terminal should be open.

Circuit diagram showing LG:



If there is no suitable grounding or no dedicated ground available, isolate the EX100 from the mounting frame. There will be no problem operating the EX100 as long as the FG terminals of the basic and expansion units are connected. However, for safety, provide single-point grounding to a point, with impedance.
Resistance: Ground the frame through a 1W - 1Kohm
Inductance: Ground the frame through 2A - 100 micro Henry

Power conditions

Retentive power failure: Normal operation for less than 10 ms
. 2mm2 (14AWG) or larger twisted-pair cable should be used for the power supply. (Terminal screw: M3.5)
. Power supply cable should be separated from other cables.

Power Supply for M20/40

Wire the external power supply to the M20/40 power supply terminals (L and N).

Power Conditions
Rated voltage: 100 to 240 Vac, +10 / -15%, 50 / 60 Hz (AC PS)
24Vdc, +20 / -15% (DC PS)
Power consumption: 50VA or less (AC PS)
22W or less (DC PS)
Retentive power failure: Normal operation for less than 10ms.
2mm2 (14AWG) or larger twisted-pair cable should be used for the power supply.
(Terminal screw: M3.5)
Power supply cable should be separated from other cables.
GND terminal should be connected to ground.

4.7 I/O wiring and application precautions

This section describes precautions for wiring the I/O signals. Precautions for applications are given only as reference for wiring field inputs and outputs.

4.7.1 I/O wiring precautions

. Refer to section 3.4 for instructions on how to properly wire the I/O terminals.
. 0.75mm2 (18AWG) to 0.3mm2 (22AWG) wires are recommended for I/O signals.
. Separate the I/O signal cables from high-power cables by at least 200mm (8in).
. If expansion unit/rack is used, separate the expansion cable from the power and I/O signal cables by at least 50mm (2in.)
. It is recommended to separate the input signal cables from output signal cables.

CAUTION Terminal blocks are detachable. However, for safety, be sure to turn off power before installing or removing the terminal block.

4.7.2 Dimensions of the terminal block

On the terminal blocks for the M20, M40, and EX100, the dimensions are as follows:
center of screw head to center of screw head: 8.6 to 8.7 mm
divider between screws: 1.3 to 1.5 mm thick
right edge of divider to left edge of next divider: 7.2 to 7.3 mm
Terminal screw size: M3.5
Applicable wire size: 2mm2 (14AWG) to 3mm2 (22AWG)

4.7.3 Application precautions for input signals

1. Minimum ON/OFF time of the input signal.
The following conditions guarantee correct reading of the ON/OFF state of the input signal:
Input ON time: ON delay time + the time for one scan
Input OFF time: OFF delay time + the time for one scan
The ON and OFF times of the input signals must be longer than these intervals.

2. The reliability of some contacts cannot be guaranteed by the specified input current. In this case, install an external bleeder resistor.

4. When a switch with an LED is used, the input sometimes cannot recognize

that the switch is off due to the current leakage through the LED. In this case, install a bleeder resistor to reduce input impedance.

5. With ac input signals, if the external cable is long or if a multi-core cable is used, an induced current can flow from the charged wire to the open wire, in proportion to the capacities of the cables. In this case, sometimes the voltage reaches the level of the ON input even though the contact is open, causing the input to malfunction for no apparent reason. The usual practice when this happens is to reduce input impedance. Install a resistor or a resistor and capacitor between the input and common terminals, or use shielded cables.

   Such precautions are necessary when dealing with a large number of ac input signals.

6. If an ac output sensor is connected, it is sometimes not possible to detect the OFF state due to a leakage current. This problem can be rectified by installing an external bleeder resistor.

   Select a bleeder resistor according to the following criteria:
   a) The voltage between the input terminals must be lower than the OFF voltage when the sensor is switched off.
   b) The current must be within the allowable values when the sensor is switched on.
   c) Calculate the wattage of the bleeder resistor by multiplying the current when the sensor is switched on times three.

4.7.4 Application precautions for output signals

1. Expected relay life is 100,000 electrical cycles and 20 million mechanical cycles.

2. The relay output does not contain protective fuses. Fuses rated for the output should be provided by the user.

3. Where an inductive load is connected to the output, a relatively high energy transient voltage will be generated when the relay turns OFF. To prevent the problems caused by this surge, install a surge absorber in parallel to the inductive load. See precautions for transistor and triac output modules.

4.7.5 Application precautions for transistor output modules

1. Power must be supplied to the internal control circuit of the transistor output module. If power is connected with the polarities reversed, the internal fuse will blow. Be sure polarity is correct.

2. Over current protection
   The transistor output module contains the fuse(s). The transistor cannot always be protected against a shorted load. The fuse can, however, protect the transistor module if the pattern inside the module burns, or if the external cable burns.

3. Output surge protection
A relatively large surge occurs if an inductive load connected to the output opens. This surge passes through the external wiring and sometimes adversely affects other systems. To eliminate this inductive load, install a surge absorber in parallel with the load.

flywheel diode (for clamping voltage)
Inverse withstand voltage: At least three times that of the power supply.
Forward current: Larger than the load current

varistor (for clamping voltage)
The voltage rating is roughly twice the maximum (peak) voltage of the power supply.

snubber (CR) (for attenuating high frequencies)
R and C in series. R is 0.5 to 1 ohm per volt coil voltage
C is 0.5 to 1 microfarad per amp of coil current (non-polarity capacitor)

4.7.6 Application precautions for triac output modules

1. Overcurrent protection
The triac module contains one 2 amp fuse for every four output points. When the load short circuits, the fuses are designed to blow to protect the triacs. However, if a fuse blows, semiconductor devices may be damaged to some extent. Therefore, when installing this module, short circuits must be avoided. Pay particular attention to the wiring.

2. Output surge protection

Install a surge absorber in parallel with the load.
Varistor: The voltage rating is roughly 1.2 times the maximum (peak) voltage of the power supply.
Snubber (CR in series for attenuating coil high frequencies)
R is 0.5 to 1 ohm per volt coil voltage
C is 0.5 to 1 uF ampere amp of current (non polarity capacitor)

4.7.7 Application precautions for relay output modules

1. A power supply of 24 Vdc must be supplied to the relay drive circuit of the relay output module. Connect a power supply of 24 Vdc +/- 10% between the positive and negative terminals.

2. The relay output module does not contain a fuse for protection against overload. Be sure to install appropriate fuses. Fuse each output or use one 5A fuse for each group of four relay outputs (4 relays per common for MRO61). If there is no fuse protecting the module, the patterns inside the module will be burned in the event of a short circuit, overload, etc.

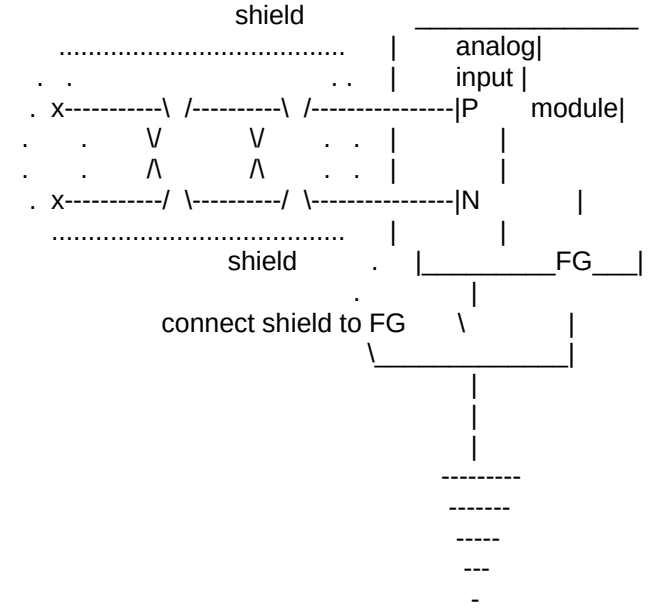
3. Output surge protection
As mentioned in the sections concerning the triac and transistor output modules, it is necessary to install a surge absorbing device for protection where an inductive load is used.

4.7.8 Application precautions for analog input modules

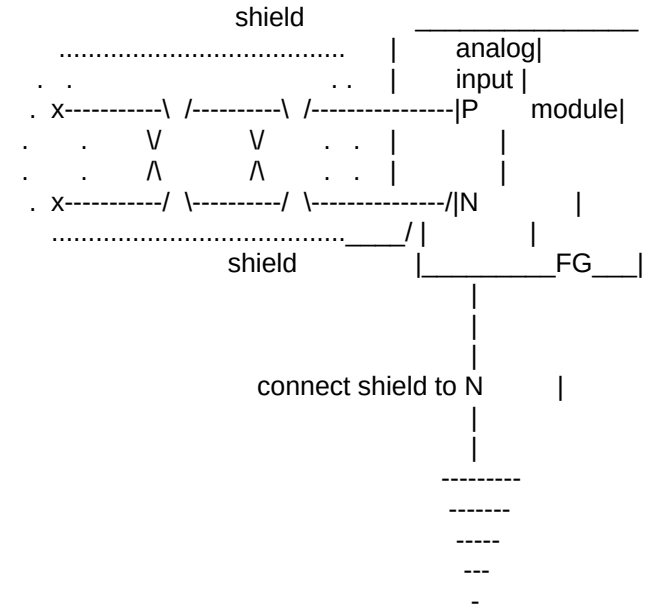
1. An external power supply of 24 Vdc +/- 10% must be supplied to the analog input module. The wires carrying the power supply should be separated from other wires to prevent signal interference.

2. The shield of the input cable should be grounded as (1) below. However, in some cases, (2) or (3) will be more effective.

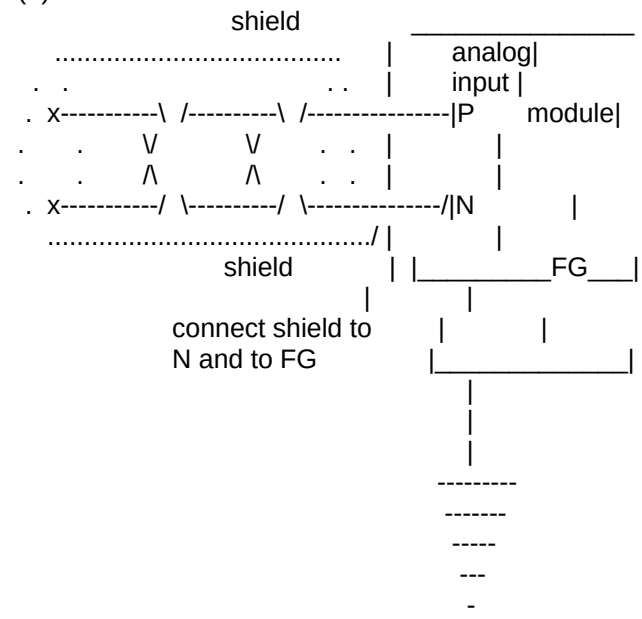
(1)



(2)



(3)

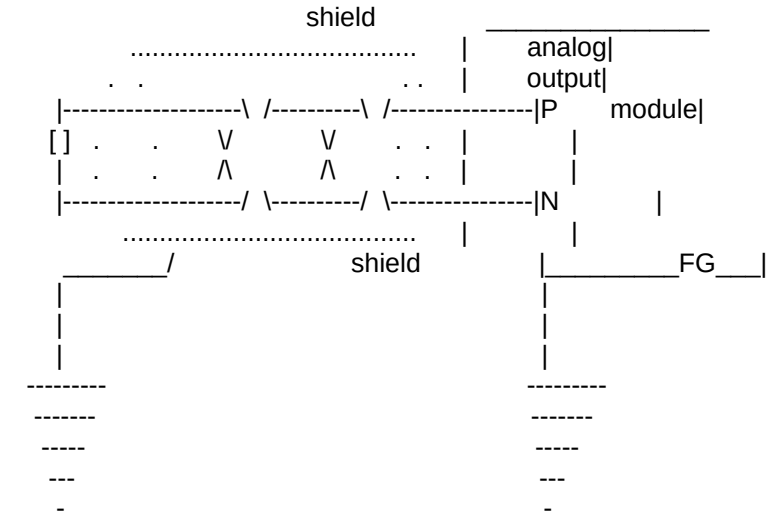


4.7.9 Application precautions for analog output modules

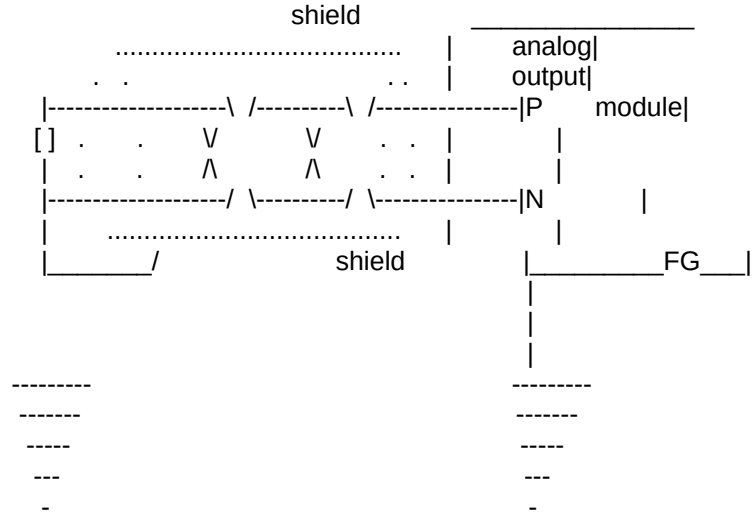
1. An external power supply of 24 Vdc +/- 10% must be supplied to the analog output module. The wires carrying the power supply should be separated from other wires to prevent signal interference.

2. The shield of the output cable should be grounded as 1) below. However, in some cases, 2) or 3) will be more effective.

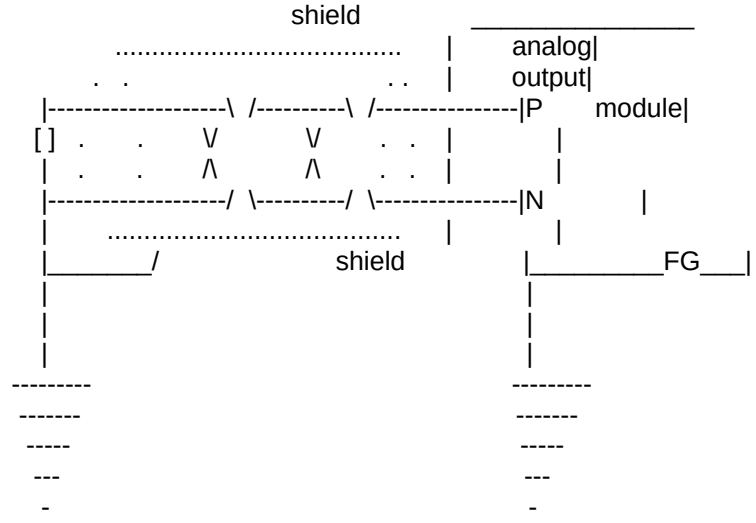
(1)



(2)


shield
analog
output
module
shield
FG

(3)


shield
analog
output
module
shield
FG

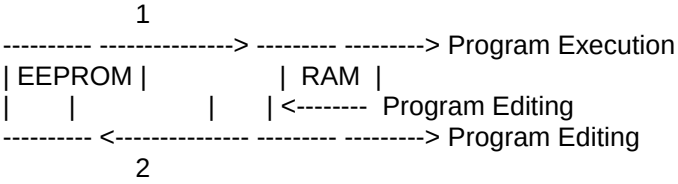
****

5. Operating System

5.1 EEPROM Operation

The M/EX100 is equipped with a built-in EEPROM as a standard feature. The user program is stored in the EEPROM so that the user program can be maintained without the need for a battery.

The user program stored in the EEPROM is transferred to the RAM when power is turned on. Subsequent program execution is done based on the RAM contents. Program editing is also done based on the RAM contents. Therefore, if the program is modified, it is necessary to execute the EEPROM write operation by using the programmer.



To write to the EEPROM with a hand held programmer (HP 100) do the following:

Put the PLC into HALT
Execute the PROM WRITE instruction on the CNTL menu

To write to the EEPROM with the PDD programming software, do the following:

Put the PLC into HALT
Go to the system information menu
Press F1
Select F5 (write EEPROM)

There are two methods for doing the EEPROM write operation with a GP programmer (OBSOLETE)

(1) Enter the EEPROM write command (control 94) from the programmer.
(Key operation using the GP110)
(CNTL) (9) (4) (EXE) (EXE)

(2) Set special relay R62E to ON by using the programmer. (Key operation example using GP100)

(MON) (EXE) Set the monitor mode.
(HOME) Move the cursor to the auxiliary data monitor area.
(STS) (R) (6) (2) (E) (EXE) Register R62E.
(DSET) (1) (EXE) Set R62E to ON

NOTE: (1) If a programmer that does not support the EEPROM write command is used, method (2) should be used.

(2) The R62E is reset to OFF automatically after EEPROM write is completed.

5.2 Memory settings

The internal memory of the M/EX100 consists of program memory and data memory. Program memory is used to store the user program. Data memory is used to store the ON/OFF status of external I/O and various control data.

As explained previously, the M/EX100 has an EEPROM for memory back-up. The user can select the EEPROM utilization in either 4K-step mode or the 3K-step mode by using the DIP switch (see 2.3.1)

4K-step mode: Program memory (4K steps max.) is stored in the EEPROM

3K-step mode: Program memory (3K steps max.) and the contents of 1024 data registers (D0512 to D1535) are stored in the EEPROM

Internal memory configurations for each mode are shown in the following illustration.

On the M20/40, the 3K/4K selection is made via DIP switches. See section 2.3.1 for information on this.

On the EX100, the 3K/4K selection is made via DIP switches on the CPU. You set switch 1 ON for 4K, OFF for 3K. Switch 2 is not used.

The following table shows the conditions for transferring program (and data) from the EEPROM to the RAM when power is turned on.

EEPROM data / Memory setting at power on	4k-step mode	3k-step mode
3K or less of program	4K-step mode: Transfers program normally	3K-step mode: Transfers program and data.(1)
3K or more program	4K-step mode: Transfer program normally	3K-step mode: Does not transfer(2)
Transfers only program	4K-step mode: Transfers program normally	3K-step mode: Transfers program and data normally.

1. The program is properly transferred, but the data in the registers D0512 to D1535 is undefined.

2. The operation is inhibited because of a memory setting mismatch. It is necessary to clear the memory.

NOTE (1) When the 3K-step mode is selected, data registers DO512 to D1536 should be specified as retentive memory. Otherwise, these data are transferred, but then cleared at initialization. (see 5.4)

(2) When the 3K-step mode is selected, data stored in the EEPROM can be accessed by the user program. (see 9.5)

5.3 Operation modes

The M/EX100 has two basic operation modes, the RUN mode and the HALT mode. The M/EX100 also has the ERROR mode for use as an abnormal state.

RUN: In the RUN mode, the M/EX100 reads external signals, executes the user program stored in the RAM, and outputs signals to the external devices according to the user program. It is in the RUN mode that the M/EX100 performs scans the user program logic, which is the basic operation of PCs. A program is not normally modified with the RUN mode changing function, which enables program changes when the system is in operation. This enables the program to be modified without turning off external outputs in the RUN mode.

HALT: In this mode, execution of a user program is stopped, and all outputs are switched off. This is the mode in which programming is normally performed. Writing the program into the EEPROM can only be performed when the M/EX100 is the HALT mode.

ERROR: The M/EX100 enters the error mode if internal trouble is detected during self-diagnosis, and normal functioning cannot continue. Program execution stops and all outputs are switched off. To exit from the error mode, enter an Error Reset command from the programmer, or cycle power off and then on again.

When the operation correct switch is in the RUN-P position, program modifications and EEPROM write operation cannot be performed. (An error message will be displayed on the programmer if an attempt is made) Therefore the user program can be protected from unauthorized operation by setting the switch in the RUN-P position.

The following table lists the available functions relevant to the program modifications in each switch position.

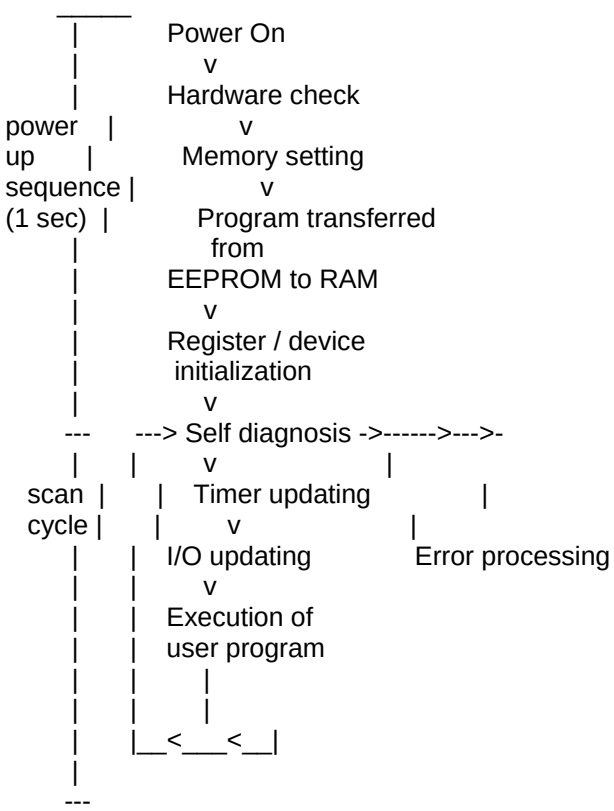
Switch position	Programmer's command	Operation mode	Available functions
HALT	RUN/RUN-F/HALF commands are disabled.	HALT	Programming and EEPROM write are available.
RUN	RUN/RUN-F/HALT commands are enabled.	RUN	On-line program changes EEPROM write is not available.
		HALT	Programming and EEPROM write are available.
RUN-P	RUN/RUN-F/HALF commands are enabled.	RUN	On-line program changes and EEPROM write are not available.
		HALT	Programming and EEPROM write are not available.

NOTE (1) Even when the key position is RUN-P, the contents of data memory in RAM can be changed.

(2) On-line program changes are not allowed when:
1) The total number of program control instructions. i.e., END, MCS, MCR, JCS, JCR, are changed.
2) Execution order of program control instructions are changed.
3) If the operation mode has entered ERROR, writing operations are disabled. Execute the error reset command to return to HALT mode.

5.4 Scanning

The flowchart below shows the internal operations performed by the M/EX100 from the time power is turned on through program execution. As the diagram shows, executing a program consists of continuous scanning operations. One scan is a cycle starting with self diagnosis and ending with the completion of program execution.



Hardware check: Checks and initializes the memory, ICs and I/O buses.

Memory setting: Sets a 3K or 4K system according to the DIP switch setting

Program transfer: Transfers data from the EEPROM to the RAM

Register/device initialization: Initializes registers and devices (see next page)

Self diagnosis: Checks for the existence of errors (see next page)

Timer updating: Updates timing relays and timer registers.

I/O updating: Updates external I/O registers and link registers.

User program execution: Executes the user program

Error processing: Processes errors detected in the M/EX100.

Initializing registers and devices when power is turned on

Register/device	Retentive memory	Retentive data (KEEP AREA TOP) Forced devices	Forced coils
External input and output (X/Y)	Cannot be set.	Holds only devices specified as forced	Holds only devices specified as forced coils
Auxiliary relay (R)	Holds registers specified for retentive memory.	Cannot be set.	Holds only devices specified as forced coils.
Timer (T)	Holds registers specified for retentive memory.	Cannot be set.	Cannot be set.
Counter (C)	Holds registers	Cannot be set.	Cannot be set.
Data register (D)	Holds registers specified for retentive memory	Cannot be set.	Cannot be set.
Link register (Z)	Cannot be set.	Holds only devices specified as forced.	Holds only devices specified as forced coils.

NOTE When power is turned on, all registers and devices that are not designated as retentive are cleared to 0. Retentive devices and registers consist of devices that are forced, and devices and registers specified in the retentive memory (Keep Area Top).

Self diagnosis

Item	Check method
Program memory	Checks the program using check sum.
Program syntax	Checks for the existence of the END instruction, checks the syntax of JCS/JCR, MCS/MCR, and output operands.
Scan time	Checks program scan line.
I/O	Checks the response from I/O modules.
Illegal instructions	Checks for the existence of illegal instructions.
TOSLINE	Checks the data link modules.
Computer link	Checks the computer link interfaces.
Watchdog timer	Checks the processor operation.

****

6. Programming

6.1 Devices and registers

The M/EX100 program consists of bit-based instructions that handle ON/OFF information, such as contact and coil instructions, and register-based (16-bit) instructions, such as those for data transfer, and arithmetic operations. Devices are used to store the on/off information of contacts and coils, and registers are used to store 16-bit data.

Devices are divided into four types:
X: External input devices
Y: External output devices
R: Auxiliary relay devices
Z: Link devices

Registers are divided into seven types:
XW: External input registers
YW: External output registers
RW: Auxiliary relay registers
ZW: Link registers
T: Timer registers
C: Counter registers

Device and register numbers

X devices share the same memory area as XW registers. Device X033, for example, represents the number 3 bit in the XW03 register.

Bit position / Number

```
    F  E  D  C  B  A  9  8  7  6  5  4  3  2  1  0
XW12 0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
```

Thus, "X123 is ON" means that bit number 3 of XW12 is 1. It also means that X123 = 1. Y, R, and Z devices work in a similar manner.

A device number consists of a register number and bit position.

```
 X 12 3
 | V |
 |  | |
 |  | |------> Represents bit position 0 to F in the register.
 |  |
 |  |--------> Decimal number representing the register
 |             containing the corresponding device.
 |
 |-----------> Represents the type of device. (X,Y,Z, or R).
```

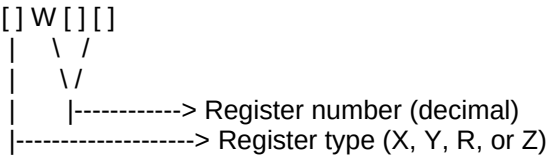
Addressing devices
External input, external output, auxiliary relay and link devices.

```
ex.  R   1   0   A    (R10A)
   [ ] [ ] [ ] [ ]
    |   \   /   |
    |    \ /    |---------> Bit position (0 to F) in the register
    |     |---------------> Register number (decimal)
    |---------------------> Device type (X,Y,R or Z)
```

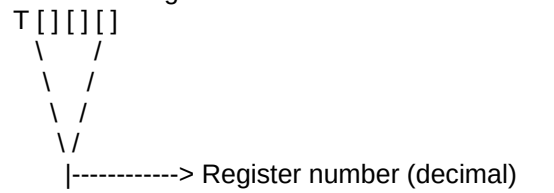
e.g. X000, Y027, R10A, Z31F, etc.

Addressing registers

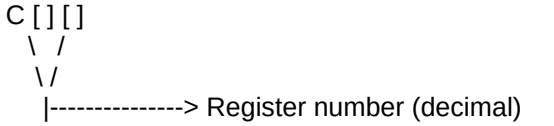
External input, external output, auxiliary relay and link registers.



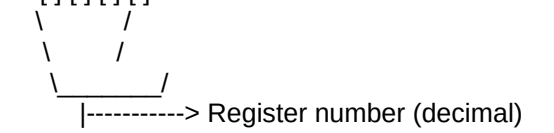
Timer register



Counter register



Data register



NOTE (1) The available data range in each register is 0 to 65535 (H0000 to HFFFF) except for the timer register. The timer register range is 0 to 32767 (H0000 to H7FFF).

(2) Double-length (32-bits) data is available in two consecutive registers. (e.g. D0100 and D0101)
(MSB) FEDCBA9876543210 FEDCBA9876543210 (LSB)
D0100 | D0101
Upper | Lower
16 bits | 16 bits

Device/register	Symbol	No. of points	Address
External input device	X	512 points total (X and Y)	X000-X31F
External output device	Y		Y000-Y31F
External input register	XW	64 registers total (X and Y)	XW00-XW63
External output register	YW		YW00-YW63
Auxiliary relay device	R	1024 points	R000-R63F
Auxiliary relay register	RW	64 registers	RW00-RW63
Data register	D	1536 registers	D000-D1535
Link device	Z	512 points	Z000-Z31F
Link register	ZW	32 registers	ZW00-ZW32
Timer register	T	128 registers	T000-T127
Counter register	C	96 registers	C00-C95

NOTE Registers can be regarded as a group of 16 continuous devices, except for data registers (D), timer registers (T), and counter

registers (C). External input registers XW00 to XW31 and external output registers YW00 to YW31 can also be used as devices. However, registers XW32 to XW63 and YW32 to YW63 can only be used as registers.

External input devices (X)
These devices indicate the on/off states of inputs through the input modules. External input devices can be used many times in a program. The function type X is assigned to input modules.

External output devices (Y)
External output devices store the on/off signals that drive the external devices via the output modules. They can be used for coils in a program. The function type Y is assigned to output modules.

External input registers (XW)
These are 16-bit registers for storing values, such as analog input, pulse input, and values received from the input modules. The function type, XW, is assigned to input modules. The number of registers is determined by the module.

External output registers (YW)
These 16-bit registers are used for storing values, such as analog output and numerical indicators for output via the output modules. The function type, YW, is assigned to output modules. The number of registers is determined by the module.

Auxiliary relay devices and registers (R/RW)
The auxiliary relay devices, R, are used to store intermediate results of sequences. The auxiliary relay registers, RW, are used to store temporary results of functional instructions. The data in R/RW cannot be output directly to the output modules. It is necessary to move the data to Y/YW. It is possible to make these registers retentive (KEEP AREA TOP) so that they retain data in the event of a power failure.
The topmost area of the devices, R600 to R63F, is assigned to the special relays, as explained below.

Data registers (D)
Data registers are the same as auxiliary relay registers, RW, except that data registers cannot be used as devices. If the memory setting is the 3K mode, 1K of the registers (D0512 to D1535) can be saved in the EEPROM as fixed data. It is possible to specify these registers in retentive memory to retain their data the event of a power failure.

Link devices and registers (Z/ZW)
Link devices and registers are used for the TOSLINE-30 data highway. Each register is specified either as a TALKER or LISTENER. The data in the TALKER register is sent to other stations in the TOSLINE-30 network, and the LISTENER register receives data from the other stations in the network. If the TOSLINE-30 is not used, these registers can be used as RW registers.

Timer registers (T)
Timer registers are used for storing the remaining time of timer instructions, such as the on and off delay timers and single-shot timers. These registers cannot be used for storing the results of functional instructions. It is possible to specify these registers in retentive memory to retain their data in the event of a power failure.
T000 to T119: 100 ms timers
T120 to T127: 10 ms timers

Counter registers (C)
Counter registers are used to store the current count of counter instructions. These registers cannot be used for storing the results of functional instructions. It is possible to specify these registers in retentive memory to retain their data in the event of a power failure.

Special relays

Devices R600 to R63F are assigned to the special relays as listed below.
These relays can be used for interlocking in a program.

Device	Name	Comments
R600	Data link normal ZW00	ON if transmission to the corresponding register is normal. OFF if a transmission error occurs, or if transmission is not used
R601	Data link normal ZW01	
R602	Data link normal ZW02	
R603	Data link normal ZW03	
R604	Data link normal ZW04	
R605	Data link normal ZW05	ON if transmission becomes normal
R606	Data link normal ZW06	
R607	Data link normal ZW07	This area can be used as a normal auxiliary relay devices if the Tosline 30 data highway is not used.
R608	Data link normal ZW08	
R609	Data link normal ZW09	
R60A	Data link normal ZW10	
R60B	Data link normal ZW11	
R60C	Data link normal ZW12	
R60D	Data link normal ZW13	
R60E	Data link normal ZW14	
R60F	Data link normal ZW15	
R610	Data link normal ZW16	
R611	Data link normal ZW17	
R612	Data link normal ZW18	
R613	Data link normal ZW19	
R614	Data link normal ZW20	
R615	Data link normal ZW21	
R616	Data link normal ZW22	
R617	Data link normal ZW23	
R618	Data link normal ZW24	
R619	Data link normal ZW25	
R61A	Data link normal ZW26	
R61B	Data link normal ZW27	
R61C	Data link normal ZW28	
R61D	Data link normal ZW29	
R61E	Data link normal ZW30	
R61F	Data link normal ZW31	

R620	Timing relay 0.1 s	Timing relays are on for half of their cycle, and off for the other half. So the 1 second timer will be on for 0.5 seconds, then off for 0.5 seconds. Note that 0.1 and 0.2 timing relays sometimes cannot be read if the scan time is long.
R621	Timing relay 0.2 s	
R622	Timing relay 0.4 s	
R623	Timing relay 0.8 s	
R624	Timing relay 1 s	
R625	Timing relay 2 s	
R626	Timing relay 4 s	
R627	Timing relay 8 s	
R628	Calendar function flag	Used for the calendar function.
R629	HOLD device	Enters the HOLD state when ON
R62A	EEPROM write flag	Used for EEPROM write instructions
R62B	Communication priority mode flag	Used for communication priority mode
R62C		Reserved by system-cannot be used
R62D	Auto RUN-F enable flag	Used for the auto RUN-F function.
R62E	Always OFF (EEPROM write device)	Writes to the EEPROM when ON.
R62F	Always ON	Always ON relay
R630	Self diagnosis error: CPU error	Watchdog timer error
R631		Reserved by the system
R632	ON if an error occurs	Reserved by the system
R633	Cleared by resetting the	EEPROM error EEPROM data is abnormal
R634	error	I/O error I/O bus is abnormal
R635		I/O reference error Incorrect I/O allocation
R636		Program error Program data abnormal
R637		Scan time over Scan time exceeded 200 ms
R638		Reserved by system
R639		Reserved by system
R63A	Communication port flag	ON=link / OFF=programmer
R63B	Programmer transmission error	ON if communication with programming abnormal.

R63C	Automatically reset to OFF when restored	TOSLINE error Computer link error.	Abnormality found in TOSLINE-30 Computer link abnormal
R63D			
R63E	Execution of diagnosis instruction		ON when user-specified diagnosis instruction is executed
R63F			Reserved by system

6.2 I/O allocation

I/O allocation is performed to assign the M/EX100's internal I/O registers to the available I/O.

The external input registers/devices (XW/X) are assigned to the input terminals. The external output registers/devices (YW/Y) are assigned to the output terminals.
Register numbers of the external input and output registers are consecutive. Thus one register number can be assigned for either input or output.

The link registers/devices (ZW/Z) are assigned to the TOSLINE-30 modules independent of the external input and output registers.

The external input and output registers can be regarded as a group of 16 continuous devices.

NOTE See 6.1 for details of registers/devices.

I/O allocation methods

There are two methods used for I/O allocation:

(1) Automatic I/O allocation:
When the automatic I/O allocation command is entered , the M/EX100 recognizes the hardware configuration, then assigns the internal I/O registers accordingly.

(2) Manual I/O allocation:
Manual I/O allocation is done by programming the type of I/O modules into the I/O slots one by one. This method is used when it is necessary to program without the M/EX100, or when it is necessary to set modules that cannot be performed by automatic I/O allocation.

I/O allocation for basic and expansion units

(1) M20 basic unit
When the automatic I/O allocation has been executed, the following I/O allocation table is created in the M20's memory.

I/O allocation table

Slot	I/O type
0	CPU
1	X 1W

2	----
3	Y 1W

Then the external input/output devices are assigned to the I/O terminals as follows.

X000 through X00B: Inputs
Y020 through Y027: Outputs

NOTE (1) X0OC to X0OF are not used in the M20 (always OFF)
(2) YW01 (Y010 to Y01F) and Y028 to Y02F are not assigned to the hardware. They can be used as internal relays.
(3) If the manual I/O allocation is used, the above I/O allocation table must be programmed.

(2) M40 basic unit

When the automatic I/O allocation has been executed, the following I/O allocation table is created in the M40's memory.

I/O allocation table

Slot	I/O type
0	CPU
1	X 1W
2	X 1W
3	Y 1W

Then the external input/output devices are assigned to the I/O terminals as follows.

X000 through X017: inputs
Y020 through Y02F: outputs

NOTE (1) X018 to X01F are not used in the M40 (always OFF).
(2) If the manual I/O allocation is used, the above I/O allocation table must be programmed.

(3) M20 and 20 point expansion unit

I/O allocation

Slot	I/O type
0	CPU
1	X 1W
2	----
3	Y 1W
4	X 1W
5	----
6	Y 1W

The external input/output devices are assigned to the I/O terminals as follows.

X000 through X00B: base unit inputs
Y020 through Y027: base unit outputs
X030 through X03B: expansion unit inputs
Y050 through Y057: expansion unit outputs

X00C to X00F, X03C to X03F are always off
YW01, Y028 to Y02F, YW04, and Y058 to Y05F are not used.

(4) M40 and 20 point expansion unit

When the automatic I/O allocation has been executed, the following I/O allocation table is created in the M20's memory.

I/O allocation table

Slot	I/O type
0	CPU
1	X 1W
2	X 1W
3	Y 1W
4	X 1W
5	----
6	Y 1W

The external input/output devices are assigned to the terminals as follows.

X000 through X00F: base unit inputs
X010 through X017: base unit inputs
Y020 through Y02F: base unit outputs
X030 through X03B: expansion unit inputs
Y050 through Y057: expansion unit outputs

NOTE (1) X0OC to X0OF, X03C to X03F are always OFF
(2) YW01, Y028 to Y02F, YW04 and Y058 to Y05F can be used as internal devices only.
(3) For the manual I/O allocation, program the above I/O allocation table.

(5) M40 and 40 point expansion unit

When the automatic I/O allocation has been executed, the following I/O allocation table is created in the M40's memory.

I/O allocation table

Slot	I/O type
0	CPU
1	X 1W
2	X 1W
3	Y 1W
4	X 1W
5	X 1W
6	Y 1W

X000 through X00F: base unit inputs
X010 through X017: base unit inputs
Y020 through Y02F: base unit outputs
X030 through X03F: expansion unit inputs
X040 through X047: expansion unit inputs
Y050 through Y05F: expansion unit outputs

NOTE (1) X018 to X01F and X048 to X04F are always OFF.
(2) For the manual I/O allocation, program the above I/O allocation table.

As shown in the examples before, the basic and expansion units have the following I/O types.

UNIT		I/O TYPE	ASSIGNED DEVICES
M20 basic unit	0	CPU	Input: X000 to X00B
	1	X 1W	Output: Y020 to Y027
	2		
	3	Y 1W	
M40 basic unit	0	CPU	Input: X000 to X017
	1	X 1W	Output: Y050 to Y057
	2	X 1W	
	3	Y 1W	
20 point	4	X 1W	Input: X030 to X03B
expansion unit	5		Output: Y050 to Y057
	6	Y 1W	
40 point	4	X 1W	Input: X030 to X047
expansion unit	5	X 1W	Output: Y050 to Y05F
	6	Y 1W	

NOTE (1) The I/O type is expressed by a combination of the function type (X: input or Y: output) and the number of registers assigned (W: word = 16 points)

(2) Slot 2 of the M20 and slot 5 of the 20 points expansion unit are allocated as blanks. One external output register (YW) assigned internally to the blank slot.

I/O allocation for expansion racks

If an expansion rack is connected to the basic unit, the expansion rack shares slots 4 and sequentially after the basic unit's allocation.
I/O modules mounted on the expansion rack have the module types as listed below. When the automatic I/O allocation has been executed, the M/EX100 recognizes the module types mounted, creates the I/O allocation table, then determines devices/registers assignments to the modules.

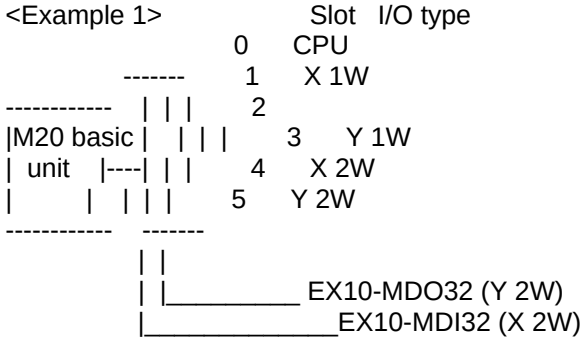
The modules types that are read to the M/EX100 CPU by automatic I/O allocation are listed in the following table.

Part No.	Description	Module Type
EX10-MDI31	16-point dc/ac input (12-24 Vdc/ac	X 1W
EX10-MDI32	32-point dc input (24 Vdc)	X 2W
EX10-MIN51	16-point ac input (120-120 Vac)	X 1W
EX10-MIN61	16-point relay output (240 Vac 23 Vdc)	X 1W
EX10-MRO61	12-point relay output (240 Vac/24 Vdc)	Y 1W
EX10-MRO62	8-point output (isolated) 240 Vac/24 Vdc)	Y 1W
EX10-MDO31	16-point transistor output (5-24 Vdc)	Y 1W
EX10-MDO32	32-point transistor output (5-24 Vdc)	Y 2W
EX10-MAC61	12-point triac output (100-240 Vac)	Y 1W
EX10-MAI21	4 ch analog input (4-20 mA/1-5 V)	X 4W
EX10-MAI22	4 ch analog input (4-20 ma/1-5 V)	X 4W
EX10-MAI31	4 ch analog input (0-10 V)	X 4W
EX10-MAI32	4 ch analog input (+/- 10 V)	X 4W
EX10-MAO31	2 ch analog input (0-10V/1-5V/4-20 mA)	Y 2W
EX10-MAO22	2 ch analog output (1-5V/4-20 mA)	Y 2W
EX10-MAO32	2 ch analog output (+/- 10 V)	Y 2W
EX10-MPI21	1 ch pulse input	X 2W
EX10-MMC11	1 axis motion control	X+Y=4W
EX10-MAS11	ASCII/BASIC	X+Y=4W
EX10-MLK11	TOSLINE-30 (wire) 8-word setting	Z 8W
	16-word setting	Z 16W
	32-word setting	Z 32W
EX10-MLK12	TOSLINE-30 (optical) 8-word setting	Z 8W
	16-word setting	Z 16W
	32-word setting	Z 32W

NOTE (1) The module type is expressed by a combination of the function type (X, Y or Z) and the number of registers assigned.

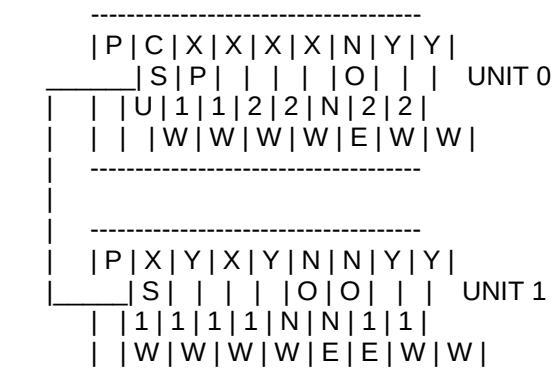
(2) For the TOSLINE-30, the link registers (ZW) are assigned independent of the external input and output registers. The number of registers assigned is determined by the set status of the DIP switches on the module. (Transmission capacity)

<Example 1>



<Example2>

Two 9 slot EX100 expandable racks



Unit	Slot	Module type	Register assignment
0	0	CPU	
	1	X 1W	XW00
	2	X 1W	XW01
	3	X 2W	XW02, XW03
	4	X 2W	XW04, XW05
	5	Blank	(YW06)
	6	Y 2W	YW07, YW08
	7	Y 2W	YW09, YW10
1	0	X 1W	XW11
	1	Y 1W	YW12
	2	X 1W	XW13
	3	Y 1W	YW14
	4	Blank	(YW15)
	5	Blank	(YW16)
	6	Y 1W	YW17
	7	Y 1W	YW18

MANUAL I/O ALLOCATION

The following table lists the module types available when using manual I/O allocation.

Type	No. of assigned registers	Remarks
X	01, 02, 04, 08	Input
Y	01, 02, 04, 08	Output
X + Y	02, 04, 08	Input/Output
Z	08, 16, 32	TOSLINE-30
Blank	01	Vacant slot
iX	01, 02, 04, 08	Input::::::::::Batch I/O
iY	01, 02, 04, 08	Output::::::::update is
iX + Y	02, 04, 08	Input/Output:::not executed
SP	01, 02, 04, 08, 16, 32	Space (Vacant slot)
OPT	---	Option

(1) Module type is expressed by a combination of the type and the number of registers assigned.

(2) A slot containing no module is allocated as blank, and one external output register (YW) is assigned internally.

(3) iX, iY, iX + Y, SP and OPT are available only when manual I/O allocation is used.

(4) Batch I/O update is not executed for modules allocated as iX, iY or iX + Y. I/O update for such modules is executed only when the immediate input or output instructions are executed.

(5) SP is used for assigning the optional number of registers to a vacant slot. External output registers (YW) are assigned internally.

(6) OPT does not occupy any registers. OPT can be used for a vacant slot to which no register assigned.

NOTE For the ASCII/BASIC module, it is necessary to use manual I/O allocation in order to change its module type from "X + Y 4W" to "iX + Y 4W".

Rules for I/O allocation

(1) If an expansion unit or rack is connected, I/O registers are assigned to it sequentially after the I/O registers assigned to the base unit.

(2) Registers are assigned to modules sequentially from left to right.

(3) Vacant slots are allocated as blanks during automatic I/O allocation. One external output register (YW) is assigned internally to the blank setting.

(4) Consecutive register numbers are assigned to input and output modules.

(5) Link registers (ZW) are assigned to the TOSLINE-30 module.

(6) The minimum allocation unit is one register, that is 16 bits. Therefore, for example, when one register is assigned to a 12-point module, bits C to F are no used.

NOTE Operation (RUN) with vacant slots between allocated modules is possible if the vacant slots have been designated as blank, SP or OPT. However, if the slot settings are other than these three, operation (RUN) is not possible. An error will occur from the module-response check. In such a case, the forced operation (RUN-F) can be used to override the module-response check. (See 9.2)

6.3 Setting the retentive memory area (Keep Area Top)

Retentive memory area can be specified for the following registers.

- Auxiliary relay devices/registers (R/RW)
- Data registers (D)
- Counter registers (C)
- Timer registers (T)

Retentive registers retain the previous data at initialization in the power up sequence. (See 5.4)

To specify registers for the retentive memory area, select the first register by using the programmer's system information editing function. By this operation, the first specified register to the highest number register is specified as retentive memory.

For example, if following settings have been entered:

RW ... 16
D ... 0
C ... 50
T ... Does not set

The M/EX100's retentive memory areas will be designated as follows.

RW16 to RW63
D0000 to D1535
C50 to C95

NOTE (1) Data in the retentive memory are backed up by a built-in capacitor. (Back-up period: 7 days at 25 degrees C). An optional battery is available for longer back up.

(2) If the memory setting has been set to the 3K mode, the contents of D0512 to D1535 are stored in the EEPROM. In this case, this area should be specified as retentive memory. Otherwise this data will be transferred from EEPROM to RAM, but then cleared at initialization. (See 5.2)

6.4 Program Configuration & Execution

The stored program comes from the system information settings and the execution program. (Normally the execution program is simply called "the user program").

The system information contains program-related data such as program ID, retentive memory areas, I/O allocation, etc. System information has a 128-step capacity.

The execution program stores the user application program, which was written in ladder diagram language. It has a 3968-step capacity when the memory setting is 4K mode. The execution program has a 2944-step capacity when the memory setting is 3K mode.

The execution program is stored in memory by pages and networks.

Each page has the following capacity.

. 14 columns by 11 lines
. 154 steps / page
. 32 steps / rung or circuit

Program execution order:

Executes sequentially from page 1 to the page having an END instruction.

On each page, the program executes sequentially rung 1, then rung 2, and so forth.

Each rung in a ladder diagram is executed according to the following rules:

Rule 1: Execution from left to right in a simple line
Rule 2: OR logic is executed first
Rule 3: Execution from upper lines to lower lines in branches.
Rule 4: Execution according to a combination of rules 2 and 3
Rule 5: Reverse power flow (right to left) is not possible.

****

7.0 INSTRUCTIONS

General: The M20/M40/EX100 series PLCs have 15 types of basic ladder instructions and 67 types of function block instructions. The operation of these functions is explained in detail in section 7.2. It is highly recommended that Section 6, PROGRAMMING, be read (for comprehension) before starting this section.

Contact and Coils: Toshiba PLCs use free format ladder logic programming. Coils and contacts can be written and used almost exactly as they would be drawn on paper. In general, a coil and a transitional contact can have only one unique address (two coils in the program can not have the same address/reference number).

Function Blocks: For efficiency and advanced programming capability, function blocks can be integrated into the ladder logic. Function blocks cause operations to be performed on registers. When working with function blocks, the following are important:

* A function block operation does not alter the numerical value of the source register(s).
* A function block operation changes the numerical value in the destination register, based on the value in the source register(s).
* Source and destination registers are also referred to as operands.
* A function block can end a circuit (a line of logic). It is not necessary to use a coil as is shown in the examples.

7.1a List of Instructions

This list of instructions is grouped according to their location in the EXPDD250 edit menu. For a detailed description of each instruction , see 7.2-7.8

[F1] - Sequence Instructions:

-] [- NO Contact
-]/[- NC Contact
-]^[- Leading edge transitional contact
-]v[- Trailing edge transitional contact
OPEN No vertical connection
CLOSE Vertical connection
----- Horizontal connection
-x x- No horizontal connection
-( )- Coil
*-( )- Forced coil
TON On-delay timer
TOF Off-delay timer
SS Single-shot timer
CNT Counter
END End of program

[F2] - Extra Sequence Instructions:

MCS Master control set
MCR Master control reset
JCS Jump control set
JCR Jump control reset
STIZ Step sequence initialize {FUN} {100}
STIN Step sequence input {FUN} {101}
STOT Step sequence output {FUN} {101}
SET Set bit {FUN} {080}
RST Reset bit {FUN} {081}
SR Shift register {FUN} {112}
F/F Flip-flop {FUN} {110}
U/D Up/down counter {FUN} {111}

[F3] - Move and Compare Instructions:

W -> W Register to register move {FUN} {000}
K -> W Constant to register move {FUN} {001}
TINZ Table initialize {FUN} {002}
T -> W Table to register move {FUN} {003}
W -> T Register to table move {FUN} {004}
T -> T Table to table move {FUN} {005}
DDSP Diagnostic display {FUN} {090}
DDSM Diagnostic display w/ message {FUN} {091}
R > R Compare register > register {FUN} {014}
R > K Compare register > constant {FUN} {024}
R = R Compare register = register {FUN} {015}
R = K Compare register = constant {FUN} {025}
R < R Compare register < register {FUN} {016}
R < K Compare register < constant {FUN} {026}

[F4] - Arithmetic Instructions:

R + R Add register to register {FUN} {010}
R + K Add register to constant {FUN} {020}
R - R Subtract register from register {FUN} {011}
R - K Subtract constant from register {FUN} {021}
R x R Multiply register by register {FUN} {012}
R x K Multiply register by constant {FUN} {022}
R / R Divide register by register {FUN} {013}
R / K Divide register by constant {FUN} {023}
R ++ R Double length register add {FUN} {017}
R -- R Double length register subtract {FUN} {018}
RT Square root {FUN} {070}
SIN Sine {FUN} {071}
ASIN Arc-sine {FUN} {072}
COS Cosine {FUN} {073}
ACOS Arc-cosine {FUN} {074}

[F5] - Logical Operation Instructions

R AND R Register AND register {FUN} {030}
R AND K Register AND constant {FUN} {040}
R OR R Register OR register {FUN} {031}
R OR K Register OR constant {FUN} {041}
R EOR R Register Exclusive OR register {FUN} {032}
R EOR K Register Exclusive OR constant {FUN} {042}
NOT Invert register bits {FUN} {033}
NEG 2's Complement of register {FUN} {046}
RTR Rotate register bits right {FUN} {035}
RTL Rotate register bits left {FUN} {036}
TEST Register bit test for 1 {FUN} {043}

[F6] - Convert & Limit Instructions

BIN BCD to binary conversion {FUN} {050}
BITC Bit count of bits = 1 {FUN} {055}
BCD1 Binary to BCD conversion {FUN} {051}
BCD2 Binary to double length BCD {FUN} {052}
ENC Encode position of bit = 1 {FUN} {053}
DEC Decode position and set bit=1 {FUN} {054}
UL Set upper limit value {FUN} {060}
LL Set lower limit value {FUN} {061}
MAX Find maximum value in table {FUN} {062}
MIN Find minimum value in table {FUN} {063}
AVE Find average value in table {FUN} {064}
FG Function generator {FUN} {065}
IN Read the input values now {FUN} {096}
OUT Write the output values now {FUN} {097}
READ Block read from OPT module {FUN} {???}
WRITE Block write to OPT module {FUN} {???}

7.2. Basic Ladder Instructions

{--| |--} NORMALLY OPEN CONTACT

INPUT --| |-- OUTPUT
(A)

LEGAL ENTRIES

(A) DEVICE TYPE R, X, Y, Z

OPERATION

A normally open contact referenced to address (A). It provides circuit continuity when (A) is ON.

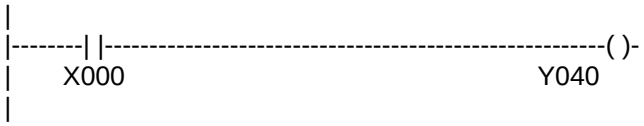
EXECUTION TIME (microseconds) 0.9

MEMORY REQUIREMENTS 1 step

EXAMPLE DESCRIPTION

Y040 will be ON only when X000 is ON. Y040 will be OFF only when X000 is OFF.

EXAMPLE CIRCUIT



TIMING DIAGRAM X = ON _ = OFF

X000 _____XXXXX_____XXXXX

Y040 _____XXXXX_____XXXXX

{--|/|--} NORMALLY CLOSED CONTACT

INPUT --|/|-- OUTPUT
(A)

LEGAL ENTRIES

(A) DEVICE TYPE R, X, Y, Z

OPERATION

A normally closed contact referenced to address (A). It opens the circuit when (A) is ON.

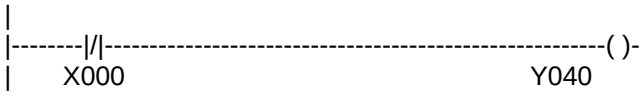
EXECUTION TIME (microseconds) 0.9

MEMORY REQUIREMENTS 1 step

EXAMPLE DESCRIPTION

Y040 will be ON only when X000 is OFF. Y040 will be OFF only when X000 is ON.

EXAMPLE CIRCUIT



TIMING DIAGRAM X = ON _ = OFF

X000 XXXXX_____XXXXX_____

Y040 _____XXXXX_____XXXXX

{--( )--} RELAY COIL

INPUT --( )--|
(A)

LEGAL ENTRIES

(A) DEVICE TYPE R, Y, Z

OPERATION

Output or status of address (A) is either ON or OFF depending on the input. Outputs may also be controlled through the "force" command. When "forced", outputs hold their last state and do not follow their inputs. A coil displayed with a small "x" on the rung to the left of the coil is a forced coil. Refer to the "forced coil" instruction for details on forcing outputs
* No other instruction can be written on the right hand side of a coil.

EXECUTION TIME (microseconds) 1.2

MEMORY REQUIREMENTS 1 step

EXAMPLE DESCRIPTION

When the input X000 is true (ON), device Y040 is ON. When the input is false (OFF), the device Y040 is OFF.

EXAMPLE CIRCUIT

```
|
|--------|/|-------------------------------------------------------( )-
|       X000                                              Y040
```

TIMING DIAGRAM X = ON _ = OFF

X000 XXXXX_____XXXXX_____

Y040 _____XXXXX_____XXXXX

-X-( )--| FORCED COIL

INPUT -X-( )--|
(A)

LEGAL ENTRIES

(A) DEVICE TYPE R, Y, Z

OPERATION

A forced coil maintains the preceding state regardless of the input condition. In this example the coil maintains the ON state. If it is then forcibly reset, it maintains the OFF state
* The force coil is a debugging function. The state of the forced coil device can be set ON or OFF by the HP100, DP100, or programming software.

EXECUTION TIME (microseconds) 1.2

MEMORY REQUIREMENTS 1 step

EXAMPLE DESCRIPTION

The Y010 output stays ON regardless of the state of the input X000.

EXAMPLE CIRCUIT

```
|
|--------|/|----------------------------------------------------X--( )-
|       X000                                              Y010
```

TIMING DIAGRAM X = ON _ = OFF

X000 __XXXX_____XXXXX_____

Y040 XXXXXXXXXXXXXXXXXX

{--|^|--} TRANSITIONAL CONTACT - RISING EDGE

INPUT --|^|-- OUTPUT
(A)

LEGAL ENTRIES

(A) DEVICE TYPE R only

OPERATION

A single pulse (one Program scan) contact closure is triggered by the input transition from OFF to ON (leading edge).
* This instruction creates a pulse signal in response to a change that sets the input ON
* Device (A) stores the input state of the preceding scan
* Do not duplicate device (A) anywhere else in the program

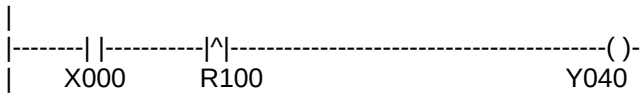
EXECUTION TIME (microseconds) 1.2

MEMORY REQUIREMENTS 1 step

EXAMPLE DESCRIPTION

Y040 will turn ON for one scan only when the preceding logic to device R100 makes the transition from OFF to ON.

EXAMPLE CIRCUIT



TIMING DIAGRAM X = ON _ = OFF

X000 _____XXXXX_____XXXXXXXXXX_____

Y040 _____X_____X_____

{--|v|--} TRANSITIONAL CONTACT - TRAILING EDGE

INPUT --|v|-- OUTPUT
(A)

LEGAL ENTRIES

(A) DEVICE TYPE R only

OPERATION

A single pulse (one program scan) contact closure is triggered by the input transition from ON to OFF (trailing edge). At transition the contact maintains circuit continuity at its output for one scan.
* This instruction creates a pulse signal in response to a change that sets the input from ON to OFF
* Device (A) stores the input state of the preceding scan
* Do not duplicate device (A) anywhere else in the program

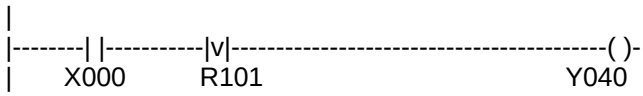
EXECUTION TIME (microseconds) 1.2

MEMORY REQUIREMENTS 1 step

EXAMPLE DESCRIPTION

Y040 will turn ON for one scan only when the preceding logic to device R101 makes the transition from ON to OFF.

EXAMPLE CIRCUIT



TIMING DIAGRAM X = ON _ = OFF

X000 _____XXXXX_____XXXXXXXXXX_____

Y040 __________X_____________X_____

{MCS}, {MCR} MASTER CONTROL SET AND MASTER CONTROL RESET

INPUT---[MCS]--|
--------[MCR]--|

OPERATION

When input to MCS is ON, operation is normal. When the input to MCS is OFF, the left hand power rail turns OFF between the MCS and MCR instructions. Therefore, all outputs are turned OFF, regardless of input logic, and no special functions are executed.
* MCS and MCR instructions must be used in pairs.
* If MCS and MCR are inserted in a program in the incorrect order, an error will result.
* Nesting of MCS and MCR instructions is not permitted.
* Input logic for MCR instruction is not required.

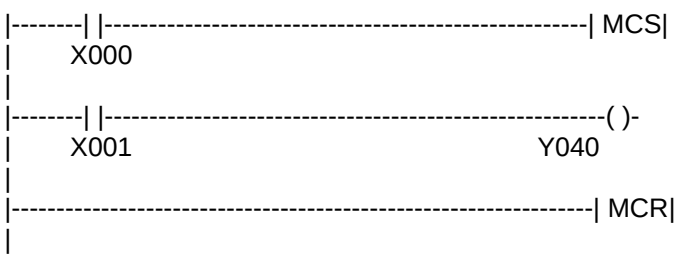
EXECUTION TIME (microseconds) 0.6

MEMORY REQUIREMENTS 1 step

EXAMPLE DESCRIPTION

When X000 is on, MCS is energized and the following logic is performed as normal. When X000 is OFF, MCS is de-energized and the logic up through the its MCR command is no longer activated. This results in any outputs between MCS & MCR being turned OFF.

EXAMPLE CIRCUIT



{JCS}, {JCR} JUMP CONTROL SET AND JUMP CONTROL RESET

INPUT---[JCS]---|
    ---[JCR]---|

OPERATION

When the input to JCS is ON, all rungs of ladder logic between JCS and JCR are skipped over and are not solved during scan execution. All outputs maintain their status unless they are manipulated elsewhere in the program. When the input to JCS is OFF, the logic is scanned and solved as in normal operation.
* JCS and JCR instructions must be used in pairs.
* If inserted in a program in the incorrect order, an error will result.
* Nesting of JCS and JCR instructions is not permitted.
* Input logic for JCR instruction is not required.
* No more than two JCS and JCR combinations may be programmed consecutively.

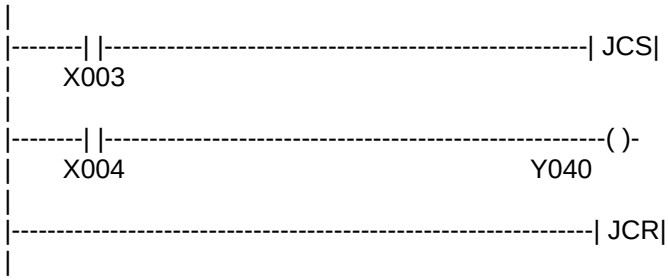
EXECUTION TIME (microseconds) 0.6

MEMORY REQUIREMENTS 1 step

EXAMPLE DESCRIPTION

When X000 is ON, JCS is energized and the logic on the lines between the JCS and the JCR are held in their last solved state until the JCS is turned OFF. In this example, the output Y040 will maintain its current state when JCS is energized.

EXAMPLE CIRCUIT



{TON} ON DELAY TIMER

INPUT---[(A) TON (B)]--TIMER OUTPUT
DATA DISPLAY-> ***** *****

LEGAL ENTRIES

(A) TIME PRESET RW, XW, YW, ZW, D, or Numerical value
(B) TIMER REGISTER T only

OPERATION

When the input is ON, the timer decrements until the lapsed time is 0. When 0 is reached, the output turns ON and remains ON as long as the lapsed time is 0 and the input logic is ON. If the input turns OFF before the lapsed time reaches 0, the output remains OFF and the lapsed time is retained. The lapsed time is only reset back to the preset value when the input changes from OFF to ON.
*Lapsed time and preset values are in tenths of a second increments with a maximum value of 3276.7 seconds for timers T0 to T119.
*Lapsed time and preset values are in hundredths of a second increments with a maximum value of 327.67 seconds for timers T120 to T127.
*Timer lapsed time can be specified as retentive on power failure (See Section KEEP AREA TOP)

EXECUTION TIME (microseconds) Not Executed 1.8 Executed 96.0

MEMORY REQUIREMENTS For (A) = Numerical value: 2 steps For (A) = Register: 3 steps

EXAMPLE DESCRIPTION

When X000 is energized, timer T000 begins to time out. If X000 remains on for 10.0 seconds, the timer output will be turned on, energizing Y010. When X000 is OFF, T000's output is turned OFF. The timer output will remain OFF until X000 remains ON long enough for T000 to time out again.

EXAMPLE CIRCUIT

```
|
|--------| |----------[00100  TON   T000 ]--------------------------( )-
|      X000                                        Y010
```

TIMING DIAGRAM X = ON _ = OFF

X000 ___XXXXXXXXXX_____XXX_____XXXXXXXXXXX

T000 ___XXXxx_____XXX_____XXXxx_____

Y010 _____XXXXX_____XXXX

NOTE: The timer times from the preset to 0. If X000 turns OFF before the preset is reached, the timer stops timing. The timer is reset to the preset when X000 turns ON again.

{TOF} OFF DELAY TIMER

INPUT---[(A) TOF (B)]--TIMER OUTPUT
DATA DISPLAY-> ***** *****

LEGAL ENTRIES

(A) TIME PRESET RW, XW, YW, ZW, D, or Numerical value
(B) TIMER REGISTER T only

OPERATION

When the input is ON, The TOF (TIMER OFF) output provides circuit continuity for an ON output. When the input changes from ON to OFF, the TOF instruction simulates circuit continuity and holds its output ON until the lapsed time decrements to 0. When the lapsed time is 0, the output turns OFF. When the input changes from OFF to ON, the lapsed time is reset to the preset value and the TOF output turns ON.
*Lapsed time and preset values are in tenths of a second increments with a maximum value of 3276.7 seconds for timers T0 to T119.
*Lapsed time and preset values are in hundredths of a second increments with a maximum value of 327.67 seconds for timers T120 to T127.
*Timer lapsed time can be specified as retentive on power failure (See Section KEEP AREA TOP)

EXECUTION TIME (microseconds) Not Executed 1.8 Executed 96.0

MEMORY REQUIREMENTS For (A) = Numerical value: 2 steps For (A) = Register: 3 steps

EXAMPLE DESCRIPTION

When X000 is energized, the timer T001 passes power immediately to turn ON R010. When X000 is turned OFF, TOF begins to time out. If X000 remains OFF for more than 10.0 seconds, the output of TOF will turn OFF. When X000 is turned ON again, TOF will reset, energize R010, and be ready to time out again when X000 goes OFF.

EXAMPLE CIRCUIT:

```
|
|--------| |----------[00100  TOF   T001 ]--------------------------( )-
|      X000                                        R010
```

TIMING DIAGRAM X = ON _ = OFF

```
X000   ___XXXXX_______________XXXX_____XXXXXX

T001   ____XXXXXXXXXXxx_______XXXXXx___XXXXXX

R010   ____XXXXXXXXXXXX_______XXXXXXXXXXXXXX
```

NOTE: The Off Delay Timer turns ON as soon as X000 turns ON. It times down to its preset after X000 turns OFF. If X000 turns ON again before T1 times down to its preset, it is reset to the preset and starts timing down all over again.

{SS} SINGLE SHOT TIMER

```
      INPUT---[ (A)  SS  (B) ]--TIMER OUTPUT
DATA DISPLAY->  *****    *****
```

LEGAL ENTRIES

(A) TIME PRESET RW, XW, YW, ZW, D, or Numerical value
(B) TIMER REGISTER T only

OPERATION
When the input turns ON, the output turns ON and stays ON until either the input turns OFF or the lapsed time decrements to 0. If the input turns OFF before the lapsed time decrements to 0, the lapsed time is retained and is reset to the preset only when the input again turns ON.
* Lapsed time and preset values are in tenths of a second increments with a maximum value of 3276.7 seconds for timers T0 to T119.
* Lapsed time and preset values are in hundredths of a second increments with a maximum value of 327.67 seconds for timers T120 to T127.
* Timer lapsed time can be specified as retentive on power failure (See Section KEEP AREA TOP)

EXECUTION TIME (microseconds) Not Executed 1.8 Executed 96.0

MEMORY REQUIREMENTS For (A) = Numerical value: 3 steps For (A) = Register value: 2 steps

EXAMPLE DESCRIPTION

In the following example, when X000 is turned ON (and stays ON for more than the preset 10.0 seconds), the output of T002 is ON for 10.0 seconds, and then turns OFF until X000 turns OFF and back ON again. If X000 goes OFF before the 10.0 seconds expires, the output of T000 goes OFF and is reset. Each time X000 goes OFF, timer T002 is reset, and will begin timing for the total preset time when X000 comes ON again.

EXAMPLE CIRCUIT

```
|
|--------| |----------[00100   SS   T002 ]--------------------------( )-
|      X000                                        Z001
```

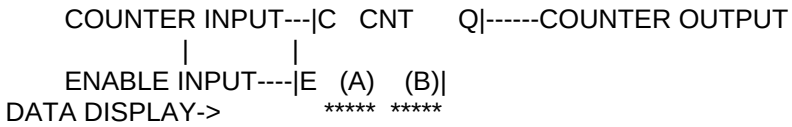
TIMING DIAGRAM X = ON _ = OFF X = ON _ = OFF

```
X000    ___XXXXXXXXXX_____________XXXXX

T002    ____XXXXXxx_________________XXXXXXXXXxx

Z001    ____XXXXXXXXXX_____________XXXXX________
```

{CNT} COUNTER



LEGAL ENTRIES

(A) COUNT PRESET RW, XW, YW, ZW, D, or Numerical value
(B) COUNTER REGISTER C only

OPERATION

This instruction counts pulses from the COUNTER INPUT if the logic preceding the ENABLE INPUT is ON. The accumulator (B) stores this count and is incremented every time the COUNTER INPUT makes the transition from OFF to ON. When the ENABLE INPUT turns OFF, the accumulator is reset to zero, and the output turns OFF. The largest legal counter preset is 65535.

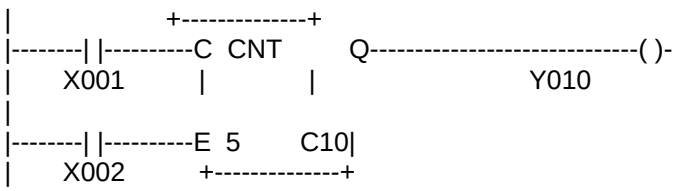
EXECUTION TIME (microseconds) Not Executed: 1.8 Executed: 92.0

MEMORY REQUIREMENTS For (A) = Numerical value: 3 steps For (A) = Register value: 4 steps

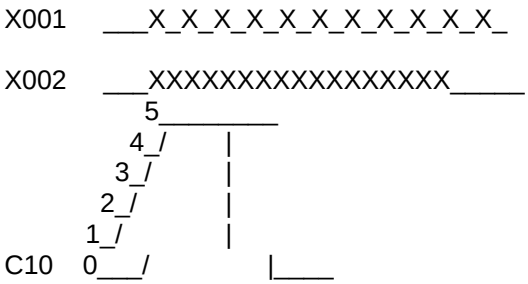
EXAMPLE DESCRIPTION

The change from OFF to ON at contact X001 (COUNTER INPUT) is counted only when contact X002 (ENABLE INPUT) is ON. The counted value is stored in C10. When the accumulated value reaches the preset value (00005), the output turns ON. Once the preset is reached and while the ENABLED input is ON, the output remains ON and the accumulated value is held at the preset. When contact X002 (ENABLE input) turns OFF, the accumulated value is zeroed and the output turns OFF. If contact X002 turns OFF during counting, the accumulated count is cleared and the output remains OFF.

EXAMPLE CIRCUIT



TIMING DIAGRAM X = ON _ = OFF



{END} END OF PROGRAM

|---[END]

OPERATION

One END instruction is required in each program. (Two or more are allowed for use as temporary END instructions in program debugging.). Programs are executed from the top left of the first page to the first END instruction. There may be additional instructions after the first END instruction, but they are NOT executed.

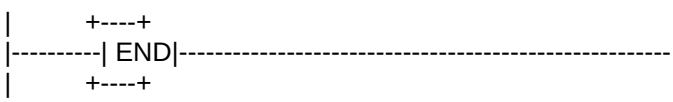
EXECUTION TIME (microseconds) 0.3

MEMORY REQUIREMENTS 1 step

EXAMPLE DESCRIPTION

The program will be executed up to the END instruction. Any program entry after that point will be skipped, and the scan will return to the beginning of the program.

EXAMPLE CIRCUIT



{FUN} {000} REGISTER TO REGISTER TRANSFER

INPUT---[(A) W-->W (B)]--EXECUTION OUTPUT
DATA DISPLAY-> ***** *****

LEGAL ENTRIES

(A) SOURCE REGISTER RW, XW, YW, ZW, D, T, C
(B) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input turns ON, the numerical data in the source register (A) is stored in the destination register (B) and the output turns ON.
* This transfer is made every scan the input is ON.
* For transfer, the source register must be compatible with the destination register.

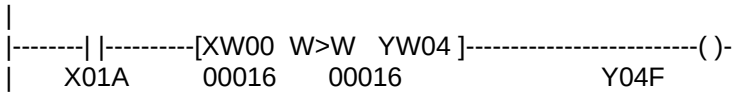
EXECUTION TIME (microseconds) Not Executed 1.8 Executed 98.0

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

When contact X01A turns ON, the contents of register XW00 (00016) are stored in register YW04 and the output turns ON. When contact X01A is OFF, there is no transfer and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

XW00 [00016] --> YW04 [00016]

{FUN} {001} CONSTANT TO REGISTER TRANSFER

INPUT---[(A) K --> W (B)]--EXECUTION OUTPUT
DATA DISPLAY-> *****

LEGAL ENTRIES

(A) CONSTANT Numerical value only
(B) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input turns ON, the numerical constant (A) is stored in the destination register (B), and the output turns ON.
* The largest legal constant entry is 65535.
* This transfer is made every scan the input is ON.
* The instruction does not execute and its output is OFF when the input is OFF.

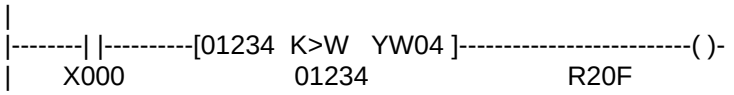
EXECUTION TIME (microseconds) Not Executed 1.8/2.5 Executed 93.0

MEMORY REQUIREMENTS For (A) < or = 255: 3 steps For (A) > or = 256: 4 steps

DESCRIPTION

When contact X000 turns ON, the numerical constant 01234 is stored in register YW04, and coil R20F is turned ON. When contact X000 if OFF, there is no transfer and coil R20F turns OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

Constant [01234] --> YW04 [01234]: R20F ON

{FUN} {002} TABLE INITIALIZATION

```
      INPUT---[ (A)  TINZ [nn]  (B) ]--EXECUTION OUTPUT
DATA DISPLAY->   *****          *****
```

LEGAL ENTRIES

(A) SOURCE REGISTER RW, XW, YW, ZW, D, T
(B) TABLE REGISTER RW, YW, ZW, D
(nn) TABLE LENGTH Numerical value only, 1 - 64

OPERATION

When the input is turned ON, the value in the source register is stored in all the registers of the table starting with (B) and with a table length of [nn].
* The maximum table length [nn] is 64 registers.
* The last register in the table is (B+nn-1).
* Source register (A) cannot be a number, it must be a register.
* This transfer is made every scan the input is ON.
* For transfer, the source register must be compatible with the registers in the table.

EXECUTION TIME (microseconds) Not Executed 2.5 Executed 98+5[nn]

MEMORY REQUIREMENTS 4 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, the value in register XW00 (00016) is transferred to the table (size = 64 registers). All 64 registers (D0100 to D0163) now have the same value as XW00 (00016) and coil Y04F turns OFF.

EXAMPLE CIRCUIT

```
|
|--------| |----------[XW00  TINZ   [64]   D0100]--------------------( )-
|      X01A         00016              00016                 Y04F
```

FUNCTION BLOCK

```
XW00 [00016]     |--->  D0100 [00016]
                 |--->  D0101 [00016]
                 |--->  D0102 [00016]
                       .       .
                  64 registers maximum
                       .       .
                 |--->  D0163 [00016]
```

{FUN} {003} TABLE TO REGISTER TRANSFER (Multiplexer)

```
      INPUT---[ (A)  T --> W [nn]  (B)  -->  (C) ]--TABLE LENGTH
DATA DISPLAY->  *****             *****     *****   ERROR
```

LEGAL ENTRIES

(A)	SOURCE TABLE	RW, XW, YW, ZW, D, T, C
(B)	TABLE POINTER	RW, XW, YW, ZW, D, T, C
(C)	DESTINATION REGISTER	RW, YW, ZW, D
(nn)	TABLE LENGTH	Numerical value only

OPERATION

When the input turns ON, the data content of one register within the source table (A) is transferred to the destination register (C). The source register, within the table, is determined by the numerical value of the table pointer (B). If the table pointer value is greater than the table length ([nn] minus one), then no transfer is made, and the output turns ON, indicating an error.
* Maximum table length is 64 registers.
* [nn] is the numerical value which sets the table length.
* The pointer considers the first register in the table to be location 0 (zero), and counts accordingly from that point. Therefore, if the table length [nn] is 50, then the maximum pointer value attainable without error would be 49.
* This transfer takes place every scan the input is ON.
* To transfer, the registers in the table must be compatible with the destination register.
* The source register is not altered by the transfer of data.

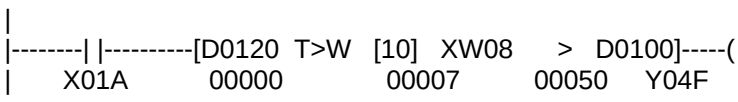
EXECUTION TIME (microseconds) Not executed 3.1 Executed 119.0

MEMORY REQUIREMENTS 5 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, the contents of register D0127 (pointer value is 7 therefore it is pointing to the eight position in the table) is stored in YW04. When data transfer is normal, coil Y04F is OFF. When the content of XW00 points to a register outside of the table (XW00 exceeds 9), the data is not transferred, and coil Y04F is turned ON. When contact X01A is OFF, the data is not transferred and the output is turned OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

TABLE POINTER

D0120 [00000] 0 YW04 [00050]
D0121 [xxxxx] 1 ^
D0122 [xxxxx] 2 |
. |
. |
. |
. |
D0127 [00050] 7---------->
D0128 [xxxxx] 8
D0129 [xxxxx] 9

{FUN} {003} TABLE TO REGISTER TRANSFER (Special EEPROM Write)

```
    INPUT---- (A)  T ---> W [nn]  (B)  --->  (C) ]--TABLE LENGTH
DATA DISPLAY-> *****              *****      *****   ERROR
```

LEGAL ENTRIES

(A)	SOURCE TABLE	RW, XW, YW, ZW, D, T, C
(B)	TABLE POINTER	RW, XW, YW, ZW, D, T, C
(C)	DESTINATION REGISTER	D512 + 16 x n (n=0,1,...63)
(nn)	TABLE LENGTH	Numerical value only

OPERATION

The EEPROM write instruction is a special mode of FUN 003 used to transfer the data of registers starting at (A) directly to the EEPROM. This instruction allows variable data to be permanently stored in the built-in EEPROM. It also transfers date into the data registers starting at (C).
* If nn is 64, register (C) can not be greater than D1472
* This instruction is valid only when the memory setting is set for the 3K mode
* This instruction functions by combining the transitional contact R62A and FUN003.
* The hex value H4000 must be in source register (B).
* Table size (number of registered to be transferred) can be specified by [nn] or by register (B) + 1 (Valid range 1-16)
* The destination registers should be set as retentive in the KEEP AREA TOP.

NORMAL OPERATION Execution output = OFF

ERROR CONDITION Execution output = ON

* Register size overflow error in register (A). Size of register (A) + (nn), or register (B) + 1 exceeded the register area
* Transfer data number error. The data in [nn] or (B) + 1 is 0 or exceeds 16.
* Setting error for register (C). The number in register (C) is not D512 + 16 x n (n = 0 to 63)
* Memory switch setting error. The memory setting switch in not in the 3K mode.
* Write protect error. The position of the key switch is set to RUN-P.
* EEPROM write error. Writing to the EEPROM did not complete normally or EEPROM data is undefined.

EXECUTION TIME (microseconds) Not Executed 3.1 Executed 119.0

MEMORY REQUIREMENTS 5 steps

EXAMPLE DESCRIPTION

When contact X000 turns ON, the contents of the 16 register table starting at register RW10 will be loaded into the internal EEPROM and transferred into a 16 register table starting at D512. The value in D100 must be H4000. For an additional example, see section 9.5.

EXAMPLE CIRCUIT

```
|
|--------| |-------|^|---------[RW10   T>W   [16]   D0100    >   D0512]
|       X000      R62A         xxxxx              H4000         xxxxx
```

FUNCTION BLOCK

```
                    DATA REGISTERS              EEPROM
RW10 [xxxxx]------------>  D512[xxxxx]------------>  [xxxxx]
RW11 [xxxxx]------------>  D513[xxxxx]------------>  [xxxxx]
       .         .                .
       .         .                .
       .         .                .
RW26 [xxxxx]              D528[xxxxx]------------>  [xxxxx]
```

{FUN} {003} TABLE TO REGISTER TRANSFER (Special Clock/Calendar Write)

```
     INPUT---- (A)  T ---> W [nn]  (B)  --->  (C) ]--TABLE LENGTH
DATA DISPLAY-> *****                *****     *****   ERROR
```

LEGAL ENTRIES

(A) SOURCE TABLE RW, XW, YW, ZW, D, T, C
(B) TABLE POINTER RW, XW, YW, ZW, D, T, C
(C) DESTINATION REGISTER D0005, Top register of the clock/calendar registers
(only D0005 is valid)
(nn) TABLE LENGTH Numerical value only

OPERATION

The clock-calendar data set instruction is a special mode of FUN 003. It is used to set the date/time value into the clock-calendar. The operation mode of FUN 003 is selected by changing bits E and F of Register (B) (Value in (B) should be H4000).

NORMAL OPERATION Execution output = OFF

ERROR CONDITION Execution output = ON

* Register size overflow error in register (A). Values in register (A) exceed allowable values for date and time.
* Transfer data number error. The number in [nn] is not 6.
* Register setting error in (C). Register (C) is not D0005.
* Calendar unmounted error. The CPU type does not support the clock-calendar function.
* Calendar is not specified. Set coil R628 ON, see special functions section.

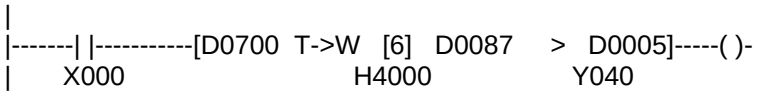
EXECUTION TIME (microseconds) Not Executed 3.1 Executed 119.0

MEMORY REQUIREMENTS 5 steps

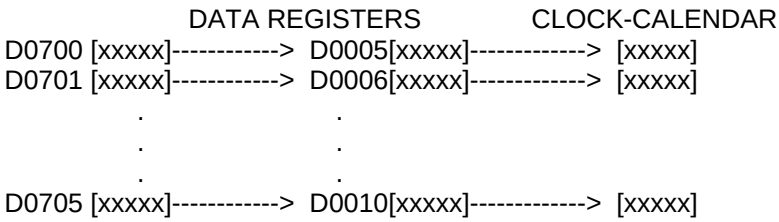
EXAMPLE DESCRIPTION

When the input X000 comes ON, the data of the 6 registers starting at D0700, are transferred to the clock-calendar registers, D0005 - D0010. The value in D0087 must be H4000. Y040 turns ON if a transmission error occurs.

EXAMPLE CIRCUIT



FUNCTION BLOCK



{FUN} {003} TABLE TO REGISTER TRANSFER (Data output instruction for intelligent I/O Modules)

```
      INPUT---[ (A)  T --> W [nn]  (B)  -->  (C) ]--TABLE LENGTH
DATA DISPLAY->  *****            *****     *****   ERROR
```

LEGAL ENTRIES

(A)	SOURCE TABLE	RW, XW, YW, ZW, D, T, C
(B)	OPERATION TYPE	RW, XW, YW, ZW, D, C
(C)	DESTINATION REGISTER	YW only (Output register assigned to module)
(nn)	TABLE LENGTH	Numerical value only, 1 - 64

OPERATION

The data output instruction for intelligent I/O modules is a special mode of FUN 003. It is used to transfer registers of data in the CPU to the intelligent modules, such as the motion control module. This operation mode of FUN 003 is selected by setting bit F ON in register (B)
* Table size equals [nn] or value in register(B) + 1. Max size = 128 unless computer link is being used. Then, maximum table size is 64 or less.
* Source table starts at (A).
* Destination table starts with the address specified by bits 0 to D in register (B), a register in the intelligent I/O module which is allocated to register (C).

NORMAL OPERATION Execution output = OFF

ERROR CONDITION Execution output = ON

* Register size overflow in register (A). Register (A), [nn], or (B) + 1 exceeded the maximum allowable register size.
* Transfer data number error. The data of [nn] or register (B) + 1 is 0 or is greater than 128.
* Overflow in register (B) + 1. When [nn] = 64, register (B) is the highest number register allowed.
* Internal memory error in the intelligent I/O module. The value of bits 0 to D of register (B) is not in the range of 0 to 350.
* Output register error. Register (C) is not a YW register
* Intelligent I/O Module error. The destination register, YW, is not assigned to an intelligent module.
* I/O Request error. The intelligent module is no responding.
* I/O Dismounted error. The intelligent I/O module is in the I/O rack. This will put the PLC in the error mode.
* I/O Response error. The operation of the intelligent I/O module is abnormal. This will put the PLC into the error mode.
* I/O Parity error. The operation of the intelligent I/O module is abnormal. This will put the PLC into the error mode.

EXECUTION TIME (microseconds) Not Executed 3.1 Executed 119.0

MEMORY REQUIREMENTS 5 steps

EXAMPLE DESCRIPTION

When input X000 turns ON, the 6 values in D0700 thru D0705 are transferred via YW11 to the 6 registers in the intelligent I/O module. Register D0005 in the intelligent I/O module defines the destination table for these six values. Y040 turns ON if a transmission error occurs.

EXAMPLE CIRCUIT

```
|
|--------| |---------[D0700  T->W   [6]   D0005    >   YW11 ]-----( )-
|       X000                                          Y040
```

FUNCTION BLOCK

```
PLC TABLE                PLC REGISTER              I/O MODULE TABLE
D0700[xxxxx]------------> YW11[xxxxx]---------------> D0006[xxxxx]
D0701[xxxxx]    |                    |         D0007[xxxxx]
     .          |                    |              .
     .          |                    |              .
     .          |                    |              .
D0705[xxxxx]---                       -------> D0011[xxxxx]
```

FEDCBA9876543210
D0006 must contain the value: [1000000000000110]
Bits 0 thru A give the value 6, therefore the first register address in the intelligent I/O module is D0011.

{FUN} { 004} REGISTER TO TABLE TRANSFER (Demultiplexer)

```
      INPUT---[ (A)  W-->T [nn]  (B)   (C) ]--TABLE LENGTH ERROR
DATA DISPLAY->  *****           ***** *****   OUTPUT
```

LEGAL ENTRIES

(A) SOURCE REGISTER RW, XW, YW, ZW, D, T, C
(B) TABLE POINTER RW, XW, YW, ZW, D, T, C
(C) DESTINATION TABLE RW, YW, ZW, D
(nn) TABLE LENGTH Numerical value only

OPERATION

When the input turns ON, the data in the source register (A) is transferred into the destination table. The destination register in the table is determined by the numerical value of the pointer (B). If the table pointer value is greater than the table length ([nn] minus one), then no transfer is made, and the output turns ON, indicating an error.
* Maximum table length is 64 registers.
* [nn] is the numerical value which sets the table length.
* The pointer considers the first register in the table to be location 0 (zero), and counts accordingly from that point. Therefore, if the table length [nn] is 50, then the maximum pointer value attainable without error would be 49.
* This transfer takes place every scan the input is ON.
* To transfer, the registers in the table must be compatible with the destination register.
* The source register is not altered by the transfer.

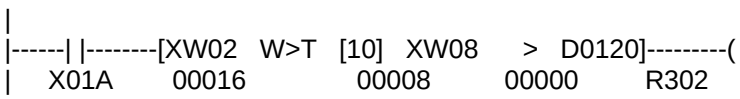
EXECUTION TIME (microseconds) Not Executed 3.1 Executed 119.0

MEMORY REQUIREMENTS 5 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, the content of register XW02 [00016] is stored in register D0128, which is the ninth register (since the content of pointer XW08 is 8) in a 10 register table. When the content of XW08 points to a register outside of the table (exceeding 9 in the example), data transfer does not occur and R302 is turned ON. When contact X01A is OFF, data transfer does not take place, and R302 turns OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

SOURCE REGISTER	DESTINATION REGISTER	POINTER
XW02 [00016]	D0120 [00000]	0
|	D0121 [xxxxx]	1
|	.	.
|	.	.
|	.	.
|		
-------->	D0128 [00016]	8
	D0129 [xxxxx]	9

{FUN} { 004} REGISTER TO TABLE TRANSFER (Special EEPROM Read; EEPROM --> Register)

LEGAL ENTRIES

(A) SOURCE REGISTER D0512-D1535
(B) TABLE POINTER RW, XW, YW, ZW, D, T, C
(C) OPERATION TYPE RW, YW, ZW, D
(nn) TABLE LENGTH 1-16 or 64

OPERATION

The EEPROM read instruction is a special mode of FUN 004 used to transfer the date in EEPROM into the RAM registers. This instruction allows the RAM registers to be re-initialized from data permanently stored in the EEPROM. Data is stored in the RAM registers starting at (C) and also in the RAM register corresponding to (A).
* The hex value H4000 must be in the table pointer (B)
* The source table starts at register (A) in the EEPROM (D0512-D1535)
* Table length is specified by [nn] or by the value in the register (B)+1. If [nn] is 64, the largest register address for (A) is D1472.
* The destination table in RAM starts at the location specified by (C). It also loads the EEPROM data in the corresponding RAM registers corresponding to (A).
* This instruction is valid only when the memory setting is set for the 3K mode.

NORMAL OPERATION Execution output = OFF

ERROR CONDITION Execution output = ON

* Register size overflow error in register (A). The size of register (A) + [nn] (or the value in register (B) +1) exceeded the maximum allowable register area.
* Transfer data number error. The value in [nn] or in the register (B) + 1 is 0 or greater than 16.
* Table length overflow in (B) + 1. The value in register (B) + 1 is greater than 64.
* Table length overflow in (C). The length of (C) is less than [nn] or (B) + 1.
* Memory setting error. The memory setting switch is not set for the 3K mode.

EXECUTION TIME (microseconds) Not Executed 3.1 Executed 119.0

MEMORY REQUIREMENTS 5 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, 16 registers in the EEPROM starting at D0512 [00016] are stored in the RAM table starting at RW20. When contact X01A is OFF, data transfer does not take place. RW00 must contain the hex value H4000.

EXAMPLE CIRCUIT

```
|
|------| |--------[D0512   W>T   [16]   RW00      >   RW20]---------( )-
|    X01A                                          R302
```

FUNCTION BLOCK

```
SOURCE              DESTINATION        DESTINATION
REGISTER (EEPROM)   REGISTER (C)       REGISTER (A) (RAM)
D0512 [xxxxx]  -->   RW20 [xxxxx]       D0512 [xxxxx]
D0513 [xxxxx]        RW21 [xxxxx]       D0513 [xxxxx]
       .             .                  .
       .             .                  .
       .             .                  .
       .
D0526 [xxxxx]        RW34 [xxxxx]      D0526 [xxxxx]
D0527 [xxxxx] -->    RW35 [xxxxx]      D0527 [xxxxx]
```

{FUN} { 004} REGISTER TO TABLE TRANSFER (Special read instruction for intelligent I/O modules.)

LEGAL ENTRIES

(A) SOURCE I/O REGISTER XW (Input register assigned to module)
(B) OPERATION TYPE/SOURCE TABLE RW, XW, ZW, D, C
(C) DESTINATION TABLE RW, YW, ZW, D
(nn) TABLE LENGTH Numerical value only, 1 - 64

OPERATION

The data input instruction for intelligent I/O modules is a special mode of FUN 004. It is used to transfer data from the intelligent I/O modules, such as the ASCII Basic module, to the CPU registers (RAM). This operation mode of FUN 004 is selected by setting the value hex 4000 (bits E=0 and F=1) in register (B).
* Table size equals [nn] or value in register(B) + 1. Max size = 128 unless computer link is being used. Then, maximum table size is 64 or less.
* Source table starts at (A).
* Destination table starts with the address specified by bits 0 to D in register (B), a register in the intelligent I/O module which is allocated to register (C).

NORMAL OPERATION Execution output = OFF

ERROR CONDITION Execution output = ON

* Register size overflow in register (A). Register (A), [nn], or (B) + 1 exceeded the maximum allowable register size.
* Transfer data number error. The data of [nn] or register (B) + 1 is 0 or is greater than 128.
* Overflow in register (B) + 1. When [nn] = 64, register (B) is the highest number register allowed.
* Internal memory error in the intelligent I/O module. The value of bits 0 to D of register (B) is not in the range of 0 to 350.
* Output register error. Register (C) is not a YW register
* Intelligent I/O Module error. The destination register, YW, is not assigned to an intelligent module.
* I/O Request error. The intelligent module is not responding.
* I/O Dismounted error. The intelligent I/O module is not in the I/O rack. This will put the PLC in the error mode.
* I/O Response error. The operation of the intelligent I/O module is abnormal. This will put the PLC into the error mode.
* I/O Parity error. The operation of the intelligent I/O module is abnormal. This will put the PLC into the error mode.

EXECUTION TIME (microseconds) Not Executed 3.1 Executed 119.0

MEMORY REQUIREMENTS 5 steps

EXAMPLE DESCRIPTION

When input X000 turns ON, the 6 values in the intelligent I/O module registers D0011 thru D0016 are transferred via XW06 to 6 registers in the CPU, D0050 thru D0055. Register D0005 in the intelligent I/O module defines the destination table for these six values. Y040 turns ON if a transmission error occurs.

EXAMPLE CIRCUIT

```
|
|--------| |---------[XW06  T->W   [6]   D0010    >   D0050] ]-----( )-
|       X000                                          Y040
```

FUNCTION BLOCK

```
I/O MODULE TABLE          PLC I/O REGISTER          CPU MODULE TABLE
D0011[xxxxx]-------------> XW06[xxxxx]-------------> D0050[xxxxx]
D0012[xxxxx]    |                    |          D0051[xxxxx]
     .          |                    |             .
     .          |                    |             .
     .          |                    |             .
D0016[xxxxx]----                      ------> D0055[xxxxx]
```

FEDCBA9876543210
D0010 must contain the value: [1000000000001011]
Bits 0 thru A give the value 11, therefore the first register address in the intelligent I/O module is D0011.

{FUN} {005} TABLE TO TABLE TRANSFER

```
      INPUT---[ (A)  T-->T  [nn]  (B) ]--EXECUTION OUTPUT
DATA DISPLAY->  *****            *****
```

LEGAL ENTRIES

(A) SOURCE TABLE RW, XW, YW, ZW, D, T, C
(B) DESTINATION TABLE RW, YW, ZW, D
[nn] TABLE LENGTH Numerical value only

OPERATION

When the input turns ON, all data in the source table is transferred to corresponding locations in the destination table.

* Maximum table length is 64 registers.
* [nn] is the numerical value which sets the table length.
* This transfer takes place on every scan the input is ON.
* The source table is not altered by the data transfer.
* To transfer, the registers in the table must be compatible with each other.

EXECUTION TIME (microseconds) NOT EXECUTED 2.5 EXECUTED 105+11[nn]

MEMORY REQUIREMENTS 4 steps

EXAMPLE DESCRIPTION
When contact X000 is ON, the data from 7 registers starting with register D0100 are block transferred to D0108 and 7 subsequent registers, and the output turns ON (R302). When contact X000 is OFF, transfer does not take place and the output R302 is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[D0100  T>T   [07]   D0108]---------------------( )
¦      X000          00005             00005                 R302
```

FUNCTION BLOCK

```
SOURCE TABLE        DESTINATION TABLE

D0100 [00005]  ---->   D0108 [00005]
D0101 [00010]  ---->   D0109 [00010]
      .                .
      .                .        7 registers
      .                .
D0106 [00060]  ---->   D0106 [00060]
```

7.4 Arithmetic Operations

{FUN} {010} REGISTER + REGISTER ADDITION

```
      INPUT---[ (A)  +  (B)  -->  (C) ]--OVERFLOW OUTPUT
DATA DISPLAY->  *****   *****     *****
```

LEGAL ENTRIES

(A) AGUEND REGISTER RW, XW, YW, ZW, D, T, C
(B) ADDEND REGISTER RW, XW, YW, ZW, D, T, C
(C) SUM REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, augend (A) is added to addend (B) and the sum is stored in (C). If the resultant sum is greater than the register's ability to store it (65535), the output turns ON, indicating an overflow. When overflow occurs, the sum register (C) will store the value 65535.
* Since the instruction is REGISTER addition, numerical values are not acceptable.
* Operands (A), (B), and (C) may be the same register.

EXECUTION TIME (microseconds) NOT EXECUTED 2.5 EXECUTED 110.0

MEMORY REQUIREMENTS 4 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, the content of register XW00 (00002) is added to XW02 (00004) and the sum (00006) is stored in register YW04. Since there is no overflow, the output is OFF (R14F). If an overflow takes place (the sum exceeds 65535), the limit value of 65535 is stored in YW04 and the output turns ON. When contact X01A is OFF, addition is not executed and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[XW00    +   XW02     >   YW04 ]------------( )-
|      X01A         00002       00004        00006            R14F
```

FUNCTION BLOCK

```
XW00 [00002]  +  XW02 [00004]  =  YW04 [00006] : R14F OFF   Normal
                                  YW04 [65535] : R14F ON    0verflow
```

{FUN} {011} REGISTER - REGISTER SUBTRACTION

```
      INPUT---[ (A)  -  (B)  -->  (C) ]--UNDERFLOW
DATA DISPLAY->  *****   *****     *****
```

LEGAL ENTRIES

(A) MINUEND REGISTER RW, XW, YW, ZW, D, T, C
(B) SUBTRAHEND REGISTER RW, XW, YW, ZW, D, T, C
(C) DIFFERENCE REGISTER RW, YW, ZW, D

OPERATION

When the input turns ON, subtrahend (B) is subtracted from minuend (A) and the difference is stored in (C). If the difference is less than zero, the output is turned ON, indicating an underflow, and the absolute value is stored in register (C).
* This instruction is REGISTER subtraction; numerical values are not acceptable.
* Operands (A), (B), and (C) may be the same register.

EXECUTION TIME (microseconds) Not Executed 2.5 Executed 110.0

MEMORY REQUIREMENTS 4 steps

EXAMPLE DESCRIPTION

When contact R01A is ON, the content of register XW02 (00004) is subtracted from that of XW00 (00016) and the difference (00012) is stored in YW04 The output is OFF if there is no underflow. If underflow takes place, the value zero is stored in register YW04 and the output turns ON. When contact R01A is OFF, subtraction is not executed and the outputs OFF.

EXAMPLE CIRCUIT

```
|
|--------| |----------[XW00    -   XW02     >   YW04 ]------------( )-
|      R01A         00016       00004        00012            R15F
```

FUNCTION BLOCK

```
XW00 [00016]  -  XW02 [00004]  =  YW04 [00012] : R15F OFF Normal
                                  =  YW04 [xxxxx] : R15F ON
```
When R15F is on, YW04 contains the absolute value of the subtraction.

{FUN} {012} REGISTER x REGISTER MULTIPLICATION

INPUT---[(A) x (B) -> -> (C)]--EXECUTION OUTPUT
DATA DISPLAY-> *****   *****   ***** *****

LEGAL ENTRIES

(A) MULTIPLICAND REGISTER   RW, XW, YW, ZW, D, T, C
(B) MULTIPLIER REGISTER   RW, XW, YW, ZW, D, T, C
(C) PRODUCT REGISTER   RW, YW, ZW, D

OPERATION

When the input is ON, the multiplicand (A) is multiplied by the multiplier (B), and the product is stored in double precision register (C). When double precision registers are used in any function, each operand is actually a pair of two consecutive registers, with the first register of each pair being the only one displayed on the screen. These registers are used as a single 32 bit value, and the value displayed is only a portion of the true total value for the operand.
* This instruction is register multiplication. Numerical values are not acceptable.

EXECUTION TIME (microseconds)   Not Executed 2.5   Executed 168.0

MEMORY REQUIREMENTS   4 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, register XW00 (1000) is multiplied by register D100 (200). The product is stored in successive registers, YW04 and YW05, the most significant bits in YW04 and the least significant bits in YW05. The output turns ON. If contact X01A is OFF, operation is not executed and the output turns OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[XW00     *   D0100   > >   YW04 ]------------( )-
|       X01A          01000        00200        00003             R15F
```

FUNCTION BLOCK

XW01 [01000] x D100 [00200] = YW04 [00003], YW05 [3395]

= [(3 x 65535) + 3395]

= 200,000

YW04 [00003], YW05 [3595] : R15F ON, Normal

{FUN} {013} REGISTER / REGISTER DIVISION

```
      INPUT---[ (A)  /  (B)  -->  (C)  ]--OVERFLOW OUTPUT
DATA DISPLAY->  *****   *****     *****
```

LEGAL ENTRIES

(A) DIVIDEND REGISTER RW, XW, YW, ZW, D, T, C
(B) DIVISOR REGISTER RW, XW, YW, ZW, D, T, C
(C) QUOTIENT REGISTER RW, YW, ZW, D

OPERATION

Dividend register (A) is a double precision register. When double precision registers are used in any function, each operand is actually a pair of two consecutive registers, with the first register of each pair being the only one displayed on the screen. These registers are used as a single 32 bit value, and the value displayed is only a portion of the true total value for the operand. When the input turns ON, the dividend (A), is divided by the divisor (B), and the quotient is stored in (C), with the remainder in (C+1). If the quotient is greater than the register's ability to store it (65535), the output turns on (indicating overflow), the quotient is set to 65535, and the remainder is set to zero.
* This instruction is register division, numerical values are not accepted.
* The dividend can be the product of a multiplication or the results of a double precision addition/subtraction.

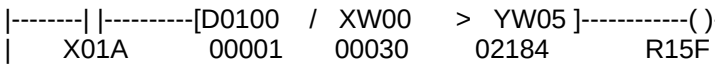
EXECUTION TIME (microseconds) Not Executed 2.5 Executed 342.0

MEMORY REQUIREMENTS 4 Steps

EXAMPLE DESCRIPTION

When contact X01A is ON, the content of successive registers D0100 (00001) and D0101 (00001) are divided by that of register XW00 (30). The quotient (02184) is stored in YW05, and the remainder (16), in YW06. The output is OFF. If and overflow (including divisor being 0) exists, the quotient is set at 65535, the remainder is set at 0, and the output is turned ON. If contact X000 is OFF, operation is not executed and the output turns OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

D0100 [00001], D0101 [00001] / XW00 [00030] = YW05 [02184], YW06 [00016]

[(1 x 65535) + 1] / 30 = 2184 R16

YW05 [02184], YW06 [00016] : R15F OFF - Normal
YW05 [65535], YW06 [00000] : R15F ON - Overflow
R = Remainder

{FUN} {014} REGISTER GREATER THAN REGISTER COMPARISON

```
     INPUT---[  (A)  >  (B)  ]--DECISION OUTPUT
DATA DISPLAY->   *****   *****
```

LEGAL ENTRIES

(A) COMPARED REGISTER RW, XW, YW, ZW, D, T, C
(B) SOURCE REGISTER RW, XW, YW, ZW, D, T, C

OPERATION

When the input turns ON, the value in register (A) is compared to the value in register (B). If the results of this comparison show register (A) to be greater than register (B), then the output turns ON.
* This instruction is register comparison; numerical values are not acceptable.

EXECUTION TIME (microseconds) Not Executed 1.8 Executed 100.0

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

When contact R01A is ON, the content of register XW00 (00018) is compared with that of XW02 (00004). Since XW00 is greater than XW00, the output turns ON. If XW00 were less than, or equal to XW02, the output would be OFF. When contact X01A is OFF, comparison is not executed and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[XW00   W>W   XW02 ]--------------------------( )-
|      R01A         00018       00004                        R10F
```

FUNCTION BLOCK

```
XW00 [00018]  >  XW02 [00004]  -->  YES: R10F ON
                                    NO:  R10F OFF
```

{FUN} {015} REGISTER EQUAL TO REGISTER COMPARISON

```
      INPUT---[ (A)  =  (B)  ]--DECISION OUTPUT
DATA DISPLAY->  *****   *****
```

LEGAL ENTRIES

(A) COMPARED REGISTER RW, XW, YW, ZW, D, T, C
(B) SOURCE REGISTER RW, XW, YW, ZW, D, T, C

OPERATION

When the input turns ON, the value in register (A) is compared to the value in register (B). If register (A) is equal to register (B), then the output turns ON.
* This instruction is a register comparison; numerical values are unacceptable.

EXECUTION TIME (microseconds) Not Executed 1.8 Executed 100.0

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

When contact R01A is ON, the content of register XW00 (00008) is compared to that of register XW02 (00008). Since XW02 equals XW00, the input turns ON. If XW00 were not equal to XW00, the output would turn OFF. When contact R01A is OFF, comparison is not executed and the output is OFF

EXAMPLE CIRCUIT

```
|--------| |----------[XW00   =   XW02 ]-----------------------------( )-
|      R01A          00008      00008                          R10F
```

FUNCTION BLOCK

```
XW00 [00008]  =  XW02 [00008]  -->  YES: R10F ON
                                    NO:  R10F OFF
```

{FUN} {016} REGISTER LESS THAN REGISTER COMPARISON

```
      INPUT---[ (A)  <  (B)  ]--DECISION OUTPUT
DATA DISPLAY->  *****   *****
```

LEGAL ENTRIES

(A) COMPARED REGISTER RW, XW, YW, ZW, D, T, C
(B) SOURCE REGISTER RW, XW, YW, ZW, D, T, C

OPERATION

When the input turns ON, the value in register (A) is compared to the value in register (B). If the results show that register (A) is less register (B), then the output turns ON.
* This instruction is a register comparison; numerical values are unacceptable.

EXECUTION TIME (microseconds) Not Executed 1.8 Executed 100.0

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, the content of register XW00 (00004) is compared with that of XW02 (00008). Since XW00 is less than XW02, the output turns ON. If XW00 were greater than, or equal to, XW02, the output would turn OFF. When contact X01A is OFF, the comparison is not executed, and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[XWOO   <   XW02 ]-----------------------------( )-
|      X01A          00004      00008                          R10F
```

FUNCTION BLOCK

```
XW00 [00004]  <  XW02  [00008]  --->  YES: R10F ON
                                 NO:  R10F OFF
```

{FUN} {017} REGISTER + REGISTER ADDITION -- DOUBLE LENGTH REGISTERS

```
      INPUT---[ (A)  ++  (B) -> ->  (C)  ]--OVERFLOW OUTPUT
DATA DISPLAY->  *****    *****      *****
```

LEGAL ENTRIES

(A) AUGEND REGISTER RW, XW, YW, ZW, D
(B) ADDEND REGISTER RW, XW, YW, ZW, D
(C) SUM REGISTER RW, YW, ZW, D

OPERATION
(A), (B), and (C) are double precision registers. When double precision registers are used in any function, each operand is actually a pair of two consecutive registers, with the first register of each pair being the only one displayed on the screen. These registers are used as a single 32 bit value, and the value displayed is only a portion of the true total value for the operand. When the input is ON, the augend registers are added to the addend registers and the result is stored in the SUM registers. If the sum is greater than double precision sum register (C)'s ability to store it, the output turns ON, indicating an overflow. When an overflow occurs, the value 65535 is stored in double precision register (C).
* This instruction is register addition; numerical values are unacceptable.
* (A) and (A)+1, (B) and (B)+1, and (C) and (C)+1 are considered 32 bit double length registers.
* (A), (B), and (C) may be even or odd registers.

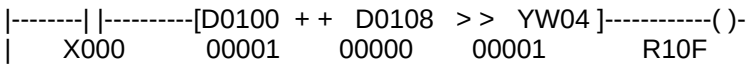
EXECUTION TIME (microseconds) Not Executed 2.5 Executed 125.0

MEMORY REQUIREMENTS 4 steps

EXAMPLE DESCRIPTION

When contact X000 is ON, contents of double length register D100 (70,000) is added to the contents of double length register D108 (20,000), the sum is stored in double length register YW10 (90,000), and the output is OFF. If an overflow occurs, the limit value of FFFFFFFF (hex) is stored in double length register YW04 and the output is turned ON. When contact X000 is Off, the operation is not executed and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

D100 [00001], D102 [04465] + D108 [00000], D109 [02000]
= YW04 [00001], YW05 [24465]

= [(1 x 65535) + 4465)] + [(0 x 65535) + 20000] = [(1 x 65535) + 24465]
= 70,000 + 20,000 = 90,000

YW04 [00001], YW06 [24465] : R10F OFF - Normal
YW04 [65535], YW06 [65535] : R10F ON - Overflow

{FUN} {018} REGISTER - REGISTER SUBTRACTION -- DOUBLE LENGTH REGISTERS

```
     INPUT---[ (A)  --  (B)  -> ->  (C)  ]--UNDERFLOW OUTPUT
DATA DISPLAY->  *****    *****       *****
```

LEGAL ENTRIES

(A) MINUEND REGISTER RW, XW, YW, ZW, D
(B) SUBTRAHEND REGISTER RW, XW, YW, ZW, D
(C) DIFFERENCE REGISTER RW, YW, ZW, D

OPERATION

(A), (B), and (C) are double precision registers. When double precision registers are used in any function, the operand is actually a pair of two consecutive registers, with the first register of each pair being the only one displayed on the screen. The registers are used as a single 32 bit value, and the value displayed is only a portion of the true total value for the operand. When the input is ON, the minuend register is subtracted from the subtrahend register and the result is stored in the difference register. If the difference is less than zero, the output is turned ON indicating underflow has occurred, and double precision register (C) is set to zero.
* This instruction is register subtraction; numerical values are unacceptable.
* (A) and (A)+1, (B) and (B)+1, and (C) and (C)+1 are considered 32 bit double length registers.
* (A), (B), and (C) may be even or odd registers.

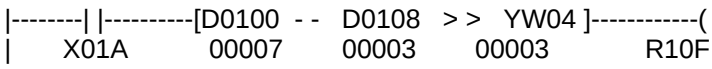
EXECUTION TIME (microseconds) Not Executed 2.5 Executed 130.0

MEMORY REQUIREMENTS 4 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, the contents of double precision register D108 (300,000) is subtracted from the contents of double precision register D100 (500,000) and the difference is stored in double precision register YW04 (200,000). The output is OFF. If and underflow occurs, the value 0 is stored in register (C) and the output turns ON. When contact X01A is OFF, the operation is not executed, and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

D100 [00007], D102 [41255] - D108 [00003], D109 [37860]
= YW04 [00003], YW05 [03395]

[(7 x 65535) + 41225] - [(3 x 65535) + 37860] = [(3 x 65535) + 3395]
= 500,000 - 300,000 = 200,000

YW04 [00003], YW05 [03395] : R10F OFF - Normal
YW04 [absolute], YW05 [value] : R10F ON - Underflow

{FUN} {020} REGISTER + CONSTANT ADDITION

```
      INPUT---[ (A)  +  (B)  -->  (C)  ]--OVERFLOW OUTPUT
DATA DISPLAY->  *****            *****
```

LEGAL ENTRIES

(A) AUGEND REGISTER RW, XW, YW, ZW, D, T, C
(B) ADDEND CONSTANT Numerical value only
(C) SUM REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, augend (A) is added to addend constant (B) and the sum is stored in (C). If the resultant sum is greater than the register's ability to store it (65535), the output turns ON, indicating an overflow. When overflow occurs, register (C) will store the value 65535.
* Operands (A) and (C) may be the same register; D100 + 1 -> D100.
* Operand (B) can not be a register, only a constant.

EXECUTION TIME (microseconds) Not Executed 2.5/3.1 Executed 110.0

MEMORY REQUIREMENTS

For (B) < or = 255 : 4 steps
For (B) < or = 256 : 5 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, the content of register XW00 (00002) is added to a constant (00004), and the sum (00006) is stored in register YW05. Since there is no overflow, output R10F is OFF. If there is an overflow, the limit value (65535) is stored in register (C), and the output turns ON. When contact X01A is OFF, addition is not executed and the output turns OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[XW00   + .   00004     >   YW05 ]------------( )-
|      X01A         00002                  00006           R14F
```

FUNCTION BLOCK

```
XW00 [00002] + 00004  =  YW05 [00006]: R14F  OFF  Normal
                         YW05 [65535]: R14F  ON   Overflow
```

{FUN} {021} REGISTER - CONSTANT SUBTRACTION

INPUT---[(A) - (B) --> (C)]--UNDERFLOW OUTPUT
DATA DISPLAY-> ***** *****

LEGAL ENTRIES

(A) MINUEND REGISTER RW, XW, YW, ZW, D, T, C
(B) SUBTRAHEND CONSTANT Numerical value only
(C) DIFFERENCE REGISTER RW, YW, ZW, D

OPERATION

When the input turns ON, subtrahend constant (B) is subtracted from minuend (A) and the difference is stored in (C). If the difference is less than zero, then the output is turned ON, indicating underflow, and the absolute value is stored in the difference register (C).
* Register (A) and (C) maybe the same register; D100 - 1 --> D100
* Operand (B) can not be a register, only a constant

EXECUTION TIME (microseconds) Not Executed 2.5/3.1 Executed 113.0

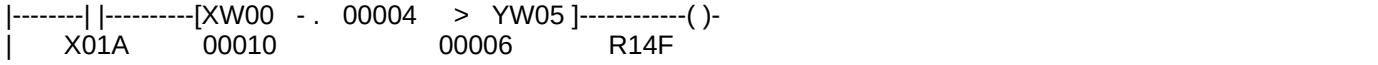
MEMORY REQUIREMENTS

For (B) < or = 255 : 4 steps
For (B) > or = 256 : 5 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, constant 0004 is subtracted from the contents of register XW00 (00010), and the difference (00006) is stored in register YW05. Since there is no underflow, output R14F is OFF. If and underflow takes place during operation, the absolute value is stored in register (C), and output R14F turns ON. When contact X01A is OFF, subtraction is not executed and the output turns OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

XW00 [00010] - 00004 = YW05 [00006]: R14F OFF Normal
YW05 [< 0]: R14F ON Underflow

{FUN} {022} REGISTER x CONSTANT MULTIPLICATION

INPUT---[(A) x (B) --> --> (C)]--EXECUTION OUTPUT
DATA DISPLAY-> ***** *****

LEGAL ENTRIES

(A) MULTIPLICAND REGISTER RW, XW, YW, ZW, D, T, C
(B) MULTIPLIER CONSTANT Numerical Value Only
(C) PRODUCT REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, the multiplicand (A) is multiplied by the multiplier constant (B), and the product is stored in the double precision register (C). When double precision registers are used in any function, each operand is actually a pair of two consecutive registers, with the first register of each pair being the only one displayed on the screen. These registers are used as a single 32 bit value, and the value displayed is only a portion of the true total value for the operand.
* Operand (B) can not be a register, only a constant
* Operand (C) is a double length register, (C) and (C)+1.

EXECUTION TIME (microseconds) Not Executed 2.5/3.1 Executed 167.0

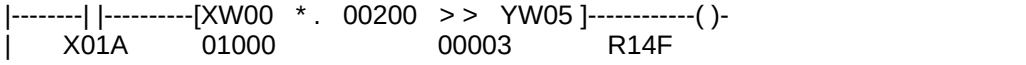
MEMORY REQUIREMENTS

For (B) < or = 255 : 4 steps
For (B) > or = 256 : 5 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, register XW00 (1000) is multiplied by the constant (00200). The product is stored in double precision registers (YW05 and YW06) and the output turns ON. If contact X01A is OFF, operation is not executed and the output turns OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

XW00 [01000] x 00200 = YW05 [00003], YW06 [03395]

1000 x 200 = [(3 x 65535) + 3395 = 200,000

{FUN} {023} REGISTER / CONSTANT DIVISION

```
     INPUT---[ (A)  /  (B)  -->  (C) ]--OVERFLOW OUTPUT
DATA DISPLAY->  *****            *****
```

LEGAL ENTRIES

(A)	DIVIDEND REGISTER	RW, XW, YW, ZW, D, T, C
(B)	DIVISOR CONSTANT	Numerical value only
(C)	QUOTIENT REGISTER	RW, YW, ZW, D

OPERATION

Register (A) is a double precision register. Register (C) is two registers, the first, the displayed register, contains the quotient. The remainder is in the next consecutive register. When double precision registers are used in any function, each operand is actually a pair of two consecutive registers, with the first register of each pair being the only one displayed on the screen. These registers are used as a single 32 bit value, and the value displayed is only a portion of the true total value for the operand.
* Operand (B) can not be a register, only a constant.
* The dividend is a double length register, (A) and (A)+1.
* The dividend can be the product of multiplication or the result of a double precision addition/subtraction.

EXECUTION TIME (microseconds) Not Executed 2.5/3.1 Executed 343.0

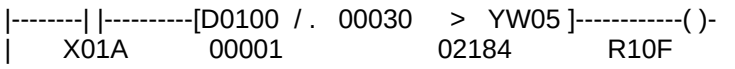
MEMORY REQUIREMENTS

For (B) < or = 255 : 4 steps
For (B) > or = 256 : 5 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, the content of successive registers D0100 (00001) and D0102 (00001) are divided by the constant (30). The quotient (02184) is stored in YW05, and the remainder (16) in YW06. The output will be OFF. If an overflow (including divisor being 0) exists, the quotient (YW05) is set to 65535, the remainder (YW06) is set to 0, and the output is turned ON. If contact X01A is OFF, operation is not executed, and the output is turned OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

D0100 [00001], D0101 [00000] / 30 = YW05 [02184], YW06 [00016]

[(1 x 65635) + 0] = 2184 R10F = 65535 / 30 = 2184 R10F

YW05 [02184], YW06 [00016] : R10F OFF - Normal
YW05 [65535], YW06 [00000] : R10F ON - Overflow

{FUN} {024} REGISTER GREATER THAN CONSTANT COMPARISON

```
      INPUT---[ (A)  >  (B) ]--DECISION OUTPUT
DATA DISPLAY->  *****
```

LEGAL ENTRIES

(A) COMPARED REGISTER RW, XW, YW, ZW, D, T, C
(B) CONSTANT Numerical value only

OPERATION

When the input turns ON, the value in register (A) is compared to the constant (B). If register (A) is greater than constant (B), the output turns ON.

EXECUTION TIME (microseconds) Not Executed 1.8/2.5 Executed 98.0

MEMORY REQUIREMENTS

For (B) < or = 255 : 3 steps
For (B) > or = 256 : 4 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, the content of register XW00 (01000) is compared with that of the constant (00100). If XW00 is greater than the constant, the output turns ON. If XW00 is less than or equal to the constant, the output turns OFF. When contact X01A is OFF, comparison is not executed and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[XW00    > .  00100]--------------------------( )-
|       X01A          01000                                  R10F
```

FUNCTION BLOCK

```
XW00 [01000] >   100  -->  YES: R10F ON
                           NO:  R10F OFF
```

{FUN} {025} REGISTER EQUAL CONSTANT COMPARISON

INPUT---[(A) = (B)]--DECISION OUTPUT
DATA DISPLAY-> *****

LEGAL ENTRIES

(A) COMPARED REGISTER RW, XW, YW, ZW, D, T, C
(B) CONSTANT Numerical value only

OPERATION

When the input turns ON, the value in register (A) is compared to the constant (B). If register (A) is equal to the constant (B), then the output turns ON.

EXECUTION TIME (microseconds) Not Executed 1.8/2.5 Executed 98.0

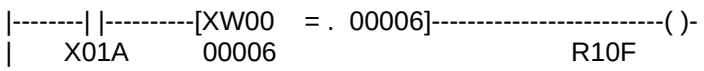
MEMORY REQUIREMENTS

For (B) < or = 255 : 3 steps
For (B) > or = 256 : 4 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, the content of register XW00 (00006) is compared with that of the constant (00006). If XW00 equals the constant, the output turns ON, and if XW00 is not equal to the constant, the output turns OFF. When contact X01A is OFF, the comparison is not executed and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

XW00 [00006] = 00006 --> YES: R10F ON
NO: R10F OFF

{FUN} {026} REGISTER LESS THAN CONSTANT COMPARISON

```
      INPUT---[ (A)  <.  (B) ]--EXECUTION OUTPUT
DATA DISPLAY->  *****
```

LEGAL ENTRIES
(A) COMPARED REGISTER RW, XW, YW, ZW, D, T, C
(B) CONSTANT Numerical value only

OPERATION

When the input turns ON, the value in register (A) is compared to the constant (B). If register (A) is less than constant (B), the output turns ON.

EXECUTION TIME (microseconds) Not Executed 1.8/2.5 Executed 98.0

MEMORY REQUIREMENTS

For (B) < or = 255 : 3 steps
For (B) > or = 256 : 4 steps

EXAMPLE DESCRIPTION

When contact X01A is ON, the content of register XW00 (00002) is compared with the constant (00005). If XW00 is less than 00005, the output turns ON. If XW00 is greater than, or equal to 00005, the output turns OFF. When contact X01A is OFF, the comparison is not executed and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[XW00    < .  00005]--------------------------( )-
|       X01A          00002                                  Y040
```

FUNCTION BLOCK

```
XW00 [00002]  <  00005  -->  YES: Y040 ON
                             NO:  Y040 OFF
```

7.5 Logical Operations

{FUN} {030} REGISTER - REGISTER LOGICAL AND

```
      INPUT---[ (A)  AND  (B)  -->  (C) ]--EXECUTION OUTPUT
DATA DISPLAY->  *****     *****     *****
```

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(C) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, bits in operand (A) are logically AND with the corresponding bits in operand (B). The results are then stored in the destination register (C).
* Operation is executed every scan the input is ON.
* This instruction uses registers only, constants are not acceptable.
* Operand and destination registers must be compatible for correct execution.
* AND truth table is as follows:

(A) AND (B) --> (C)

0	0	0
1	0	0
0	1	0
1	1	1

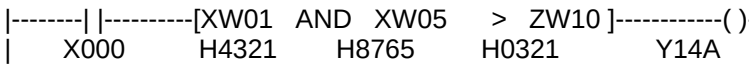
EXECUTION TIME (microseconds) Not Executed 2.5 Executed 107.0

MEMORY REQUIREMENTS 4 steps

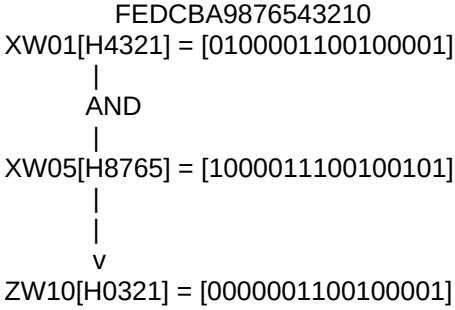
EXAMPLE DESCRIPTION

When contact X000 is ON, the data in register XW01 and XW02 are logically ANDed bit-by-bit. The results are stored in register ZW10 and the output is ON. When contact X000 is OFF, operation is not executed and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK



{FUN} {031} REGISTER - REGISTER LOGICAL OR

INPUT---[(A) OR (B) --> (C)]--EXECUTION OUTPUT
DATA DISPLAY-> ***** ***** *****

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(C) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, bits in operand (A) are logically ORed with the corresponding bits in operand (B). The result is stored in the destination register (C).
* Operation is executed every scan the input is ON.
* Operand and destination registers must be compatible for correct execution.
* This instruction uses registers only. Numerical values are not acceptable.
* OR truth table is as follows:

(A)	OR	(B)	-->	(C)
0		0		0
1		0		1
0		1		1
1		1		1

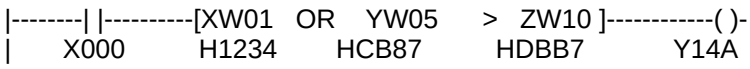
EXECUTION TIME (microseconds) Not Executed 2.5 Executed 107.0

MEMORY REQUIREMENTS 4 steps

EXAMPLE DESCRIPTION

When contact X000 is ON, the data in registers XW01 and YW05 are logically ORed bit-by-bit; the result is stored in register ZW10, and the output is ON. When contact X000 is OFF, operation is not executed and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

```
            FEDCBA9876543210
XW01[H1234] = [0001001000110100]
          |
         OR
          |
YW05[H8765] = [1100101110000111]
          |
          |
          v
ZW10[H0321] = [1101101110110111]
```

{FUN} {032} REGISTER - REGISTER LOGICAL EXCLUSIVE OR

```
      INPUT---[ (A)  EOR  (B)  -->  (C) ]--EXECUTION OUTPUT
DATA DISPLAY->  *****     *****     *****
```

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(C) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input X000 is ON, bits in operand (A) are logically EORed with the corresponding bits in operand (B). The results are then stored in the destination register (C).
* Operation is executed every scan the input is ON.
* Operand and destination registers must be compatible for correct execution.
* This instruction uses registers only; constants are not acceptable.
* EOR truth table is as follows:

(A)	EOR	(B)	-->	(C)
0		0		0
1		0		1
0		1		1
1		1		0

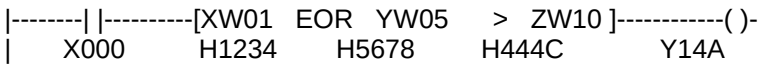
EXECUTION TIME (microseconds) Not Executed 2.5 Executed 107.0

MEMORY REQUIREMENTS 4 steps

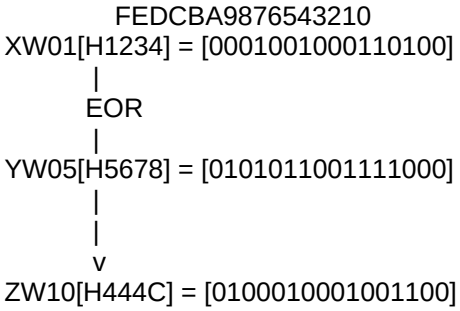
EXAMPLE DESCRIPTION

When contact X000 is ON, the data in registers XW01 and YW05 are logically Exclusive-ORed bit-by-bit. The result is stored in register ZW10, and the output is ON. When contact X000 is OFF, operation is not executed and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK



{FUN} {034} REGISTER LOGICAL NOT

```
      INPUT---[ (A)  NOT  (B) ]--EXECUTION OUTPUT
DATA DISPLAY->  *****     *****
```

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, bits in the operand (A) are inverted and stored in the destination register (B). Zeroes become ones, and ones become zeroes.
* The operation is executed every scan the input is ON.
* This instruction uses registers only. Constants are not acceptable.
* Operand data is not affected by execution of this instruction
* NOT truth table is as follows:

(A)	NOT	(B)
0		1
1		0

EXECUTION TIME (microseconds) Not Executed 1.8 Executed 100.0

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

When contact X000 is ON, all the bits in register XW01 are inverted. The result is stored in register YW05 and the output is ON. When contact X000 is OFF, operation is not executed, and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[XW01  NOT  YW05 ]----------------------------( )-
      X000          H1234        HEDCB                          Y14A
```

FUNCTION BLOCK

```
             FEDCBA9876543210
XW01[H1234] = [0001001000110100]
           |
          NOT
           |
           v
YW05[HEDCB] = [1110110111001011]
```

{FUN} {035} ROTATE BITS RIGHT

```
     INPUT---[ (A)  RTR  (B)  -->  (C) ]--BIT 0 DECISION OUTPUT
DATA DISPLAY->  *****              *****
```

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) NO. POSITIONS ROTATED Numerical value only
(C) DESTINATION REGISTER RW, XW, YW, ZW, D, T, C

OPERATION

When the input is ON, the bit pattern in the operand (A) is rotated to the right (toward the least significant bit) by the number of positions indicated in (B). The result is then stored in the destination register (C). The least significant bit wraps around to the most significant bit. The status of the output is determined by the bit in position zero (the least significant bit) in (C). If bit zero is a one, the output is ON. Otherwise, the output is OFF.
* The valid range of the number of bits for rotation (B) is 0 to 15.
* The operand register (A) is not changed by rotation.
* Operation is executed every scan the input is ON.
* Operand and destination registers must be compatible for correct execution.

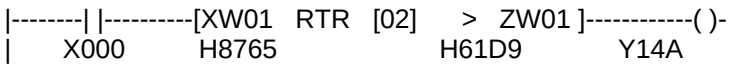
EXECUTION TIME (microseconds) Not Executed 2.5 Executed 106.0 + 5[nn]

MEMORY REQUIREMENTS 4 steps

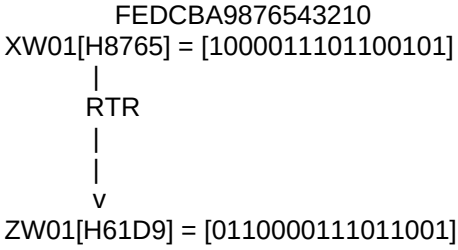
EXAMPLE DESCRIPTION

When contact X000 is ON, the bit pattern in register XW01, is rotated to the right 2 positions, and the resultant bit pattern is stored in register ZW01. In this case, the least significant bit, 0, is equal to 1, so the output is ON. When contact X000 is OFF, rotation is not performed and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK



{FUN} {036} ROTATE BITS LEFT

```
     INPUT---[ (A)  RTL  (B)  -->  (C) ]--BIT 0 DECISION OUTPUT
DATA DISPLAY->  *****              *****
```

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) NO. BITS ROTATED Numerical values only
(C) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, the bit pattern in operand (A) is rotated to the left, toward the most significant bit. It is rotated by (B) positions, and stored in the destination register (C). The most significant bit wraps around to the least significant bit. The status of the output is determined by bit zero (the least significant bit) in (C). If bit zero is a 1, the output is ON. If not, the output is OFF.
* The valid range for the number of rotated bits is 0 through 15.
* The operand register (A) is not changed by rotation.
* Operation is executed every scan the input is ON.
* Operand and destination registers must be compatible for correct execution.

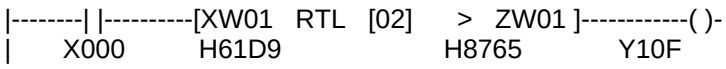
EXECUTION TIME (microseconds) Not Executed 2.5 Executed 106.0 + 5[nn]

MEMORY REQUIREMENTS 4 steps

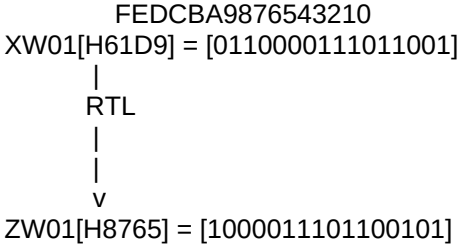
EXAMPLE DESCRIPTION

When contact X000 is ON, the bit pattern in register XW01 is rotated to the left by 2 positions, and the resultant bit pattern is stored in register ZW01. In this case, the least significant bit, 0, equals 1, so the output is ON. When the least significant bit, 0, equals 0, the output is OFF. When contact X000 is OFF, rotation is not executed and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK



{FUN} {040} REGISTER - CONSTANT LOGICAL AND

```
      INPUT---[ (A)  AND  (B)  -->  (C) ]--EXECUTION OUTPUT
DATA DISPLAY->  *****              *****
```

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) OPERATION DATA Numerical value only
(C) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, bits in operand (A) are logically ANDed with the corresponding bits in the constant (B). The result is then stored in the destination register (C).
* Operation is executed every scan the input is ON
* Operand and destination registers must be compatible for correct execution.
* AND truth table is as follows.

(A)	AND	(B)	-->	(C)
0		0		0
1		0		0
0		1		0
1		1		1

EXECUTION TIME (microseconds) Not Executed 2.3/3.1 Executed 109.0

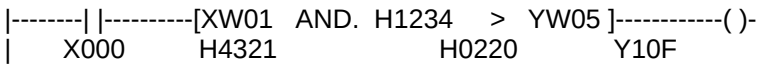
MEMORY REQUIREMENTS

For (B) < or = FF (HEX) : 4 steps
For (B) > or = 100 (HEX) : 5 steps

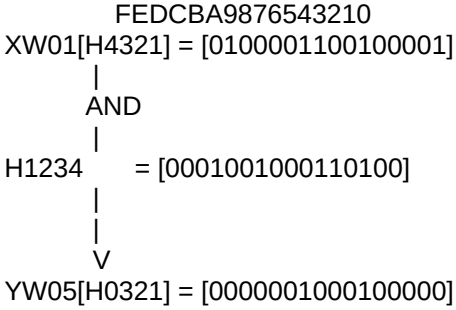
EXAMPLE DESCRIPTION

When contact X000 is turned ON, the data of register XW01 and constant H1234 are logically ANDed bit by bit. The result is stored in register YW05 and the output is turned ON. When N.O. contact X000 is turned OFF, operation is not executed and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK



{FUN} {041} REGISTER - CONSTANT LOGICAL OR

```
      INPUT---[ (A)  OR  (B)  -->  (C) ]--EXECUTION OUTPUT
DATA DISPLAY->  *****             *****
```

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) OPERATION DATA Numerical value only
(C) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, bits in operand (A) are logically OR with the corresponding bits in the constant (B). The results are stored in the destination register (C).
* Operation is executed every scan the input is ON.
* Operand and destination registers must be compatible for correct execution.
* OR truth table is as follows:

(A)	OR (B)	--> (C)
0	0	0
1	0	1
0	1	1
1	1	1

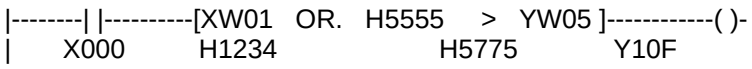
EXECUTION TIME (microseconds) Not Executed 2.5/3.1 Executed 109.0

MEMORY REQUIREMENTS
For (B) < or = H00FF : 4 steps
For (B) > or = H0100 : 5 steps

EXAMPLE DESCRIPTION

When contact X000 is turned ON, the data of register XW01 and constant H1234 are logically ORed bit by bit. The result is stored in register YW05 and the output is turned ON. When N.O. contact X000 is turned OFF, operation is not executed and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

```
             FEDCBA9876543210
XW01[H1234] = [0001001000110100]
          |
         OR
          |
H5555       = [0101010101010101]
          |
          |
          v
YW05[H5775] = [0101011101110101]
```

{FUN} {042} REGISTER - CONSTANT LOGICAL EXCLUSIVE OR

```
      INPUT---[ (A)  EOR  (B)  -->  (C) ]--EXECUTION OUTPUT
DATA DISPLAY->  *****              *****
```

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) OPERATION DATA Numerical value only
(C) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, bits in operand (A) are logically EORed with the corresponding bits in the constant (B). The results are then stored in the destination register (C).
* Operation is executed every scan the input is ON.
* Operand and destination registers must be compatible for correct execution.
* EOR truth table is as follows:

(A)	EOR	(B)	-->	(C)
0		0		0
1		0		1
0		1		1
1		1		0

EXECUTION TIME (microseconds) Not Executed 2.5/3.1 Executed 109.0

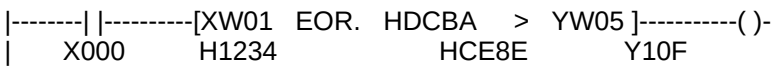
MEMORY REQUIREMENTS

For (B) < or = H00FF : 4 steps
For (B) > or = H0100 : 5 steps

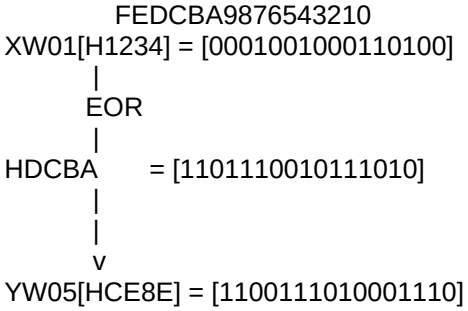
EXAMPLE DESCRIPTION

When contact X000 is turned ON, the data in register XW01 and the constant HDCBA are logically Exclusive ORed bit by bit. The result is stored in register YW05 and the output is turned ON. When contact X000 is OFF, operation is not executed, and the output is turned OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK



{FUN} {043} REGISTER - CONSTANT BIT TEST

INPUT---[(A) TEST (B)]--DECISION OUTPUT
DATA DISPLAY-> *****

LEGAL ENTRIES

(A) OPERATION REGISTER RW, XW, YW, ZW, D, T, C
(B) TEST DATA Numerical value only

OPERATION

When the input is ON, the bits in operand (A) are logically ANDed with the corresponding bits in the constant (B). If the logical AND of any two bits results in a one, the output turns ON. If the results of all comparisons are zeroes, then the output is OFF.
* Operation is executed every scan the input is ON.
* Operand data is not affected by execution of this instruction.
* Operand and destination registers must be compatible for correct execution.
* AND truth table is as follows:

(A)	AND	(B)	-->	(C)
0		0		0
1		0		0
0		1		0
1		1		1

EXECUTION TIME (microseconds) Not Executed 1.8/12.5 Executed 98.0

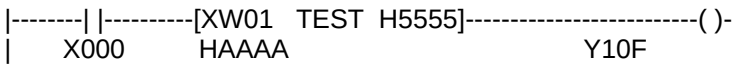
MEMORY REQUIREMENTS

For (B) < or = H00FF : 3 steps
For (B) > or = H0100 : 4 steps

EXAMPLE DESCRIPTION

When contact X000 is ON, the data in register XW01 and constant H5555 are logically ANDed bit by bit. In the example the result equals zero, and the output is OFF. When contact X000 is OFF, operation is not executed and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

```
             FEDCBA9876543210
XW01[HAAAA] = [1010101010101010]
          |
         AND
          |
H5555       = [0101010101010101]
          |
          |
          v
Result      = [0000000000000000] : = 0: Y10F  OFF
                                   > 0: Y10F  ON
```

{FUN} {046} TWO'S COMPLEMENT

```
      INPUT---[ (A)  NEG  (B) ]--EXECUTION OUTPUT
DATA DISPLAY->  *****     *****
```

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) COMPLEMENT REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, all bits in the operand (A) are inverted (similar to the NOT as in FUN 034), and one (1) is added before being stored in the destination register (B).
* Operation is executed every scan the input is ON.
* Operand and destination registers must be compatible for correct execution.
* This instruction uses registers only. Constants are not acceptable.
* Operand data is not effected by execution of this instruction.

EXECUTION TIME (microseconds) Not Executed 1.8 Executed 100.0

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

When contact X000 is ON the two's compliment of the contents in register XW01 (inverted data + 1) is obtained and stored in register YW05, and the output is turned ON. When contact X000 is OFF, operation is not executed and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[XW01  NEG  YW05 ]----------------------------( )-
|      X000         H1234       HEDCC                        Y10F
```

FUNCTION BLOCK

```
             FEDCBA9876543210
XW01[H1234] = [0001001000110100]
                 |
            Invert all bits
                 |
Result      = [1110110111001011]
                 |
                + 1
                 |
                 |
                 v
YW05        = [1110110111001100] : Y10F is ON
```

7.6 Data Conversion Instructions

{FUN} {050} BCD TO BINARY CONVERSION

```
      INPUT---[ (A)  BIN  (B) ]--ERROR OUTPUT
DATA DISPLAY->  *****     *****
```

LEGAL ENTRIES

(A)	OPERAND REGISTER	RW, XW, YW, ZW, D, T, C
(B)	BINARY REGISTER	RW, YW, ZW, D

OPERATION

When the input is ON, the BCD (Binary Coded Decimal) value in operand register (A) is converted to binary data and stored in the destination register (B).
* If any of the BCD digits are greater than nine, indicating a hexadecimal number, they are assumed to be nine and the output turns ON to indicate a possible error.
* Operation is executed every scan the input is ON.
* Operand and destination registers must be compatible for correct execution.
* This instruction uses registers only. Constants are not acceptable.
* Operand data is not affected by execution of this instruction.

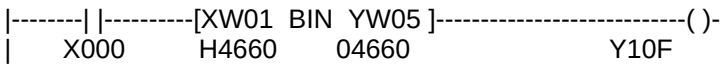
EXECUTION TIME (microseconds) Not Executed 1.8 Executed 194.0

MEMORY REQUIREMENTS 3 steps

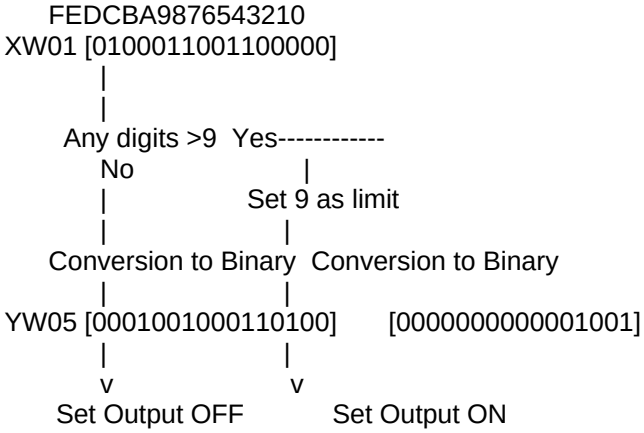
EXAMPLE DESCRIPTION

When contact X000 is ON, the content of XW01 is assumed to be 4-digit BCD data and is converted to binary data. The result is stored in YW05. When the BCD data contains digits larger than nine, they are assumed to be nine, and the output is turned ON. When contact X000 is turned OFF, operation is not executed and the output is turned OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK



{FUN} {051} BINARY TO BCD CONVERSION

```
      INPUT---[ (A)  BCD1  (B) ]--ERROR OUTPUT
DATA DISPLAY->  *****      *****
```

LEGAL ENTRIES

(A)	OPERAND REGISTER	RW, XW, YW, ZW, D, T, C
(B)	DESTINATION REGISTER	RW, YW, ZW, D

OPERATION

When the input is ON, binary data in operand register (A) is converted to 4-digit BCD (Binary Coded Decimal), and stored in the destination register (B). When the converted BCD data is greater than 9999, 9999 is stored in the destination register (B), and the output is turned ON as an error indication.
* Operation is executed every scan the input is ON.
* Operand and destination registers must be compatible for correct execution.
* This instruction uses registers only. Constants are not acceptable.
* Operand data is not effected by execution of this instruction.

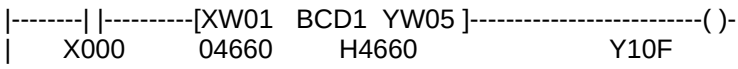
EXECUTION TIME (microseconds) Not Executed 1.8 Executed 125.0

MEMORY REQUIREMENTS 3 steps

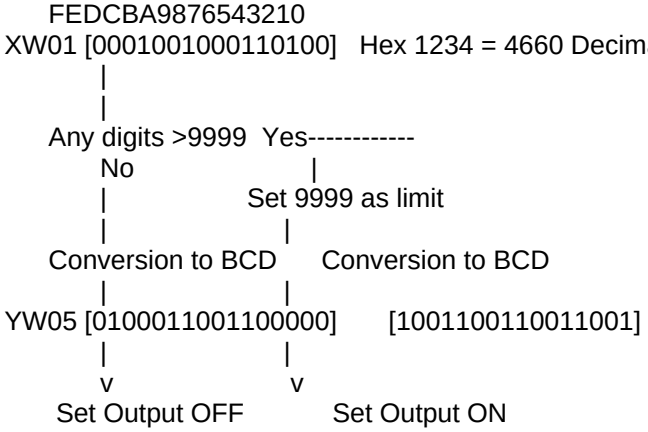
EXAMPLE DESCRIPTION

When contact X000 is ON, the content of register XW01 is converted into 4-digit BCD data, and stored in register YW05. When the converted BCD data is greater than 9999, 9999 is stored in YW05, and the output is turned ON.
When contact X000 is OFF, operation is not executed and the output is turned OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK



{FUN} {052} DOUBLE LENGTH BINARY TO BCD CONVERSION

```
     INPUT---[ (A)  BCD2  (B) ]--ERROR OUTPUT
DATA DISPLAY->  *****      *****
```

LEGAL ENTRIES

(A)	OPERAND REGISTER	RW, XW, YW, ZW, D
(B)	DESTINATION REGISTER	RW, YW, ZW, D

OPERATION

Operand register (A) is a double precision register. When double precision registers are used in any function, each operand is actually a pair of two consecutive registers, with the first register of each pair being the only one displayed on the screen. These registers are used as a single 32 bit value, and the value displayed is only a portion of the true total value for the operand. BCD register (B) is an extended register consisting of three consecutive registers, and is handled much like a double precision register. The three registers are needed in order to store the maximum value of 4,294. When the input is ON, the double precision register (A) is converted into BCD (Binary Coded Decimal), and stored in the extended register (B).

* Operation is executed every scan the input is ON.
* This instruction uses registers only. Constants are not acceptable.
* Operand data is not affected by execution of this instruction.

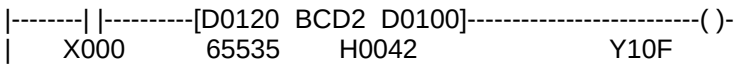
EXECUTION TIME (microseconds) Not Executed 1.8 Executed 290.0

MEMORY REQUIREMENTS 3 steps

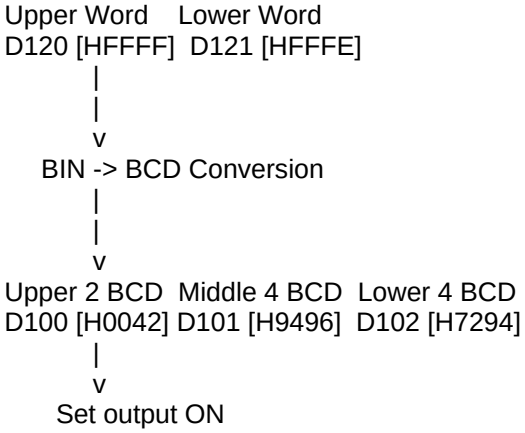
EXAMPLE DESCRIPTION

When contact X000 is turned ON, the double-precision data in register D0120 and D0121 is converted into BCD data, and the result is stored in registers D0100,D0101, and D0102. When the operation is executed, the output is ON. When contact X000 is OFF, the operation is not executed, and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK



{FUN} {053} ENCODE

INPUT---[(A) ENC (B)]--EXECUTION OUTPUT
DATA DISPLAY-> ***** *****

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, the position (0-F) of the highest order bit turned ON (1) in operand (A) is stored in Binary in the four least significant bits in the destination register (B). If the content of the operand (A) is all zeros, then ones are stored in all 16 bits of the destination register (B).
* Operation is executed every scan when the input is ON.
* This instruction uses registers only. Constants are not acceptable.
* Operand data is not effected by execution of this instruction.

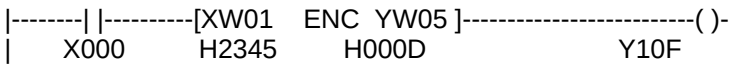
EXECUTION TIME (microseconds) Not Executed 1.8 Executed 104.0

MEMORY REQUIREMENTS 2 steps

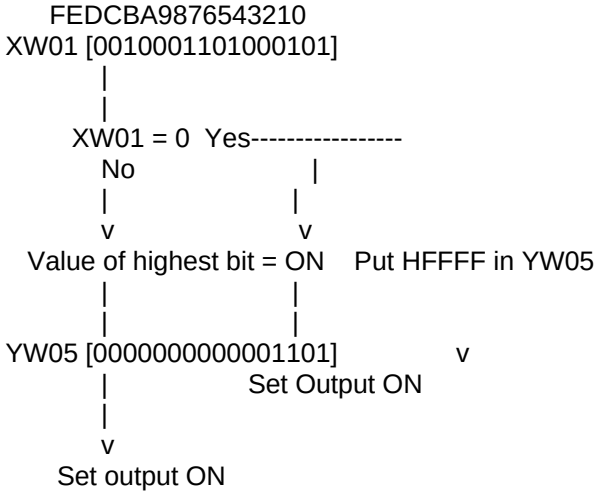
EXAMPLE DESCRIPTION

When contact X000 is ON, the highest bit position ON in XW01, which is D, is stored in register YW05, and the output is ON. (Note that while the Hex value of "D" is 13, it is the 14th position therefore the binary equivalent of 14 is stored in YW05.). When the content of register XW01 is zero (0), the value HFFFF is stored in YW05, and the output is turned ON. When the contact X000 is OFF, encode is not executed, and the output is turned OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK



{FUN} {054} DECODE

INPUT---[(A) DEC (B)]--EXECUTION OUTPUT
DATA DISPLAY-> ***** *****

LEGAL ENTRIES
(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, the 4 least significant digits of the operand register (A) are converted into one hexadecimal digit. This digit identifies the one bit which will be set ON (1) in the destination register (B).
* Other bits in the operand are not applicable to the operation of this instruction.
* Operation is executed every scan the input is ON.
* This instruction uses registers only. Constants are not acceptable.
* Operand data is not affected by execution of this instruction.

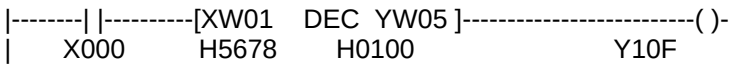
EXECUTION TIME (microseconds) Not Executed 1.8 Executed 104.0

MEMORY REQUIREMENTS 3 steps

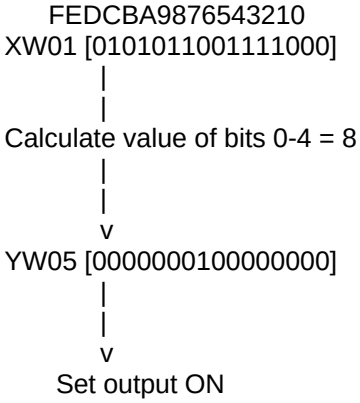
EXAMPLE DESCRIPTION

When contact X000 is ON, the bit position number 8 (note this is the ninth bit position because numbering begins with the first position being #0) in YW05 is set to 1 (ON) as specified by the 4 least significant bits in the register XW01, and the output is ON. Data in register XW01 is valid only for the 4 least significant bits, and invalid for all others. When contact X000 is OFF, decode is not executed, and output is turned OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK



{FUN} {055} BIT COUNT

```
     INPUT---[ (A)  BITC  (B) ]--EXECUTION OUTPUT
DATA DISPLAY->  *****      *****
```

LEGAL ENTRIES

(A) OPERAND REGISTER W, XW, YW, ZW, D, T, C
(B) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, the ON (1) bits in the operand register (A) are counted and the total is stored (in hexadecimal) in the destination register (B).
* The total in (B) should be a number representative of ON bits in 0 through F.
* Operation is executed every scan the input is ON.
* This instruction uses registers only. Constants are not acceptable.
* Operand data is not affected by execution of this instruction.

EXECUTION TIME (microseconds) Not Executed 1.8 Executed 178.0

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

When contact X000 is turned ON, the number of bits which are ON (1) in register XW01 are counted (in this case, 5). The total is stored in register YW05, and the output is ON. When contact X000 is OFF, the bit count is not executed, and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[XW01   BITC  YW05 ]--------------------------( )-
|      X000         H1234       H0005                        Y10F
```

FUNCTION BLOCK

```
    FEDCBA9876543210
XW01 [0001001000110100]
          |
          |
Count number of bits =1, 5
          |
          |
          v
YW05 [0000000000000101]
          |
          v
    Set output ON
```

7.7 Special Functions

{FUN} {060} UPPER LIMIT SET

```
      INPUT---[ (A)  UL  (B)  -->  (C) ]--DECISION OUTPUT
DATA DISPLAY->  *****    *****     *****
```

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) UPPER LIMIT REGISTER RW, XW, YW, ZW, D, T, C
(C) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, the value of operand register (A) is compared to the value of the upper limit register (B). If operand register (A) is greater than the upper limit register (B), the upper limit has been exceeded, the destination register is set to the value of the upper limit register (B), and the output turns ON. If the operand register (A) is less than the upper limit register (B), then the destination register is set to the value of operand register (A), and the output is OFF.

* Operation is executed every scan the input is ON.
* This instruction uses registers only. Constants are not acceptable
* Operand data is not effected by execution of this instruction.

EXECUTION TIME (microseconds) Not Executed 2.5 Executed 116.0

MEMORY REQUIREMENTS 4 steps

EXAMPLE DESCRIPTION

When contact X000 is turned ON, the upper limit is set in register YW05 at 3333. Since 5000 is larger than 3333, the value of YW05 (3333) is stored in register ZW01, and the output is turned ON. When the content of XW00 is less than the limit value , the content of XW00 is stored in ZW01, and the output is turned OFF. When contact X000 is OFF, the instruction is not executed, and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[XW01    UL   YW05     >   ZW01 ]------------( )-
|      X000         05000       03333       03333            Y10F
```

FUNCTION BLOCK

```
XW01 [05000] > YW05 [03333] --> YES: ZW01 [03333], Y10F ON
                                NO:  ZW01 [05000], Y10F OFF
```

{FUN} {061} LOWER LIMIT SET

```
      INPUT---[ (A)  LL  (B)  -->  (C) ]--DECISION OUTPUT
DATA DISPLAY->  *****    *****     *****
```

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) LOWER LIMIT REGISTER RW, XW, YW, ZW, D, T, C
(C) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, the value of operand register (A) is compared with the value of lower limit register (B). If operand (A) is less than lower limit register (B), the lower limit has been exceeded ,and the destination register is set to lower limit register (B). The output turns ON. If operand (A) is greater than lower limit register (B), then the destination register is set to operand (A), and the output turns OFF.
* Operation is executed every scan the input is ON.
* This instruction uses registers only. Constants are not acceptable.
* Operation data is not affected by execution of this instruction.

EXECUTION TIME (microseconds) Not Executed 2.5 Executed 116.0

MEMORY REQUIREMENTS 4 steps

EXAMPLE DESCRIPTION

When contact X000 is ON, the lower limit is set by register YW05 at 3500. Since 2000 is less than 3500, the value of YW05 (3500) is stored in register ZW01, and the output is turned ON. When the content of XW01 is larger than the limit, XW01 is stored in ZW01 and the output is turned OFF. When contact X000 is OFF, the instruction is not executed, and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[XW01    LL   YW05      >   ZW01 ]------------( )-
|      X000         02000       03500        03500            Y10F
```

FUNCTION BLOCK

```
XW01 [02000]  <  YW05 [03500] --> YES: ZW01 [03500], Y10F ON
                                NO:  ZW01 [02000], Y10F OFF
```

{FUN} {063} TABLE SEARCH - MAXIMUM VALUE

```
     INPUT---[ (A) MAX [nn] (B) ]--EXECUTION OUTPUT_
DATA DISPLAY-> *****         *****
```

LEGAL ENTRIES

(A)	TABLE START REGISTER	RW, XW, YW, ZW, D, T, C
(B)	MAXIMUM VALUE REGISTER	RW, YW, ZW, D
[nn]	TABLE LENGTH	Numerical value only

OPERATION

When the input is ON, the table (defined by the first register in the table, (A) and a length of [nn]) is searched for the maximum numerical value. When found, the maximum value is stored in the maximum value register (B), and the location of the pointer is stored in location register (B+1).
* If there are two equal maximum values in the table, the pointer will identify the one with the smaller pointer value.
* Operation is executed every scan the input is ON.
* The maximum table size is 64 registers.
* The first register location in the table is at pointer position zero.

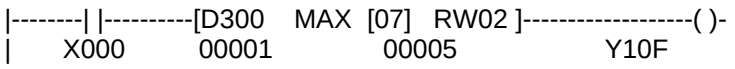
EXECUTION TIME (microseconds) Not Executed 2.5 Executed 110 + 9[nn]

MEMORY REQUIREMENTS 4 steps

EXAMPLE DESCRIPTION

When contact X000 is turned ON, the maximum value is retrieved from the table starting at D300 and stored in register RW02. The table pointer, located in RW03, indicates the location of the maximum value in relation to the first register in the table, which is position zero. The output is turned ON. When contact X000 is OFF, operation is not executed and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

TABLE	POINTER	DESTINATION
D300 [00001]	0	RW02 [00005] Maximum value
D301 [00002]	1	RW03 [00003] Table location
D302 [00003]	2	of maximum value
D303 [00005]	3	
D304 [00004]	4	
D305 [00000]	5	
D306 [00001]	6	

{FUN} {063} TABLE SEARCH - MINIMUM VALUE

```
      INPUT---[ (A)  MIN [nn]  (B) ]--EXECUTION OUTPUT
DATA DISPLAY->  *****          *****
```

LEGAL ENTRIES

(A) TABLE START REGISTER RW, XW, YW, ZW, D, T, C
(B) MINIMUM VALUE REGISTER RW, YW, ZW, D
[nn] TABLE LENGTH Numerical value only

OPERATION

When the input is ON, the table (defined by the table start register (A) and the length [nn]) is searched for the minimum numerical value. When found, the minimum value is stored in the minimum value register (B) and the location of the pointer is stored in the pointer location register (B+1).
* If there are two equal minimum values in the table, the pointer will identify the one with the smaller pointer value.
* Operation is executed every scan the input is ON.
* The maximum table size is 64 registers.

EXECUTION TIME (microseconds) Not Executed 2.5 Executed 110.0 + 9[nn]

MEMORY REQUIREMENTS 4 steps

EXAMPLE DESCRIPTION

When contact X000 is turned ON, the minimum value is retrieved from the table starting at D0310 and stored in register D0318. The table pointer, located in D0319, indicates the location of the minimum value in relation to the first register in the table which is position zero, and the output is ON. When contact X000 is OFF, operation is not executed and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[D0310    MIN  [06]   D0318 ]--------------------( )-
|      X000          00008              00001                   Y10F
```

FUNCTION BLOCK

```
TABLE          POINTER    DESTINATION
D0310 [00008]  0          D0318 [00001] Minimum value
D0311 [00007]  1          D0319 [00002] Table location
D0312 [00001]  2                        of minimum value
D0313 [00002]  3
D0314 [00003]  4
D0315 [00005]  5
```

{FUN} {064} TABLE AVERAGE - ALL TABLE VALUES

```
      INPUT---[ (A)  AVE [nn]  (B) ]--EXECUTION OUTPUT
DATA DISPLAY->  *****          *****
```

LEGAL ENTRIES

(A) TABLE START REGISTER RW, XW, YW, ZW, D, T, C
(B) AVERAGE VAL. REGISTER RW, YW, ZW, D
[nn] TABLE LENGTH Numerical value only

OPERATION

When the input is ON, all table values defined by the table start register (A) and the length [nn], are averaged and the average value is stored in the average value register (B). The output is turned ON.
* Operation is executed every scan the input is ON.
* The maximum table size is 64 registers.

EXECUTION TIME (microseconds) Not Executed 2.5 Executed 147.0 + 8[nn]

MEMORY REQUIREMENTS 4 steps

EXAMPLE DESCRIPTION

When contact X000 is ON, the average of the table, length = 06 registers, starting at register D320 is obtained, and the result is stored in register D328. The output is turned ON. When contact X000 is OFF,the operation is not executed, and the output is turned OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[D320    AVG  [06]   D328 ]-------------------( )-
|       X000          00002              00003                  Y10F
```

FUNCTION BLOCK

```
TABLE         POINTER        DESTINATION
D0320 [00002]  0
D0321 [00001]  1
D0322 [00003]  2
D0323 [00006]  3      -----> D0328 [00003] Average value
D0324 [00007]  4
D0325 [00003]  5
```

{FUN} {065} FUNCTION GENERATOR

INPUT---[(A) FG [nn] (B) --> (C)]--EXECUTION OUTPUT
DATA DISPLAY-> ***** ***** *****

LEGAL ENTRIES

(A) INPUT DATA REGISTER RW, XW, YW, ZW, D, T, C
(B) TABLE START REGISTER RW, XW, YW, ZW, D
(C) OUTPUT REGISTER RW, YW, ZW, D
[nn] TABLE LENGTH Numerical value only

OPERATION
Two dimensional graphic data is entered in the table starting at (B), the first half of the [nn] table is the X values and the second is the Y values. When an X value is input in (A), the corresponding Y value will be output at (C). For points not given in the table, the Y value will be calculated using the adjacent X-X : Y-Y points and the equation for a straight line, Y = mX + B. The function generator can also be called an interpolator; given x find f(x).
* When (A) is > or = to the largest value of X given in the table, the value of Y corresponding to the largest X is stored in (C).
* When (A) is < or = to the smallest value of X given in the table, the value of Y corresponding to the smallest X is stored in (C).
* The data table should be filled with values of X in ascending order, hen values of Y in ascending order.
* The table length [nn] is specified by the number of X values. The actual length is 2 x [nn] since the Y values must also be entered in the table.

EXECUTION TIME (microseconds) Not Executed 3.1 Executed 367.0 + 37[nn]

MEMORY REQUIREMENTS 5 steps

EXAMPLE DESCRIPTION

X values are entered in the data table starting at D400 and Y values are entered starting at D406. When X000 is ON, the X value in register XW02 [00010] is used to calculate the corresponding Y value and store it in register YW04 [00500]. The output is turned ON. When X000 is OFF, the operation is not executed and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[XW02    FG   [06]   D0400     >   YW04]-----( )-
|       X000          00010               00000         00500    Y10F
```

FUNCTION BLOCK

```
D0400 [00000]  X1        D0406 [00200]  Y1
D0401 [00005]  X2        D0407 [00275]  Y2
D0402 [00010]  X3        D0408 [00500]  Y3
D0403 [00015]  X4        D0409 [00875]  Y4
D0404 [00020]  X5        D0410 [01400]  Y6
D0405 [00025]  X6        D0411 [02075]  Y7
```

OR

Solves the formula Y = 200 + 3X**2. This does not have to be a continuous function.

{FUN} {070} SQUARE ROOT

```
      INPUT---[ (A)  RT  (B) ]--DECISION OUTPUT
DATA DISPLAY->  *****    *****
```

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, the square root of the content in operand register (A) and (A + 1) is stored in the destination register (B).
* When double precision registers are used in any function, each operand is actually a pair of two consecutive registers, with the first register of each pair being the only one displayed on the screen. These registers are used as a single 32 bit value and the value displayed is only a portion of the true total value for the operand.
* This instruction uses registers only. Constants are not acceptable.

EXECUTION TIME (microseconds) Not Executed 1.8 Executed 413.0

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

When contact X000 is turned ON, the square root of the 32 bit (double precision) register D0360 (and D0361) is obtained and stored in register D0366. The output is turned ON. When X000 is OFF, operation is not executed, and the output is turned OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[D0360  RT  D0366]--------------------------( )-
|      X000          00867      07580                        Y10F
```

FUNCTION BLOCK

RT(D0360 [00867], D0361 [47044]) = D0366 [07580]
RT(867 x 65535 + 47044) = 7580

{FUN} {071} SINE

INPUT---[(A) SIN (B)]--DECISION OUTPUT
DATA DISPLAY-> ***** *****

LEGAL ENTRIES
(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, the sine of operand register (A) is calculated and stored in the destination register (B). Operand (A) is expressed as an angle x 100, and the sine is derived by dividing the destination register (B) by 10000.

* The range of operand (A) is 0 to 65535.
* The relative error is less than +/- 0.8%.
* This instruction uses registers only. Constants are not acceptable.

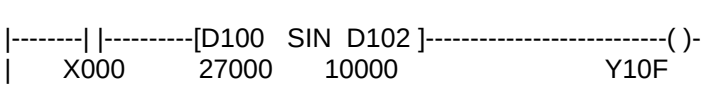
EXECUTION TIME (microseconds) Not Executed 1.8 Executed 666.0

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

When contact X000 is ON, the SINE of the content in register D100, is obtained and the result is stored in register D102. When SIN(A/100) is positive, the output is OFF. When it is negative, the output is ON, In this example, 10000 is stored in D102 and Y10F is turned ON.

EXAMPLE CIRCUIT



FUNCTION BLOCK

SIN (27000) = SIN | D100 [27000] x 1000/100 |
  = D102 [10,000]
OR

(B) = 10000 x |Sin(A)/100|

{FUN} {072} ARC-SINE

INPUT---[(A) ASIN (B)]--EXECUTION OUTPUT
DATA DISPLAY-> ***** *****

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, the ARC-SINE of operand register (A) is calculated and stored in the destination register (B). Operand register (A) is expressed as ARC-SIN x 10000, and the ARC-SINE is derived by dividing the destination register (B) by 100.
* The range of operand (A) is 0 to 10000.
* The relative error is less than +/- 1%
* This instruction uses registers only. Constants are not acceptable.
* When operand (A) is greater than 10000, the destination register (B) is set to 9000.

EXECUTION TIME (microseconds) Not Executed 1.8 Executed 819.0

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

When contact X000 is ON, the ASIN of the contents in register D100, which in this example is 5000, is obtained and stored in register D102, and the output is turned ON. When contact X000 is OFF, operation is not executed and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[D0100  ASIN  D0102]-------------------------( )-
|      X000          05000       03000                      Y10F
```

FUNCTION BLOCK

ASIN (5000) = ASIN | D0100 [5000 x 10,000] |
= ASIN | D0100 [50000000] |
= D0102 [3000]

OR

(B) = 100 x ASIN ((A)/1000)

{FUN} {073} COSINE

INPUT---[(A) COS (B)]--DECISION OUTPUT
DATA DISPLAY-> ***** *****

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, the cosine of operand register (A) is calculated and stored in the destination register (B). Operand (A) is expressed as an angle x 100, and the cosine is derived by dividing the destination register (B) by 10000.
* The range of operand (A) is 0 to 65535.
* The relative error is less than +/- 0.8%.
* This instruction uses registers only. Constants are not acceptable.

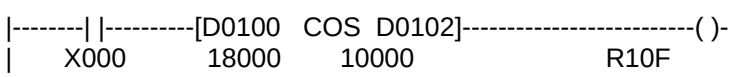
EXECUTION TIME (microseconds) Not Executed 1.8 Executed 674.0

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

When contact X000 is ON, the COSINE of the content in register D100 is obtained and stored in register D0102. When COS((A) / 100) is positive, the output is OFF. When it is negative, the output is ON. In this example, 10000 is stored in D0102 and R10F is turned ON. When contact X000 is OFF, operation is not executed and output is turned OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

(B) = 10,000 x |Cos ((A)/100)|
  = 10,000

{FUN} {074} ARC-COSINE

INPUT---[(A) ACOS (B)]--EXECUTION OUTPUT
DATA DISPLAY-> ***** *****

LEGAL ENTRIES

(A) OPERAND REGISTER RW, XW, YW, ZW, D, T, C
(B) DESTINATION REGISTER RW, YW, ZW, D

OPERATION

When the input is ON, the ARC-COSINE of operand register (A) is calculated and stored in the destination register (B). Operand (A) is expressed as (Arc-Cosine x 10000)and the Arc-cosine is derived by dividing the destination register (B) by 100.
* The range of operand (A) is 0 to 10000.
* The relative error is less than +/- 1%.
* When operand (A) is greater than 10000, the destination register (B) is set to zero.
* This instruction uses registers only. Constants are not acceptable.

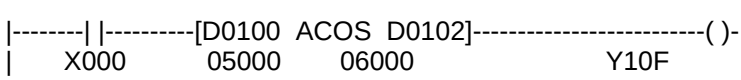
EXECUTION TIME (microseconds) Not Executed 1.8 Executed 824.0

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

When contact X000 is on the ACOS of the contents in register D0100 (5000) is obtained and stored in register D0102, and the output is turned 0N. When contact X000 is OFF, operation is not executed, and the output is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK

(B) = 100 x ACOS((A)/ 10000)
  = 6000

{FUN} {080} DEVICE SET

```
   INPUT---[ SET  (A) ]--EXECUTION OUTPUT
DATA DISPLAY->    xxxxx
```

LEGAL ENTRIES

(A) SET DEVICE R, Y, Z

OPERATION

When the input turns ON, device (A) is set ON until a device reset command turns it OFF.
* Device set is usually paired with device reset.
* Device set is similar to the "latch" command.

EXECUTION TIME (microseconds) Not Executed 1.2 Executed 90.0

MEMORY REQUIREMENTS 2 steps

EXAMPLE DESCRIPTION

When contact X000 is turned ON, the device Z100 is set ON, and the output is turned ON. Even if X000 is turned OFF, the set state continues. When contact X000 is OFF, "device set", is not executed, and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[ SET  Z100 ]----------------------------------( )-
|       X000                                          Y10F
```

TIMING DIAGRAM X = ON _ = OFF

```
X000    __XXXXXXXXX__________

Z100    __XXXXXXXXXXXXXXXXXXX

Y10F    __XXXXXXXXX__________
```

{FUN} {081} DEVICE RESET

```
   INPUT---[ RST  (A) ]--EXECUTION OUTPUT
DATA DISPLAY->    xxxxx
```

LEGAL ENTRIES

(A) RESET DEVICE R, Y, Z

OPERATION

When the input turns ON, device (A) is used to reset (unlatch) a device which has previously been set (latched).
* Device reset is usually paired with device set.
* Device reset is similar to the "unlatch" command.

EXECUTION TIME (microseconds) Not Executed 1.2 Executed 93.0

MEMORY REQUIREMENTS 2 steps

EXAMPLE DESCRIPTION

When contact X000 is turned ON, device Z100 is reset to OFF, and the output is turned ON. Even if X000 is turned OFF, the reset state is unaffected. When contact X000 is OFF, "device reset" is not executed, and the output is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[ RST  Z100 ]----------------------------------( )-
|      X000                                          Y10F
```

TIMING DIAGRAM X = ON _ = OFF

```
X000    _____XXXXXXXXXX_____

Z100    XXXXX_________________

Y10F    _____XXXXXXXXXX_____
```

{FUN} {090} DIAGNOSTIC DISPLAY - ERROR CODE

```
   INPUT---[ DDSP (A) ]--EXECUTION OUTPUT
DATA DISPLAY->    xxxxx
```

LEGAL ENTRIES

(A) ERROR CODE DATA Numerical value only

OPERATION

When the input turns ON, the number assigned as the error code (A) will be stored in the error code registration table in location D0001. Previous error codes are bumped down one register in the table with the error code previously in location D004 being discarded. Register D0000 is the error counter and D0001 through D0004 store the errors for display. When a new error is entered into the table, a flashing DIAG. is visible in the alarm area on the programmer. If the display is switched to the System Information Page, screen 1, all four slots of the Diagnostic Error Table can be viewed.

* Data registers D0000 through D0004 must be reserved for this function.
* The input should be a transitional contact, or transfer will occur every scan.
* For adding an alphanumeric description to the error, see the DDSM (FUN 091) instruction.

EXECUTION TIME (microseconds) Not Executed 1.2-1.8 Executed 144.0-176.0

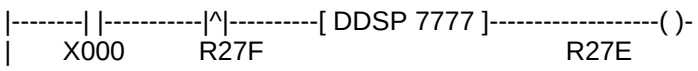
MEMORY REQUIREMENTS

For (A) < or = 99: 2 steps.
For (A) = or > 100: 3 steps.

EXAMPLE DESCRIPTION

When contact X000 turns ON, 7777 is sent to the error table location D0001, diagnostic alarms already in the table are bumped down one register, and the error in D004 is discarded. DIAG will appear in the alarm area of the programmer display. When the System Information Page, screen 1, is displayed, 7777 will be in slot 1 of the Diagnostic Alarm Block.

EXAMPLE CIRCUIT



FUNCTION BLOCK

D0000 [3] Counter
D0001 [H7777] | Last error code
D0002 [H0123] | Previous error code
D0003 [H1001] v 1st Error code
D0004 [ ]

{FUN} {091} DIAGNOSTIC DISPLAY - ERROR MESSAGE

```
      INPUT---[ DDSM (A)     (B) ]--EXECUTION OUTPUT
DATA DISPLAY->              *****
```

LEGAL ENTRIES

(A) ERROR CODE DATA Numerical value only
(B) MESSAGE DATA REGISTER D only

OPERATION

When the input turns ON, the number assigned as the error code (A) will be stored in the Error Code Registration Table in location D0001. Previous error codes are bumped down one register location in the table with the error code previously in location D0004 being discarded. Register D0000 is the error counter and D0001 through D0004 store the errors for display. When a new error is entered in the table, a flashing DIAG. is visible in the alarm area of the programmer. If the display is switched to the System Information Page, screen 1, all four slots of the Diagnostic Error Table can be viewed. The alphanumeric description of the error is contained in the data table starting with register (B). The description of the error is displayed in the event column on the System Information Page

* The message registration table is limited to six registers
* Invalid ASCII characters (control codes, etc.) will show on the display as dark boxes.
* The input should be a transitional contact, or transfer will be made every scan.
* When just a numerical diagnostic code is desired, use the instruction DDSP (FUN090).
* Data registers D0000 through D0004 must be reserved for this function.

EXECUTION TIME (microseconds) Not Executed 1.2-1.8 Executed 161.0-189.0

MEMORY REQUIREMENTS

For (A) < or = 99: 2 steps
For (A) > or = 100: 3 steps

EXAMPLE DESCRIPTION

When contact X004 turns ON, 8888 is sent to the error table location D0001, diagnostic alarms already in the table are bumped down one register and the error in D0004 is discarded. DIAG. will appear in the alarm area of the programmer display. When the System Information Page, screen 1, is displayed, 8888 will be in position 1 of the Diagnostic Alarm Block. In addition, the message "LIMIT OVER" will be displayed under the event column if the Data Registers D020 through D025 have been programmed as shown below:

EXAMPLE CIRCUIT

```
|--------| |-----------|^|----------[DDSM  8888   D0100]------------( )-
|      X004          R27D                                  R27C
```

FUNCTION BLOCK

D0000 [4] Counter
D0001 [H8888] Last error code
D0002 [H7777] Previous error code
D0003 [H1001] Previous error code

D0004 [H1001] 1st Error code

	MSB	LSB	
D0020	L	I	ASCII code, 12 characters
D0021	M	I	
D0022	T	-	
D0023	O	V	
D0024	E	R	
D0025	-	-	

{FUN} {096} IMMEDIATE INPUT

INPUT---[IN [nn] (A)]--EXECUTION OUTPUT

LEGAL ENTRIES

(A) 1ST INPUT REGISTER XW only
[nn] TABLE SIZE Numerical value 1 to 16

OPERATION

When the input turns ON, the CPU stops and updates its I/O table by reading [nn] input registers starting at (A)
* Output registers in the specified range are ignored.

EXECUTION TIME (microseconds) Not Executed 1.8 Executed 117 + 63[nn]

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

When input Z010 is ON, the CPU stops and reads a block of 10 I/O registers starting at XW00. The output R27C is set ON. When Z010 is OFF, no action is performed and R27C is OFF.

EXAMPLE CIRCUIT

```
|--------| |----------[ IN  [10]  XW00 ]----------------------------( )-
|      Z010                                          R27C
```

FUNCTION BLOCK

```
I/O Modules        CPU I/O Table
[  X-2W  ] ----->   XW00  [xxxxx]
[  Y-2W  ]   .      XW01  [xxxxx]
[  X-1W  ]          YW02  [xxxxx]  Ignored
[  Y-2W  ]   .      YW03  [xxxxx]  Ignored
[  X-2W  ]          XW04  [xxxxx]
[  Y-1W  ] - .      YW05  [xxxxx]  Ignored
          |         YW06  [xxxxx]  Ignored
          |.        XW07  [xxxxx]
          |         XW08  [xxxxx]
           ---->  YW09  [xxxxx]  Ignored
```

{FUN} {096} IMMEDIATE OUTPUT

INPUT---[OUT [nn] (A)]--EXECUTION OUTPUT

LEGAL ENTRIES

(A) 1ST OUTPUT REGISTER YW only
[nn] TABLE SIZE Numerical value 1 to 16

OPERATION

When the input turns ON, the CPU stops and the values of [nn] registers in the CPU I/O table starting at (A) are sent to the I/O modules.
* Input registers in the specified range are ignored.

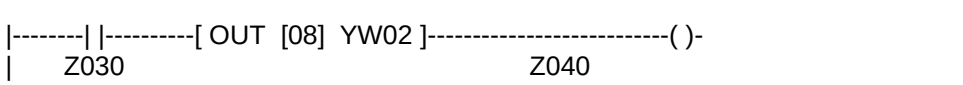
EXECUTION TIME (microseconds) Not Executed 1.8 Executed 117 + 43[nn]

MEMORY REQUIREMENTS 3 steps

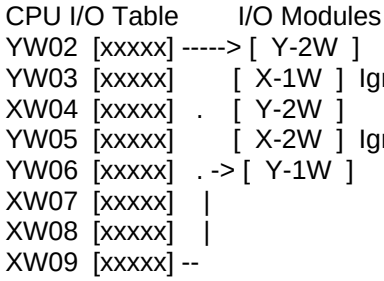
EXAMPLE DESCRIPTION

When input Z030 is ON, the CPU stops and writes a block of 8 I/O registers starting at YW02 to the I/O modules. The output Z040 is set ON. When Z010 is OFF, no action is performed and Z040 is OFF.

EXAMPLE CIRCUIT



FUNCTION BLOCK



{FUN} {100} STEP SEQUENCE INITIALIZATION

INPUT---[STIZ [nn] (A)]--EXECUTION OUTPUT

LEGAL ENTRIES

(A) START SEQUENCE DEVICE R only
[nn] STEP SEQUENCE LENGTH Numerical value 1-64

OPERATION

When the input turns ON, the first device in the step sequence table (A) is set ON (1), and the remainder of devices in the table are set OFF. This initializes the sequence. A step sequencer or drum sequencer uses the following three functions:
1) STEP SEQUENCE INITIALIZE (FUN100): Resets the table, sets the first device ON, and sets all subsequent devices OFF.
2) STEP SEQUENCE INPUT (FUN101): Transfers the sequence step from it's addressed device to the first input in the next step.
3) STEP SEQUENCE OUTPUT (FUN102): Turns on when all input conditions (including the step sequence input) have been satisfied.
* Maximum sequence length is 64 steps.
* This instruction must have a transitional contact proceeding the step sequence initialization block.
* The output coil is not necessary

EXECUTION TIME (microseconds) Not Executed 1.8 Executed 105.0-154.0]

MEMORY REQUIREMENTS 3 steps

EXAMPLE DESCRIPTION

Prior to starting a step sequence, this initialization operation should be executed. When contact X000 is turned ON, the step sequence first device R400 is set ON, and all the succeeding devices down to R409 are set OFF (since [nn] = 10, R400 + 10 - 1 = R409). The output is turned ON. When contact X000 is OFF, initialization is not executed, and the output is turned OFF.

EXAMPLE CIRCUIT

```
|--------| |-----------|^|---[STIZ  [10]   R400 ]--------------------( )-
|      X000         R23F                                 R241
```

FUNCTION BLOCK

```
R400  [  ON  ]
R401  [  OFF ]
        .
        .
R409  [  OFF ]
```

{FUN} {101} STEP SEQUENCE INPUT

INPUT---[(A)]--EXECUTION OUTPUT

LEGAL ENTRIES

(A) DEVICE R only

OPERATION

The step sequence input should only be programmed as the first device in a circuit. In the circuit, it acts like a normally open contact. When the step output in the previous circuit turns ON, it also turns ON and provides circuit continuity. It will reset (turn OFF) when the step output in its circuit turns ON.
* This command is combined with FUN100, STEP SEQUENCE INITIALIZATION and FUN102, STEP SEQUENCE OUTPUT.
* No more than 14 step inputs can be series or parallel connected in a circuit (see step sequence output application).

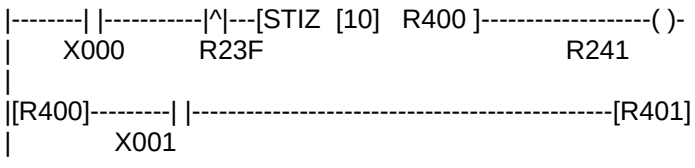
EXECUTION TIME (microseconds) Not Executed 1.2 Executed 98.0-118.0]

MEMORY REQUIREMENTS 2 steps

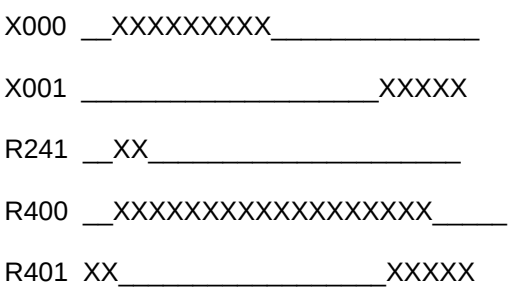
EXAMPLE DESCRIPTION

When contact X000 is turned ON, FUN100 "STEP SEQUENCE INITIALIZATION" is executed. R400 is set ON and all other step inputs are set OFF. While X001 is OFF, the step input R400 stays ON and the step output R401 is OFF. When X001 is turned ON, R400 is set OFF and step output R401 is set ON.

EXAMPLE CIRCUIT



TIMING DIAGRAM X = ON _ = OFF



{FUN} {102} STEP SEQUENCE OUTPUT

INPUT---[(A)]--|

LEGAL ENTRIES

(A) STEP SEQUENCE DEVICE R only

OPERATION

When the input(s) turns ON, the step output (A) in the step sequence turns ON. The previous step output which was ON will turn OFF and the step input in the following circuit will turn ON.
* This command is combined with FUN100, STEP SEQUENCE INITIALIZATION and FUN101, STEP SEQUENCE INPUT.
* All outputs are turned OFF by FUN100, STEP SEQUENCE INITIALIZE.
* Each step sequencer can have up to 64 step sequence outputs, [nn] = 64 max.

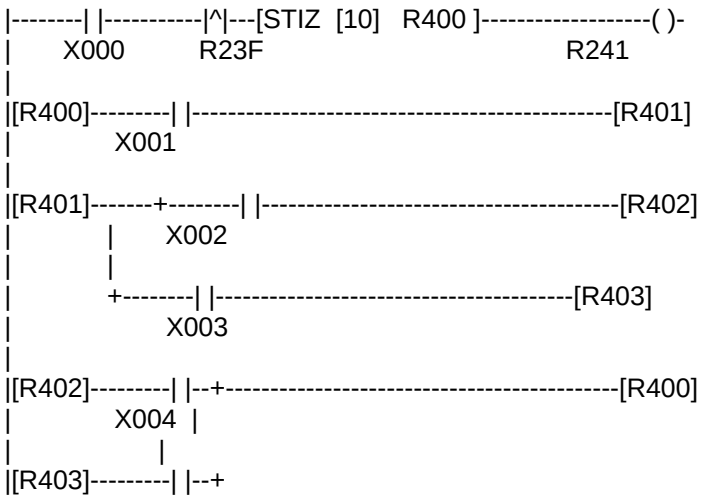
EXECUTION TIME (microseconds) Not Executed 0 Executed 83.0-141.0

MEMORY REQUIREMENTS 2 steps

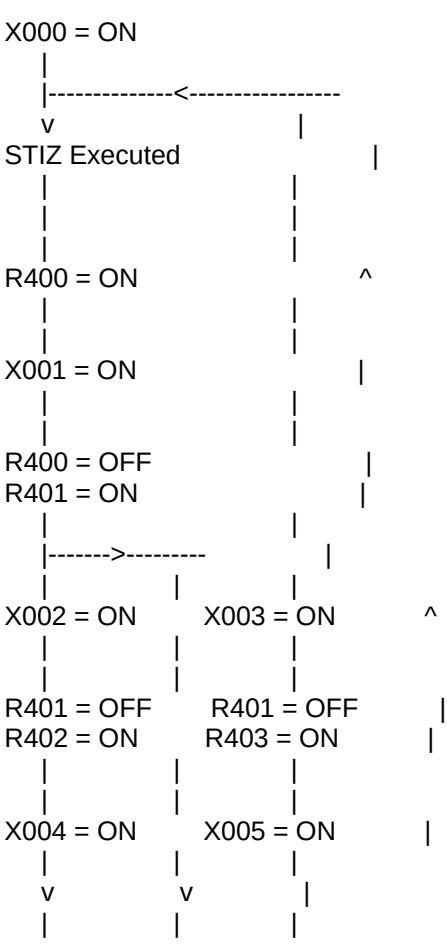
EXAMPLE DESCRIPTION

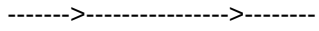
When the input (X000) is turned ON, all the devices registered via Step Sequence Initialization are cleared to OFF, and step input R400 is turned ON. As the program is executed, it sequences through step outputs from R401 to R405 as the contacts X001 through X007 are turned ON and OFF. At the completion of the sequence, the sequencer will re-initialize.

EXAMPLE CIRCUIT

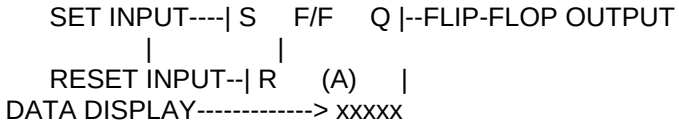


FUNCTION BLOCK





{FUN} {110} FLIP-FLOP



LEGAL ENTRIES

(A) FLIP-FLOP DEVICE R, Y, Z

OPERATION

When the SET INPUT is ON, the output turns ON. When the RESET INPUT turns ON, the output turns OFF.
When both SET and RESET inputs are ON, the output is OFF (RESET takes priority).
* Devices controlled by the flip-flop have the same state as the output.
* Logic table:

S	R	Q
OFF	OFF	Previous state
OFF	ON	OFF
ON	OFF	ON
ON	ON	OFF

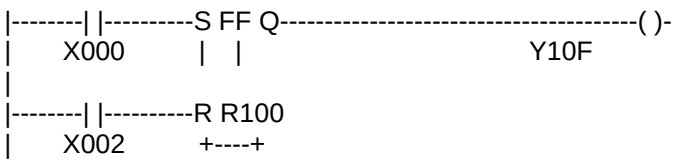
EXECUTION TIME (microseconds) Not Executed 0 Executed 102.0

MEMORY REQUIREMENTS 2 steps

EXAMPLE DESCRIPTION

When SET INPUT X000 is ON and RESET INPUT X002 is OFF, both R100 and the output Y040 are turned ON. When RESET INPUT X002 is ON, R100 and the output are turned OFF, irrespective of the state of SET INPUT X000. When both the SET INPUT and RESET INPUT are OFF, the state of R100 and the output Y040 are maintained. The Y10F output is not necessary for this instruction to operate properly.

EXAMPLE CIRCUIT



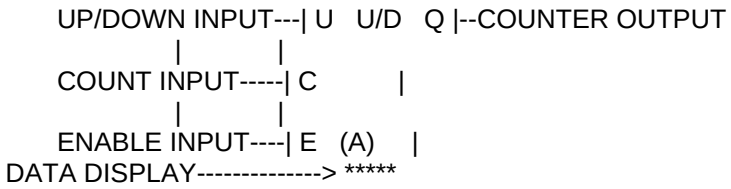
TIMING DIAGRAM X = ON _ = OFF

X000 __XXX_____XXX_____

X002 _____XXX_____XXX_____

R100 __XXXXXXXXX_____XXXXX_____

{FUN} {111} UP/DOWN COUNTER



LEGAL ENTRIES

(A) COUNTER REGISTER C only

OPERATION

When the ENABLE INPUT is ON, transitions from OFF to ON at the COUNT INPUT enter a count into the counter. If the UP/DOWN INPUT is ON, the total count will increment; if it is OFF, the total count will decrement. When the ENABLE INPUT is OFF, the count value is cleared to zero and the output is OFF.

* This instruction requires a transitional contact on the count input or count will occur every scan.
* Output will be ON at maximum count (65535) and at minimum count (0) for one scan only (the enable input must also be ON).

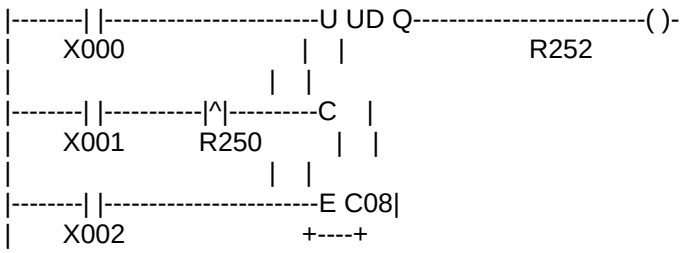
EXECUTION TIME (microseconds) Not Executed 0 Executed 98.0-116.0

MEMORY REQUIREMENTS 4 Steps

EXAMPLE DESCRIPTION

When enable input X002 is on, the counter will count one every time input X001 turns ON. If X000 is ON, the count will be up. If X000 is OFF, the count will be down. If X002 is OFF, the counter will clear to zero and the counter will not count until X002 turns ON again.

EXAMPLE CIRCUIT



TIMING DIAGRAM X = ON _ = OFF

X000 XXXXXXXXXXXXXX_____

X001 __XXX_____XXX_____XXX_____XXX_____

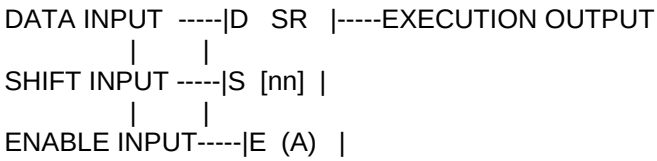
R250 __X_____X_____X_____X_____

X002 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

C08 __+1_____+1_____-1_____-1_____

Counter_____X_____
Output

{FUN} {112} SHIFT REGISTER



LEGAL ENTRIES

(A) DEVICE TYPE R Y Z
[nn] SHIFT REGISTER LENGTH Numerical value 1 to 64

OPERATION

When the enable input is ON and the shift input turns ON, all bits in the shift register are shifted to the next higher position. If the data input is ON, a 1 is put into the first bit position identified by (A). If the data input is OFF, a 0 is put into the first bit position (A). When a 1 is in the last bit position the output is ON, otherwise it is OFF. When the enable input is OFF, the shift register is cleared--all bits are set to 0.
* A transitional contact is required before the shift input, otherwise shift will occur every scan.
* Maximum shift register length is 64 bits.

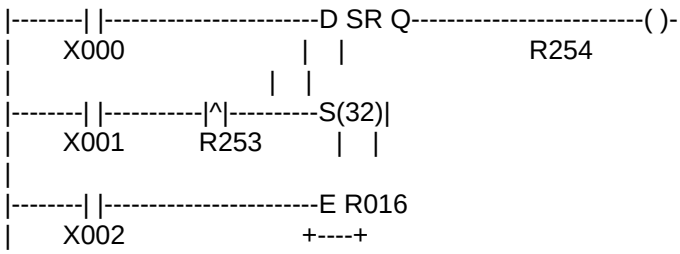
EXECUTION TIME (microseconds) Not Executed 0 Executed 118.0-404.0

MEMORY REQUIREMENTS 3 Steps

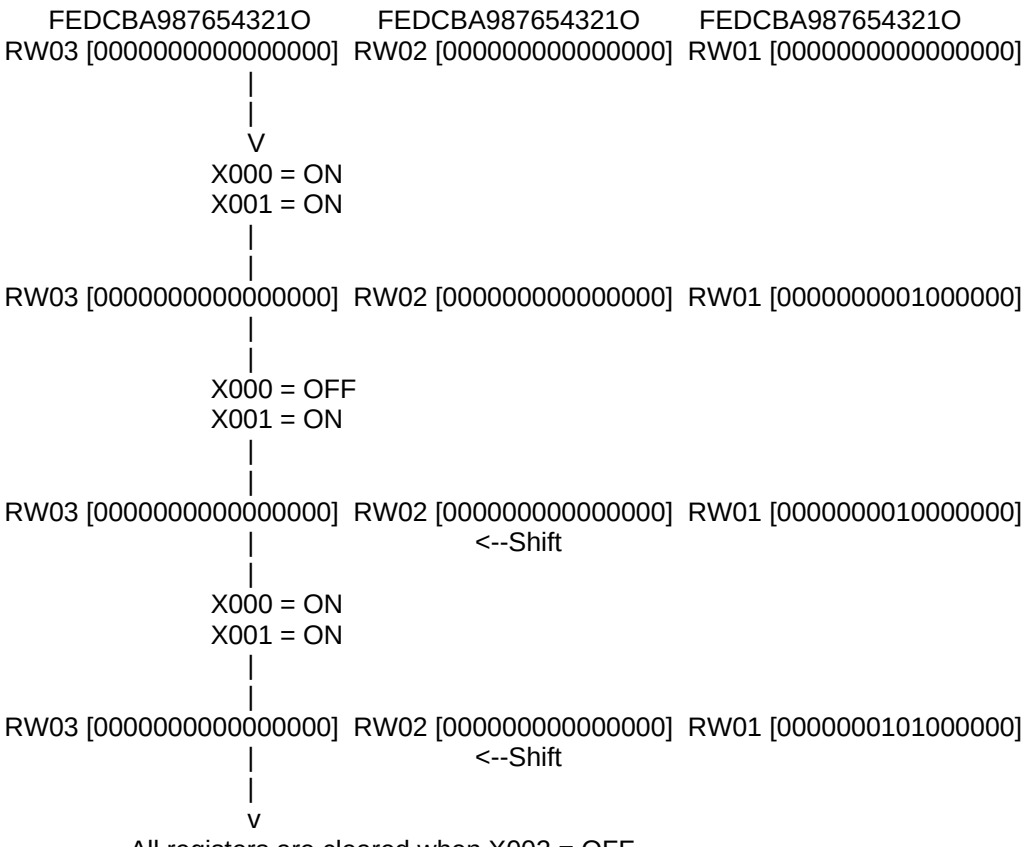
EXAMPLE DESCRIPTION

When the enable input X002 is ON and the shift input turns ON, all bits are shifted to the next higher bit position starting from the first position R016. If the data input X000 is ON, a 1 is put into R016. If X000 is OFF, a 0 is put into R016. If the enable input X002 is OFF, the shift register is cleared and no shifting is possible. The output is ON when the last bit, R035 is ON.

EXAMPLE CIRCUIT



FUNCTION BLOCK



****

8. Basic Programming Procedures

8.1 System design overview

This section describes the basic operations from designing a program for the M/EX100 to programming the M/EX100 and debugging the program. This basic procedure is recommended when designing a program.

1. Designing the system
Study the configuration of the systems to be controlled. Study the system's operation and fail-safe sequences thoroughly. Make sure the controllers selected can satisfy the number of I/O points, the memory capacity, required processing speed and other requirements.

2. Designing the program
Having designed the system, it is then necessary to write the program that will run the system.
The I/O assignment list should be decided first. Then write the M/EX100 program in accordance with the system operation sequence.

3. Assembling the unit
Select the M/EX100 memory setting (3K or 4K) and install the units/modules.

4. Initializing the system
Set the M/EX100 operation control switch to the HALT position and turn on the power. Execute the memory clear command from the programmer to initialize the memory in the M/EX100. It is then necessary to allocate the I/O registers. When these operations have been performed, the M/EX100 is ready to receive the program that you have written to run your system.

5. Entering the program
Write/load the program into the M/EX100. Refer to the appropriate manual for details on this procedure.

6. Debugging the program
After programming the M/EX100, set the operation control switch to the RUN position to debug the program. Be careful not to damage the field devices or other equipment when simulating or debugging the program.

7. Writing the program into the EEPROM
After debugging the program, and BEFORE turning the power to the M/EX100 off, be sure to write the program into the EEPROM. In the M/EX100, the program is transferred from the EEPROM to the RAM when the power is turned on. Therefore, make sure that the new program that has been modified is written into the EEPROM before turning the power off.

8.2 Basic programming procedures (shown here for HP100, it is very similar for the computer programming software).

The following flowchart shows the programming procedure using the HP100, in the chart, HP and PC represent the HP100 and the M/EX100 respectively. Use the HP100 manual (UM-EX25UB-E022) to look up the actual keystrokes to perform the following functions.

1 Connect HP and PC by the dedicated cable

2 Set the M/EX100's operation control switch to HALT — Programmer/link selector switch should be set to the programmer side

3 Apply power to the PC

4 Initial display (system information) appears on HP

5 Clear the PC memory

6 Perform I/O allocation

7 Confirm I/O allocation status

8 Call page to be programmed

9 Move to the edit mode

10 Write the program on the page

11 Enter edited program page into PC memory

12 Call the next page

13 Return to step 9 and repeat steps 9-12 until program is complete

14 Set program ID and retentive memory areas

15 Move PC's operation control to RUN, and debug program

16 Set PC operation mode to HALT

17 Return to EDIT mode to correct program if required

18 Change to HALT mode, then write the program into EEPROM

19 Remove power from PC — If the RUN/RUNP/HALT switch is set to the RUN position, the PC will enter the RUN mode automatically the next time power is applied.

NOTE When using a programmer that does not have the EEPROM write

command, the special relay R62E of the M/EX100 should be used to perform step 18. (See 2.5 and 5.1)

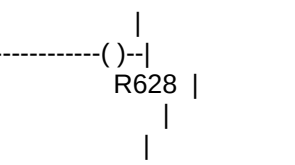
****

9. Special Functions

9.1 The clock-calendar function

The M/EX100 contains a clock-calendar that automatically keeps track of the year, month, day, day of the week, hour, minute, second. This function greatly simplifies scheduled operations and batch processing.

Operating method
To use the clock-calendar function, create the circuit shown on the right on the first page of the program. Creating this circuit assigns the data registers D0005 to D0010 for clock-calendar data.



Calendar registers NOTE THAT THESE ARE IN HEX, NOT IN DECIMAL

	F..8	7..0	Example
D005	----	Year	H0090 \
D006	----	Month	H0002 \
D007	Week	Day	H0227 ----- Feb. 27, 1990 (Tuesday)
D009	----	Minute	H0010 / 13:10:44
D010	----	Second	H0044 /

NOTE (1) Clock-calendar data is expressed in two-digit BCD codes.

(2) The day of the week is expressed as follows.
0: Sunday, 1: Monday, 2: Tuesday, 3: Wednesday,
4: Thursday, 5: Friday, 6: Saturday

Initializing
To initialize the clock-calendar data, either of the following two methods are available.

(1) Use the programmer or the computer link to write the initial values into D0005 to D0010.

(2) Execute the clock-calendar data set instruction.

NOTE (1) When power is turned off, the clock-calendar is updated by the M/EX100's built-in capacitor (backup: 7 days/25 degrees C). If power is not turned on before this backup period has elapsed, the data in the clock-calendar registers may be lost, and will have to be re-set.
If the optional battery is used, the clock-calendar will be backed up for two years.

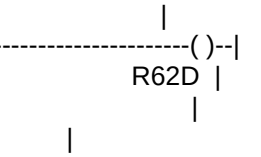
(2) The accuracy of the clock-calendar is +/- 30 seconds per month.

9.2 Forced operation
(Automatic RUN-F)

The M/EX100 can be operated when some of its I/O slots are vacant, provided the I/O allocation for those slots is blank, SP, or OPT. However, if slots whose I/O allocation is not blank, SP, or OPT are left vacant, under normal operation, an error will result when the M/EX100 makes the I/O response check prior to operation.

When necessary, however, the M/EX100 can be operated without all modules being mounted by using its forced operation, Auto RUN-F. This function enables a program to be debugged without the modules actually being installed.

Specifying forced operation
Create the circuit shown on the right on the first page of the program. RUN activation is now ready.



NOTE (1) Although forced operation allows the M/EX100 to be operated if some modules are not mounted according to I/O allocation, an error results if modules of a different type are mounted.

(2) This function is identical to the RUN-F command issued from a programmer.

9.3 The write-protect function

The M/EX100 provides a write-protect function using the program ID. By using the write-protect function, the following operation can be inhibited:
. Program modification, both in the RUN and HALT modes
. Changing the allocation of I/O registers
. Forcing or releasing devices and/or coils, both in the RUN and HALT modes.

To specify the write-protect, set the three lower characters of the program ID to FFF

Program ID [. . . . . F F F]

. Unrestricted

NOTE (1) This function does work with the computer link.

(2) The program ID can be changed either in the RUN or HALT mode.

(3) After specifying the write-protect, it is not possible to modify the program or perform any other writing operation. If writing is attempted, a MODE ERROR will be displayed.

9.4 The hold function

The hold function enables program execution to be stopped, with only input and output updating being executed. It is therefore possible to suspend program execution while holding the output state. Moreover, a desired output state can be established by setting any data in the external output register while in the hold state. This function is useful for checking external wiring or output devices. To enter the hold state, turn on special relay R629 while in the RUN mode. Relay R629 can be turned on by means of a program instruction, greatly simplifying program debugging.

NOTE (1) To reset the M/EX100 to the RUN mode, simply turn off special relay R629 by using a programmer.

(2) When in the hold state, the RUN LED will blink.

9.5 The EEPROM read/write function

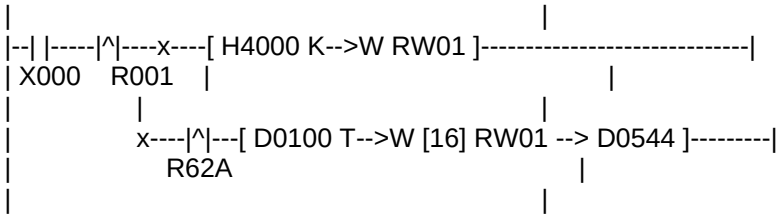
When the M/EX100's memory selection is set to the 3K mode, and the retentive memory area is set to include the registers to be written to EEPROM, the data of 1024 registers (DO512 to D1535) are stored in the EEPROM. (see also 5.2) The EEPROM read and write instructions enable access to the data stored in the user program. This allows storage and read-out of the variable data, providing completely maintenance-free back up operation. By using a step sequencer to write 16 registers per scan, as many registers as desired can be written to the EEPROM by the program while it is in RUN.

NOTE (1) This function is valid only when the M/EX100's memory selection is set to the 3K mode.

(2) See pages the example below for detailed explanations of these instructions.

Sample program (Write data to EEPROM)

. When X000 comes ON, 16 words of the data in D0100 to D0115 are written into D0544 to DO559 of the EEPROM.



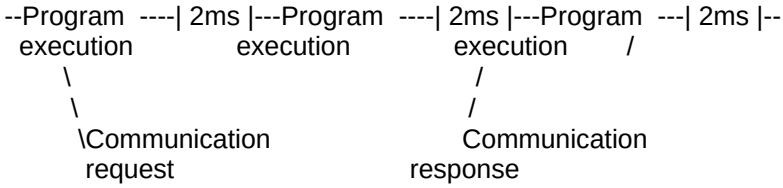
NOTE The EEPROM write instruction functions by combining the transitional contact R62A and FUN003.

9.6 Communication priority mode

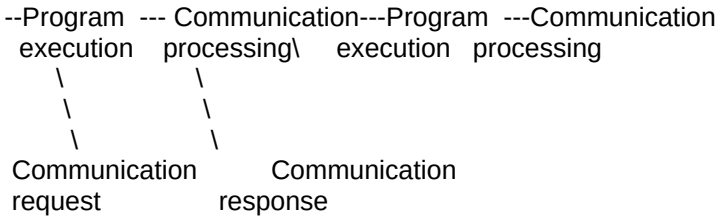
Normally the M/EX100 processes communication requests from the computer link or programmer in a period of 2 ms at the end of each scan. This is done to minimize scan time.

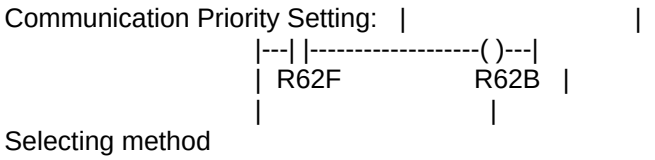
In the communication priority mode, the M/EX100 processes communication requests without a break after program execution. This mode is effective for applications requiring a more rapid response to the computer. (See "Communication priority mode", shown below)

Normal Mode



Communication Priority mode





Selecting method

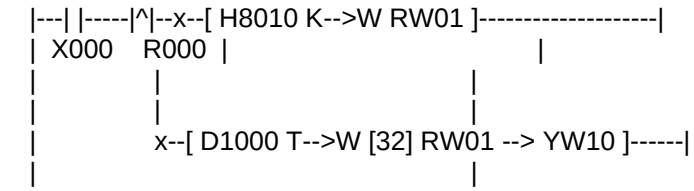
To select the communication priority mode, create the circuit shown above on the first page of the program.

9.7 Data input/output function for special modules

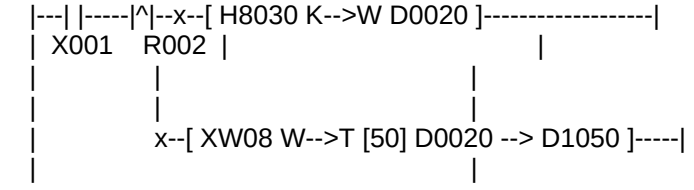
To facilitate use of the special modules, the M/EX100 provides special instructions to exchange a large amount of data between the M/EX100's CPU and special modules such as the motion control module and the ASCII/BASIC module.
Thus, a complicated handshake program is not required for exchanging data.

Sample programs

. When X000 comes ON, 32 registers of data in D1000 to D1031 are transferred to locations beginning at address H10 of the internal memory of the special module that is allocated to YW10.



. When X001 comes ON, 50 registers of data stored beginning with address H30 in the internal memory of the special module (XW08) are read out to D1050 and subsequent registers.



NOTE (1) Up to 128 registers can be input/output at a time. However, if the computer link is used at a transmission rate of 9600 bps, simultaneous input/output capacity will be limited to 64 words or less.

(2) Refer to the user's manuals of the special module for details on the mapping of the internal memory and additional application examples.

****

10. Maintenance and Checks

10.1 Daily checks

Check the following items each day to ensure that the M/EX100 is in proper operating condition.

Item	Check	Corrective measure
MOUNTING	Tightness of the M/EX100 unit's mounting screws.	Tighten screws as necessary.
	Hooks on the I/O modules are securely engaged.	Push the module toward the rack until the hook is securely engaged.
	Detachable terminal blocks are securely engaged.	Secure as necessary.
Connections	Tightness of terminal screws on power cable and I/O wiring.	Tighten screws as necessary.
	Tightness of expansion cable connectors.	Secure connectors as necessary.
Status LEDs	POWER Must be lit when power is on.	
	RUN Must be lit when PLC is in RUN mode.	See Section 11 Troubleshooting procedures.
	CPU Must be lit when the I/O is normal.	
Input status LED's	Must be lit if input is on; off when output is off.	
Output status LED's	Must be lit when output is on; off when output is off.	

10.2 Periodic checks

Check the following items at least once every six months, or when the operating environment changes.

Item	Check	Criterion
Power supply	Check power supply voltage at terminals.	Must be within specified range.
	Check the wiring screws.	Must not be loose.
	Visually check the wires and cables.	Must not be damaged.
I/O	Check the voltage at the I/O terminals.	Must be within specified range.
	Turn on input equipment and check that LEDs light.	Each corresponding LED must light.
	Forcibly turn on output and check that LEDs light.	Each corresponding LED must light.
	Check connection of expansion unit/rack.	Expansion cable must be connected securely.
	Check I/O module mounting.	Each module must be attached securely.
	Check mounting of detachable terminal blocks.	Must not be loose.
	Check wiring screws.	Must not be loose or be in contact with adjoining parts.
	Visually check wires and cables.	Must not be damaged.
	Check terminal block connectors.	Must be cleaned.
Environment	Check temperature, humidity, vibration, and dust levels, etc.	Must be within specifications.
Mounting	Check M/EX100's mounting.	Must not be loose.
Programs	Check program. Compare it with master program, if available.	There must be no program errors.
Optional-battery check	Replace the battery once a year.	Wipe the new battery with a clean, dry cloth and check that it is inserted correctly.

10.3 Spare parts to keep in stock

Keep the following spare parts in stock so that system down time will be minimal in the event of a failure.

. Input and output modules
One of each type of input and output module used should be stocked as a back-up.

. Fuses
Spare fuse for each module used should be kept in stock. The following table lists fuse specifications.

Power Supply Module (EX100)

Module	Fuse rating
PS31	125V 2A, normal fusing
PS51	125V 2A, normal fusing
PS61	250V 1A, normal fusing

Output Modules

Module	Fuse rating
DO31	250V/5A quick fusing
DO31	250V/2A quick fusing
AC61	250V/2A normal fusing

10.4 Installing the optional battery

The optional battery can be installed in the battery compartment behind the M20/40's front cover.

EX2040PBATT
3V
1200 mAh
Replace every 2 years (recommended)

To install/remove the optional battery:

1. Turn off power.

2. Remove the front cover of the M20/40.

3. Remove the old battery, if any.

4. Check for proper pole alignment of the new battery and insert the battery connector.

5. Assure that the battery is securely seated in compartment and replace cover.

CAUTION (1) Do not use manganese or alkaline batteries as substitutes because they cannot supply the proper voltage.

(2) Be careful not to short-circuit the battery's plus and minus terminals.

(3) Do not expose battery to flames, excessive heat, corrosive gases or chemicals.

(4) The battery cannot be recharged.

On the EX100, the battery is in the CPU, which will have to be removed from rack to install.

3V
180 mAh
Replacement period 1 year (recommended)
Part number EX10-ACR2 for the EX100 battery.

CAUTION (1) Do not use manganese or alkaline batteries as substitutes because they cannot supply the proper voltage.

(2) Be careful not to short-circuit the battery's plus and minus terminals.

(3) Do not expose battery to flames, excessive heat, corrosive gases or chemicals.

(4) The battery cannot be recharged.

****

11. Troubleshooting

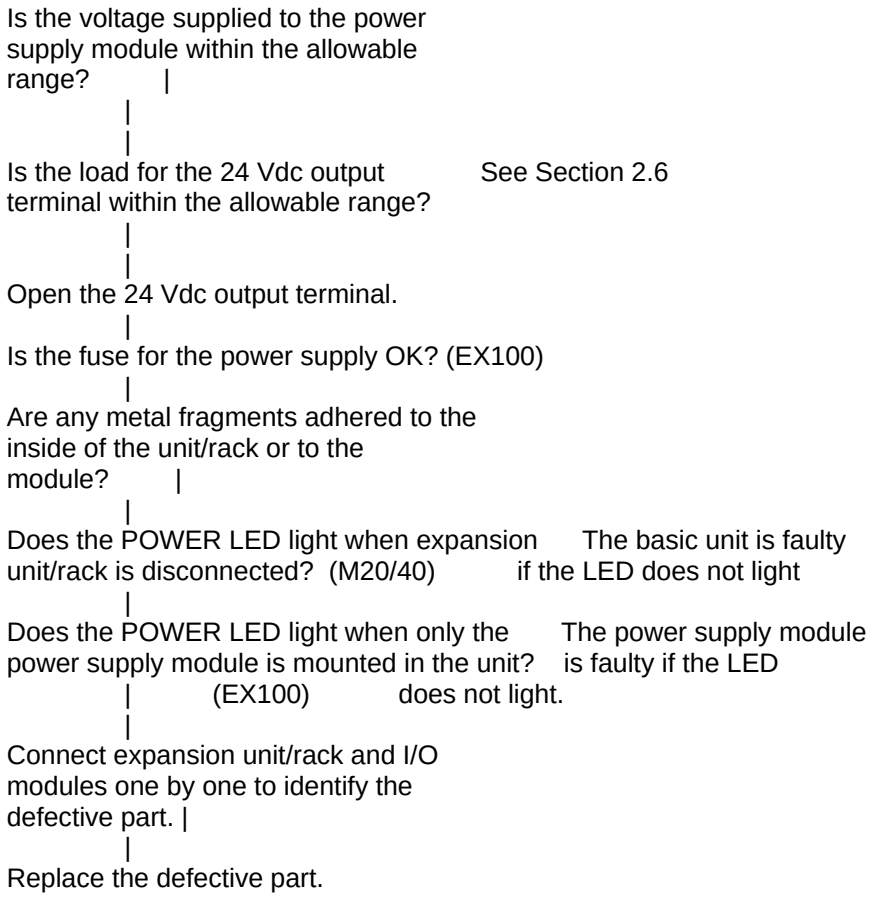
11.1 Troubleshooting procedure

If a system failure occurs, it is important to accurately determine the cause of the trouble. It is of primary importance to first determine whether the problem lies with the machinery or with the controller. In many cases, one problem causes secondary problems. When determining the cause of the initial problem, it is important to consider the system as a whole.

CAUTION When troubleshooting, take the following precautions to prevent human injury and/or damage to the M/EX100:
(1) Disconnect power from the M/EX100 before changing units/modules or disconnecting cables.
(2) Disconnect power to all I/O devices before changing units/ modules.

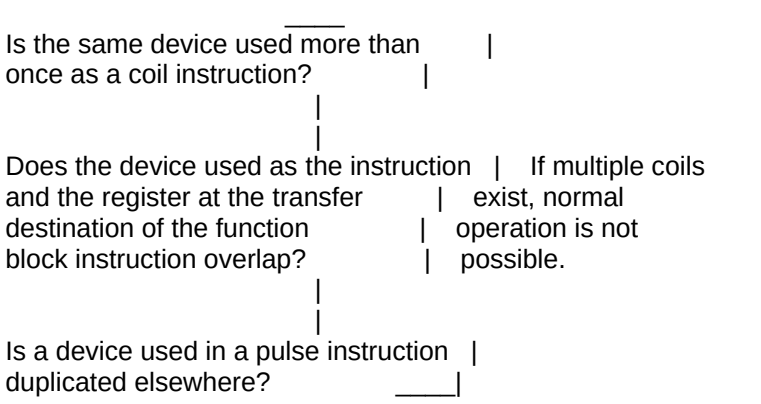
Faulty power supply

The following flowchart lists troubleshooting procedures when the POWER LED does not light after connecting power to the M/EX100.



Faulty operation due to a software problem

If the system runs but the program execution is faulty, there is good reason to suspect that the problem is caused by the user program. Check the following items.



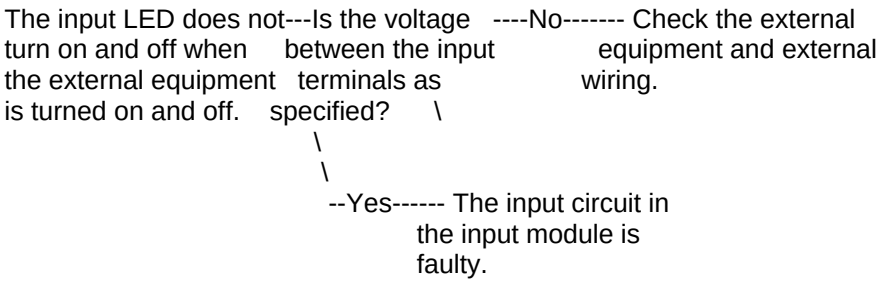
Is an input signal changing faster than the scan time?	Such a signal cannot be processed properly.
Is a timer register used in the timer instruction being duplicated?	The timer will operate abnormally.
Is a counter register used in the counter instruction being duplicated?	The counter will operate abnormally.

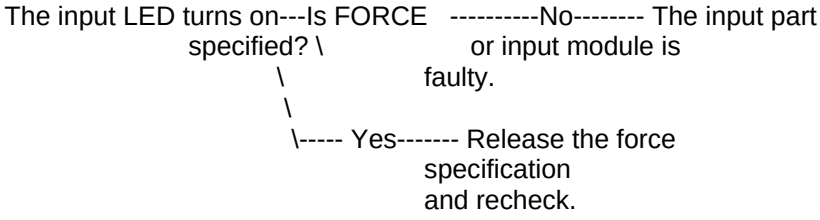
Faulty operation due to a hardware problem

If the input data is not read properly when the system is operating or if it is not possible to output data although the monitor indicates that the output is operating, check the following items to pinpoint the cause.

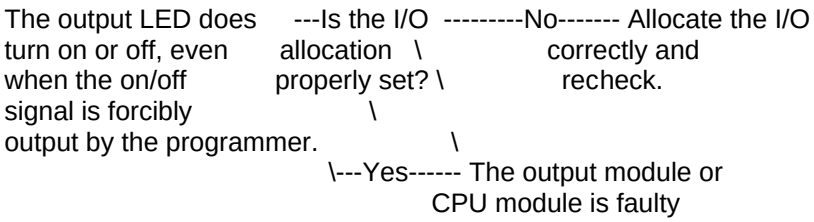
If the input is faulty, is the level of the input signal as specified?	Refer to the I/O specification
If the output is faulty, is the external power supplied as specified?	
Are the I/O registers correctly allocated?	
Check the status of the LEDs for input and output. Does the LED indication correspond to that on the monitor?	See the following check list.

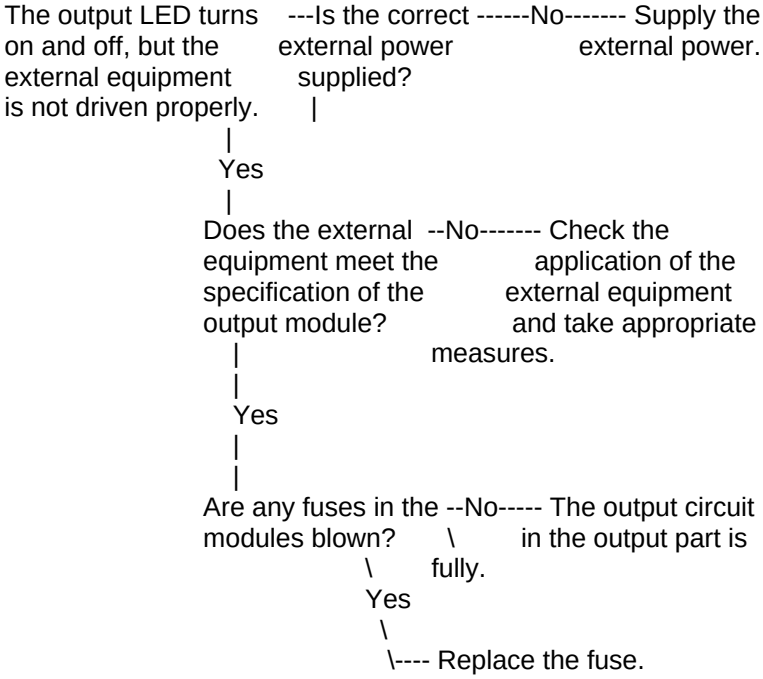
Input check





Output check





11.2 Diagnostic Check List

The following table lists the items checked during the M/EX100's diagnostic check. If the system does not operate properly even when the POWER LED on the power supply module is lit, use the following table as a guide for trouble-shooting.

Item / Special Relay	Phenomenon	Error reg'd	Displayed message
Illegal instruction R636	Illegal instruction detected during execution.	Yes	Illegal Inst.
Watchdog timer R630	The watchdog timer cannot be reset within 350 ms.	Yes	WD-Timer
I/O bus trouble R634	A problem has been detected in the I/O bus, or the CPU is mounted in the wrong slot.	Yes	IO Bus Error
I/O response error (activated by Run switch) R634	An I/O module is not responding.	Yes	IO No Sync
I/O reference error (When activated by a programmer)	An I/O module is not responding.	No	IO No Sync
I/O reference error (activated by Run switch) R635	Incorrect I/O allocation	Yes	IO Unmatch
I/O response error (When activated by a programmer)	Incorrect I/O allocation	No	IO Unmatch
CPU trouble R630	There is a problem in the hardware.	Yes	CPU Error
EEPROM trouble R633	The EEPROM data is abnormal.	Yes	ROM Error
Scan time over R637	A program scan has exceeded 200 ms.	Yes	Scan Over
Program error (When activated by the RUN switch) R636	The program data is abnormal or contains an error.	Yes	-1)
Program error (When activated by a programmer)	The program data is abnormal or contains an error.	No	-1)
Transmission trouble R63B	Communication problem with external device.	No	-2)
TOSLINE trouble R63C	A problem has been detected in the TOSLINE transmission.	No	TL

Computer link trouble R63D	A problem has been detected in the computer link.	No	CL

****

APPENDIX A :Instruction execution times

Instruction FUN No.	Symbol	Execution time (uS) Non-execution	Instruction FUN No.	Symbol	Execution time (S) Non-execution
	-| |-	-- 0.9	031	OR (R-R)	2.5 107
	-|/|-	-- 0.9	032	EOR (R-R)	2.5 107
	-( )-	-- 1.2			
	*-( )	-- 0.9	033	NOT	1.8 100
	-|^|-	-- 1.2	035	RTR	2.5 106+nn+5
	-|,|-	-- 1.2	036	RTL	2.5 106+nnx5
	MCS	-- 0.6	040	AND (R-K)	2.5/3.1 109
	MCR	-- 0.6			
	JCS	-- 0.6	041	OR.(R-K)	2.5/3.1 109
	JCR	-- 0.6	042	EOR.	2.5/3.1 109
	TON	1.8 96	043	TEST	1.8/2.5 98
	TOF	1.8 96	046	NEG	1.8 100
	SS	1.8 96	050	BIN	1.8 194
	CNT	1.8 92	051	BCD1	1.8 125
	END	-- 0.3	052	BCD2	1.8 290
000	W W	1.8 98	053	ENC	1.8 104
001	K W	1.8/2.5 93	054	DEC	1.8 104
002	TINZ	2.5 98 + nn x 5	055	BITC	1.8 178
003	T W	3.1 119	060	UL	2.5 116
004	W T	3.1 119	061	LL	2.5 116
005	T T	2.5 105 + nn x 11	062	MAX	2.5 110 + nn x 9
010	+ (R-R)	2.5 110	063	MIN	2.5 110 + nn x 9
011	- (R-R)	2.5 110	064	AVE	2.5 147 + nn x 9
012	x (R-R)	2.5 168	065	FG	3.1 367 + nn x 37
013	/ (R-R)	2.5 342	070	RT	1.8 413
014	> (R-R)	1.8 100	071	SIN	1.8 666
015	= (R-R)	1.8 100	072	ASIN	1.8 819
016	< (R-R)	1.8 100	073	COS	1.8 674
017	++(R-R)	2.5 125	074	ACOS	1.8 824
018	--(R-R)	2.5 130	080	SET	1.2 90
020	+ (R-K)	2.5/3.1 110	081	RST	1.2 93
021	- (R-K)	2.5/3.1 113	090	DDSP	1.2/1.8 144~176
022	x.(R-K)	2.5/3.1 167	091	DDSM	1.2/1.8 161~189
023	/.(R-K)	2.5/3.1 343	096	IN	1.8 117 + nn x 63
024	> (R-K)	1.8/2.5 98			
025	= (R-K)	1.8/2.5 98	097	OUT	1.8 117 + nn x 43
026	<.(R-K)	1.8/3.5 98	100	STIZ	1.8 105~154
030	AND (R-R)	2.5 107	101	STIN	1.2 98~118
			102	STOT	-- 83~141
			110	F/F	-- 102
			111	U/D	-- 98~116
			112	SR	-- 118~404

****

APPENDIX B: List of modules and types

M20/40 units

Model	Description	Part No.
	12 pts 24 Vdc input/8 pts relay output 100~240 Vac power supply	EX10M20DR5
M20	12 pts 24 Vdc input/8 pts relay output, 24 Vdc power supply	EX10M20DR6
	24 pts 24 Vdc input/16pts relay output, 100~240 Vac power supply	EX10M40DR5
M40	24 pts 24 Vdc input/16 pts relay output, 24 Vdc power supply	EX10M40DR6
	24 pts 120 Vac input/16 pts relay output, 100~240 Vac power supply	EX10M40AR5

Expansion units

Type	Description	Part No.
20 points expansion	12 pts 24 Vdc input/8 pts relay output, No power supply	EX10E20DR5
40 points expansion	24 pts 24 Vdc input/16 pts relay output, No power supply	EX10E40DR5

Expansion racks

Type	Description	Part No.
2-slot rack	2 slots for modules, No power supply	EX10EUB2
4-slot rack	4 slots for modules, No power supply	EX10EUB4

Expansion cables

Description	Part No.
10 cm length expansion cable	EX10PEXPC1
30 cm length expansion cable	EX10PEXPC3

Options

Description	Part No.
Cover for vacant slot	EX10-ABP1
Lithium battery	EX2040PBATT

EX100 I/O modules

Type	Description	Code	Part No.
DC input	16-point dc/ac input (12-24 Vac/Vdc)	DI31	EX10-MDI31
	32-point dc input (24 Vdc)	DI32	EX10-MDI32
AC input	16-point ac input (100-120 Vac)	IN51	EX10-MIN51
	16-point relay output (200-240 Vac)	IN61	EX10-MIN61
Relay output	12-point relay output (240 Vac/24 Vdc)	RO61	EX10-MRO61
	8-point isolated relay output (240Vac/24 Vdc)	RO62	EX10-MRO62
DC output	16-point transistor output (5-24 Vdc)	DO31	EX10-MDO31
	32-point transistor output (5-24 Vdc)	DO32	EX10-MDO32
AC output	12-point triac output (100-240 Vac)	AC61	EX10-MAC61
Analog input	4-ch analog input (4-20 mA/1-5V) (8-bit)	AI21	EX10-MAI21
	4-ch analog input (0-10 V) (8-bit)	AI31	EX10-MAI22
	4-ch analog input (+/-10V) (12-bit)	AI32	EX20-MA132
Analog output	2-channel analog output (5/10 V, 20mA) (8-bit)	AO31	EX10-MAO31
	2-ch analog (4-20mA/1-5V) (12-bit)	AO22	EX10-MAO22
Pulse I/P	1-channel pulse input (5/12V, 100kHz)	PI21	EX10-MPI21
Position	1-axis position control	MC11	EX10-MMC11
ASCII	ASCII/BASIC module	AS11	EX10-MAS11
Transmission	TOSLINE-30 (wire)	LK11	EX10-MLK11
	TOSLINE-30 (optional)	LK12	EX10-MLK12

Peripheral devices

Description	Code	Part No.
Handy programmer	HP100	EX25UHP-100
Data access panel	DP100	EX25UDP-100
Program storage module	RM11	EX10-PRM11
Programming software		EXPDD250-V2
Converter for software		EX25GP232A-TIC2

****

Appendix C: I/O Wiring Diagrams (Terminal Connections)

Due to the complexity of these drawings, they are contained in a special IO_WIRE.DXF (DXF = Drawing Exchange Format). The DXF file can be imported into any CAD program (Autocad, DesignCAD, Turbocad, etc.). The DXF is arranged as follows:

Layer 1: M20/40 input wiring
Layer 2: M20/40 24Vdc power supply & output wiring
Layer 3: Standard EX100 discrete I/O modules
Layer 4: EX100 32 point I/O modules
Layer 5: EX100 analog modules, high speed counter module, and intelligent I/O modules