CL200

Operations List Software Manual

Edition 108

CL200

Operations List Software Manual

1070 072 151-108 (99.12) GB

1 Safety Instructions

Before you start programming and working with the CL200 controller, we recommend that you thoroughly familiarize yourself with the contents of this instruction manual. Keep this manual in a place where it is always accessible to all users.

1.1 Standard Operation

This instruction manual presents a comprehensive set of instructions and information required for the standard operation of the described products. The described products are used for programming and operating the CL200 control unit.

The products described hereunder

- were developed, manufactured, tested and documented in accordance with the relevant safety standards. In standard operation, and provided that the specifications and safety instructions relating to the project phase, installation and correct operation of the product are followed, there should arise no risk of danger to personnel or property.

The prerequisites for trouble-free service and safe operation of the product are proper transport, handling and storage, placement and installation, plus careful operation of the equipment.

1.2 Qualified Personnel

The requirements pertaining to qualified personnel are based on the job specifications as outlined by the ZVEI (central association of the electrical industry) and VDMA (association of German machine and plant builders) professional associations in Germany. Please refer to the following German-language publication:

Weiterbildung in der Automatisierungstechnik
Hrsg.: ZVEI und VDMA
MaschinenbauVerlag
Postfach 71 08 64
60498 Frankfurt

This instruction manual is specifically designed for PLC technicians. They will require specific knowledge of the CL200 controller.

Interventions in the hardware and software of our products which are not described in this instruction manual may only be performed by specially trained Bosch personnel.

Unqualified interventions in the hardware or software or non-compliance with the warnings listed in this instruction manual or indicated on the product may result in serious personal injury or damage of property.

Installation and maintenance of the products described hereunder is the exclusive domain of trained electricians as per VDE 1000-10, who are familiar with the contents of this manual.

Trained electricians are persons of whom the following is true:

- They are capable, due to their professional training, skills and expertise, and based upon their knowledge of and familiarity with applicable technical standards, of assessing the work to be carried out, and of recognizing possible hazards.
- They possess, subsequent to several years' experience in a comparable field of endeavour, a level of knowledge and skills that may be deemed commensurate with that attainable in the course of a formal professional education in this area.

With regard to the foregoing, please read the information about our comprehensive training program. You will find a listing of our seminars on the front inside cover of this instruction manual. The professional staff at our training centre will be pleased to provide detailed information. You may contact the centre by telephone at (+49) 6062 78-258.

1.3 Safety Labels Affixed to Components

Danger: High voltage!

Danger: Battery acid!

Electrostatically sensitive devices!

Disconnect at mains before opening!

Pin for connecting PE conductor only!

Functional earthing / low noise earth

For screened conductor only!

1.4 Safety Instructions in this Manual

DANGEROUS ELECTRICAL VOLTAGE
This symbol is used to warn of the presence of a **dangerous electrical voltage.** Insufficient compliance with or failure to observe this warning may result in **personal injury.**

DANGER
This symbol is used wherever insufficient or lacking compliance with instructions may result in **personal injury**.

CAUTION
This symbol is used whenever insufficient or lacking compliance with instructions may result in **damage to equipment or data files**.

⇒ **This symbol is used to alert the user to an item of special interest.**

1.5 Safety Instructions for the Described Product

DANGER
Fatal injury hazard through ineffective Emergency-OFF safety devices!
Emergency-OFF safety devices must remain effective and accessible during all operating modes of the system.
The release of functional locks imposed by Emergency-OFF devices must never be allowed to cause an uncontrolled system restart! Before restoring power to the system, test the Emergency-OFF sequence!

DANGER
Danger to Personnel and Equipment!
Test every new program before operating the system!

DANGER
Retrofits or modifications may interfere with the safety of the products described hereunder!

The consequences may be severe personal injury, damage to equipment or environmental hazards. Therefore, any system retrofitting or modification utilizing third-party components will require express approval by Bosch.

1.6 Documentation, Version and Trademarks

Documentation

The present instruction manual provides the user with comprehensive information about programming the CL200, and about the instruction set used by the control unit.

List of instruction manuals:

Instruction manual	Language	Order no.
Die Welt der SPS	German	1070 072 407
CL200 Manual	English	1070 072 145
SFC Sequence Function Chart	English	1070 072 186
KETTE200 Software module description	English	1070 072 150
BT-MADAP Software manual	English	1070 072 163
Catalogue, Programmable Logic Controllers	English	1070 072 160

⇒ **Throughout this instruction manual, the floppy disk drive shall always have drive letter A:, and the hard disk drive shall have drive letter C:.**

Special keys or keyboard shortcuts (key combinations) are enclosed in pointed brackets:

- Special keys, example: <Enter>, <PgUp>, <Del>
- Key combination (pressed simultaneously), e.g.: <Ctrl> + <PgUp>

Trademarks

All trademarks referring to software that is installed on Bosch products when shipped from the factory represent the property of the respective manufacturers.

When shipped from the factory, all installed software is protected by copyright. It may therefore be duplicated only with prior permission by Bosch or in accordance with the licensing agreements with the respective manufacturer or copyright owner.

MS-DOS® and Windows™ are registered trademarks of Microsoft Corporation.

2 Introduction

2.1 Programming Differences between “PROFI” and “WinSPS” Software

The present documentation discusses the representation of constants, program module calls, and jump instructions in the notation generated by the “WinSPS” programming unit software. However, when processed with the “PROFI” programming unit software (an earlier, DOS-based programming device application), the representation of constants, program module calls and jump instructions will differ to some extent.

The referred differences are exemplified in the following side-by-side comparison:

Differences in programming and notation of word constants

Data type		PLC utility programs	
Explanation	**Notation**	**PROFI**	**WinSPS**
UINT (unsigned integer)	Binary / Dual	K00000000 00000000B K11111111 11111111B	2#0000000000000000 2#1111111111111111
	Decimal, word	K00000D - K63535D	00000 - 65535
	Decimal, byte / byte	K000/000 - K255/255	Not defined in IEC1131 Part 3
	Hexadecimal	K0000H - KFFFFH	16#0000 - 16#FFFF
INT (integer)	Decimal, word	K-32768 - K+32767 K-32768D - K+32767D	-32768 - +32767
Text, STRING(2)	ASCII	K'AB’	‘AB’
Time value, TVALUE	Time val. (+timebase r) r: 0=10ms, 1=100ms 2=1s, 3=10s	K0.r - K1023.r	T#10ms - T#10230s T#0.r - T#1023.r

Differences in programming and notation of module calls

	PLC utility programs	
	PROFI	**WinSPS**
Program module / function call (IEC1131-3)	CM PM	CM FC

Differences in programming and notation of jump instructions

	PLC utility programs	
	PROFI	**WinSPS**
Jump instruction	JPx -label	JPx label
Jump destination	-label	label:

3 Memory Management & Peripheral Operation

In their standard configuration, the CL200 central processing units are equipped with RAM and FLASH RAM memory modules. The application software used by the programming unit determines the various memory modes available for controller operation.

3.1 Memory Modes

RAM Mode

All data is stored in the volatile read-write memory, which requires a backup battery top ensure data security. The battery also powers the buffers used by remanent markers, times, counters and the data field.

While operating in RAM Memory mode, program modules and current data modules can be copied from RAM and written directly to the internal FLASH memory, and to the MemoryCard. This ensures that, subsequent to a battery failure, the backed-up data can again be copied into RAM from either FLASH EPROM or MemoryCard.

To define this mode of operation, the software of the programming unit (PG) tags all modules with the “R” identifier (RAM).

EPROM Mode

All data is stored in non-volatile FLASH memory, which does not require a buffer battery to ensure data security. In EPROM Mode, the buffer battery serves only to protect the remanent areas – and the data field, which is always stored in internal RAM – against a sudden voltage loss.

In the event that, due to battery failure or operation without battery, the absence of an executable PLC program in RAM is detected upon startup, unconditional loading from FLASH memory into RAM will occur. While stored in RAM, the program is executed, utilizing all benefits intrinsic to RAM Mode operation. It should be considered, however, that in the case of battery-less operation or in the event of a buffer failure, each Off/On cycling of the mains power will cause the data module contents stored in FLASH memory to be regenerated.

The battery-less operation that is possible in EPROM Mode requires that no remanence is used and that the data field not contains data requiring buffering.

⇒ **For reasons of compatibility with other Bosch PLCs, this memory mode is designated EPROM Mode, although the non-volatile memory used here is a FLASH memory module.**

To define this mode of operation, the software of the programming unit (PG) tags all modules with the “E” identifier (EPROM).

Mixed Memory Mode

This memory mode is set by the software of the programming unit in cases where individual modules are defined as RAM, and other, tested modules are defined as EPROM modules. The essential advantage of this mode of operation lies in the doubling of the memory capacity available to the PLC program. As a prerequisite for Mixed Memory Mode, a battery for buffering the RAM area must always be present.

In addition, the following items should be noted:

- Mixed Memory Mode can only be managed directly via the programming unit (backing up to the MemoryCard is not possible).
- After a battery failure, reloading via the programming unit is required.
- Attempts to save to FLASH memory will be rejected.

To select Mixed Memory Mode, the software of the programming unit defines the memory area in which the individual modules are stored:

“R” identifier: storage in RAM

“E” identifier: storage in FLASH

3.2 Operating Peripherals

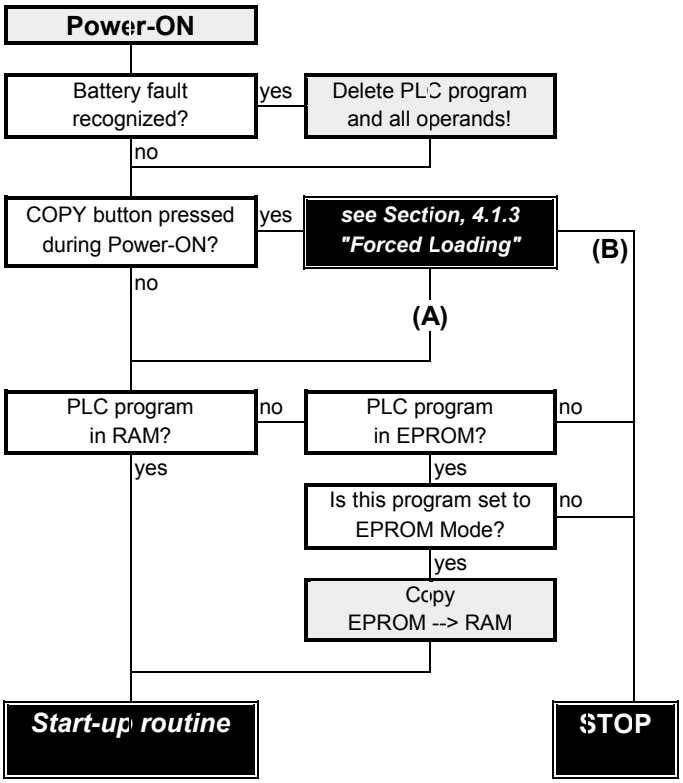
Following the end of the program, the operation of peripherals progresses only up to the highest address that is hardware-equipped. This means that the I/O state is optimized automatically during the program execution interval. To effect a further optimization of the I/O state, the number of times and counters in the OM2 initialization module can be reduced to the required minimum.

4 Startup Modes

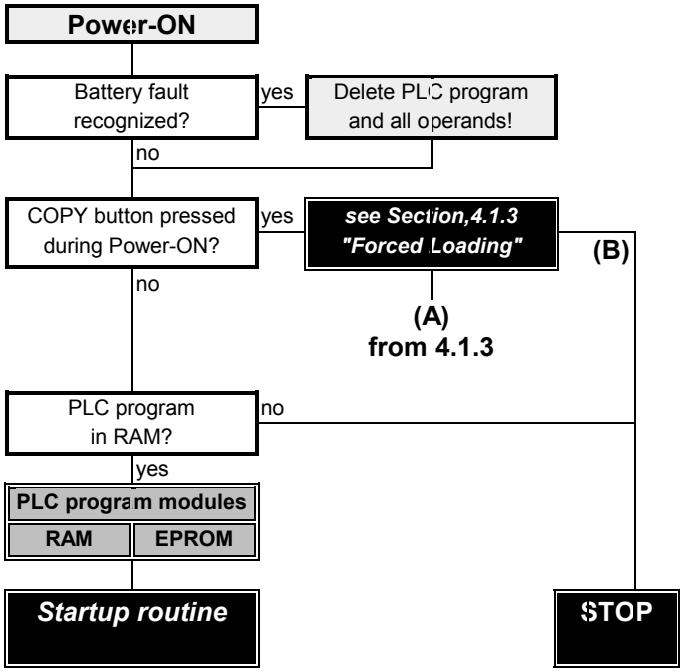
4.1 System Startup & Backup Copies

In the event that errors are recognized during the system startup or copying routine, an appropriate error message indicating the cause of the fault will be generated. This message can be displayed by means of the programming unit.

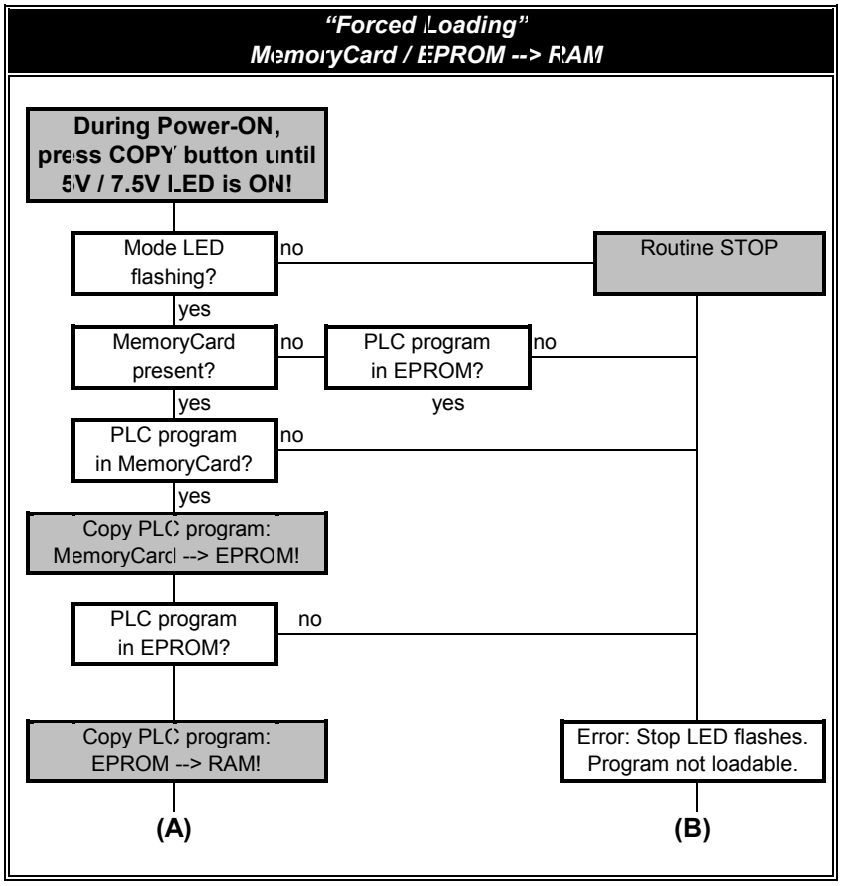
4.1.1 System Startup in RAM and EPROM Memory Modes



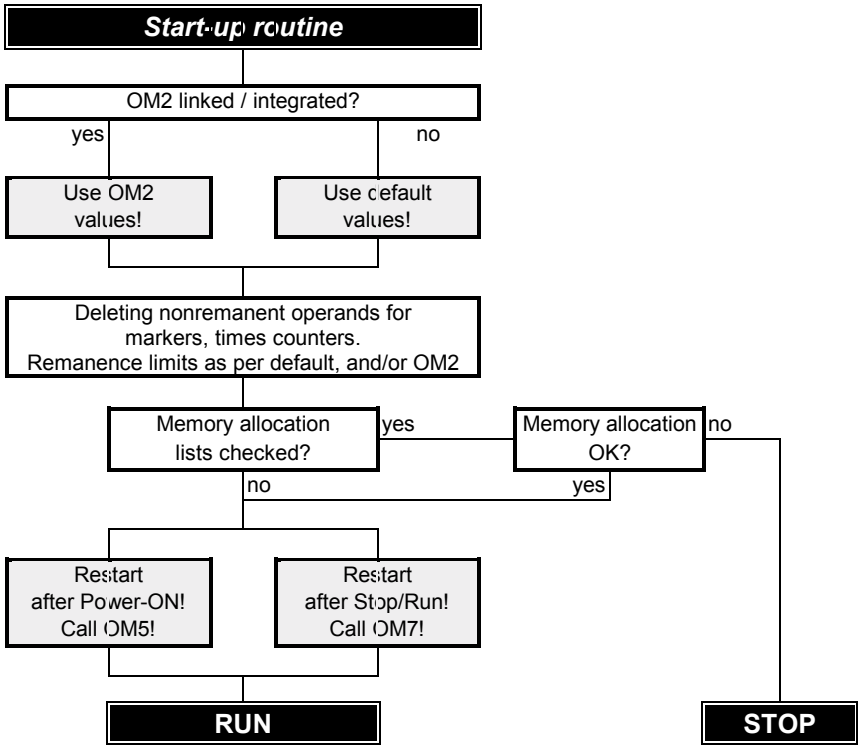
4.1.2 System Startup in Mixed Memory Mode



4.1.3 Forced Loading of MemoryCard and/or EPROM Contents into RAM

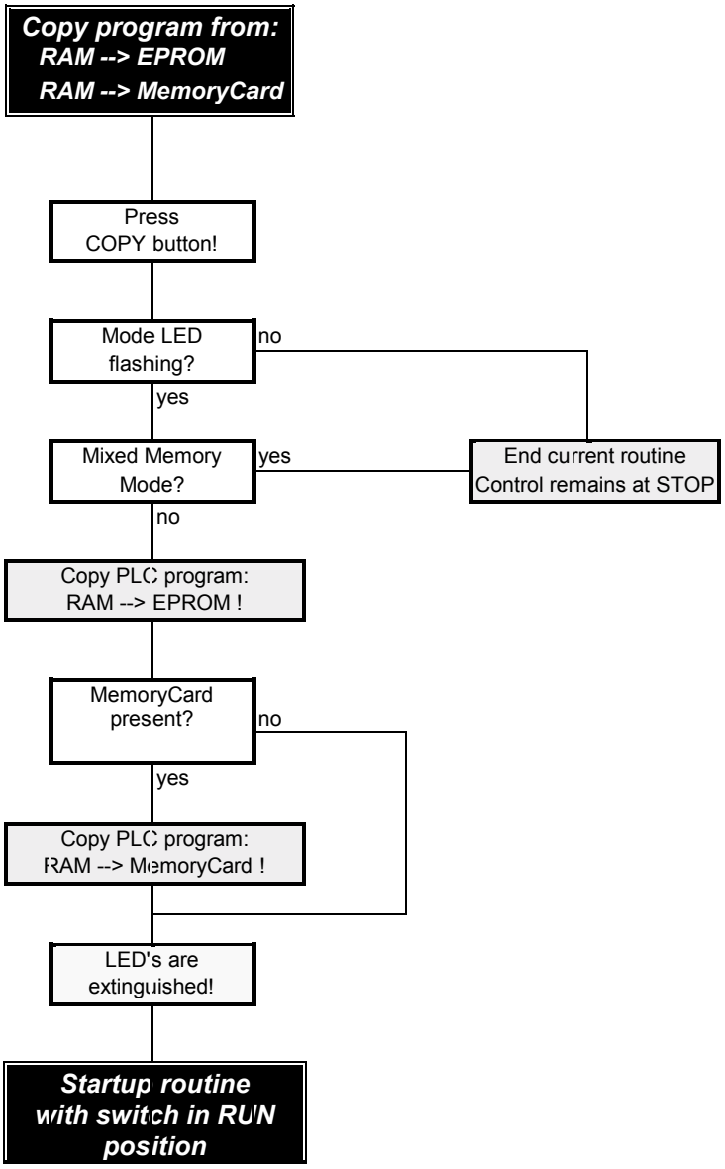


4.1.4 Startup Routine following Mode-dependent System Start



4.1.5 Backing up Programs from RAM to EPROM or MemoryCard

⇒ **Notes on handling the MemoryCard:**
- **MemoryCard must be firmly seated in slot!**
- **Positive contact must be ensured.**



5 Status Messages on the NT200 Power Supply

The following status messages are indicated by LED's on the NT200 power supply module:

The "**5V / 7.5V"** and "**Battery Low"** LED's are directly controlled by the power supply module. They provide information about the supply status of the different voltages as well as possible buffer errors.

◎ **"5V/7.5V"**

Green LED	ON	Power supply OK.

◎ **"Battery Low"**

Red LED	ON	Buffer error batteryless operation

The "**PG Mode"** and "**Stop"** LED's are controlled by the central processing unit, and indicate various system statuses of the controller.

◎ **"PG Mode"**

Red LED	Steady ON:	Outputs disabled, and/or inputs/outputs fixed.
	Flashing (2 Hz)	Copying: RAM → EPROM RAM → MemoryCard, via programming unit or COPY button pressed on NT200 power supply. MemoryCard → EPROM during startup.

◎ **"Stop"**

Red LED	Steady ON:	Controller is in STOP mode (LED extinguishes in RUN mode).
	Flashing (2 Hz)	Copy error: MemoryCard → EPROM, during startup, with MemoryCard installed. RAM → MemoryCard RAM → EPROM

Additional system messages:

"PG Mode" and **"Stop"** LED's

Simultaneous flashing (2 Hz)	No firmware loaded
Alternating flashing (2 Hz)	Internal error → Use programming unit to obtain detailed information
Rapid flashing (8 Hz)	Hardware fault → Replace ZE200xx

6 Programming Basics

Programmable Logic Controllers execute a machine program describing the tasks to be performed by the controller. To do this, a special programming language is used which may be displayed and printed out via various methods of representation or notations.

6.1 Representation Methods

Instruction List (IL)

Structure of controller instructions

Controller Instruction			
Operation part	Operand attribute	Source operand	Destination operand
OPP	OPA	SRC ,	DEST

Examples:

```
A                    I0.0
A       W            -Name   ,   A
L       BY           O0      ,   B
T       W            C       ,   M10
MUL     W            K1234D  ,   D
```

Ladder Diagram (LD)

When using the LD representation method, the controller tasks are described by means of standard circuit diagram symbols.

Function Diagram (FUD)

When using the FUD representation method, a graphical symbol display (flow chart) illustrates the logical links.

Sequential Function Chart (SFC)

The SFC represents a graphical programming interface, which is used to describe the sequentially processed machine tasks in the form of a cascade sequence. Before it can be loaded into the PLC, this representation is then translated into the executable IL programming language.

6.2 Program Structure

To attain a clear PLC program structure which is easily readable, Bosch uses a consistent approach to structured programming for its programmable logic controllers, i.e., the programs are divided into functionally interconnected program segments or modules. To support the referred structuring, several module types performing various special functions are available.

6.3 Module Types

The controllers utilize the following module types:

Organization modules

Program modules

Data modules

All modules are enabled by being invoked and/or activated in the course of program processing. Such a module call may occur either unconditionally or contingent on a binary link, on the result of a Compare function, and/or on an arithmetical operation.

6.3.1 Organization Modules (OM)

The organization modules perform all administrative, or management functions for the controller program. Although they are programmed in the same manner as the program modules, only the system program invokes organization modules. All organization modules make use of the full instruction set of the PLC. There is no limitation to module size.

Organization modules may be divided into 6 function groups:

OM1 Module which is cyclically called by the system program, and which may be utilized as a distribution module for the entire program.

OM2 Non-executable definition module (initialization table), in which specifications (remanence limits) for the controller system are declared by modifying certain entries.

OM5, OM7 Startup modules for processing a variety of program sequences during a controller power-up or restart.

OM9 Error module which processes responses to program errors or fault conditions.

OM10-OM12 Interrupt module for immediate responses to peripheral events.

OM18 & OM19 Time-controlled processing (time matrix definable in OM2).

To ensure subsequent processing of the input/output cycle (I/O state), the OM1 must always be concluded with an End Of Program (EP) instruction. With the exception of OM2, and dependent upon the specific tasks to which they are assigned, the remaining organisation modules may be concluded with either the EP instruction or with End Of Module (EM). For programming the OM9 error module, it is useful to insert a definite HALT instruction (HLT) into the program immediately following the error response.

6.3.2 Program Modules (FC)

The program modules (FC) contain program segments that are technically and functionally interrelated. From within program modules, any number of additional program modules and data modules may be called. In addition, all program modules have access to the entire command set of the PLC. The modules are not subject to a size limit.

As a rule, program modules are concluded with an End of Module (EM) instruction. If the End of Program (EP) instruction is used, the program will be aborted immediately after the instruction has been processed, the input/output cycle activated, and further program processing again commence with the OM1 organization module.

Due to the option of parameterization, the program modules may be written independently of absolute operands. During the module call-up, the operands required for the current processing task are transferred to the program module in the form of parameter values.

The following input and output parameters may be specified:

- Input parameters: Operands, constants and modules
- Output parameters: Operands

6.3.3 Data Modules (DM)

The data modules (DM) serve as storage areas for all fixed and variable values and text blocks that are used by the program. Therefore, during PLC program processing, there exists the option of always keeping two data modules enabled, each of which provides up to 512 bytes of memory capacity.

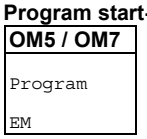
The following applies to the processing of data modules:

- **Before their respective data may be accessed, the data modules must be enabled from within the program by means of module call instructions (i.e., CM for the 1st DM, and CX for the 2nd DM).**
- **Within a given organization module (OM) or program module (FC), the data modules remain current until other data modules are enabled by the program.**
- **After the return to the primary module, the data modules active at the time of the call-up of the base module are again activated.**
- **When the OM1 (cyclical program processing), and the start-up modules OM5 and OM7 are called, no data module is active.**

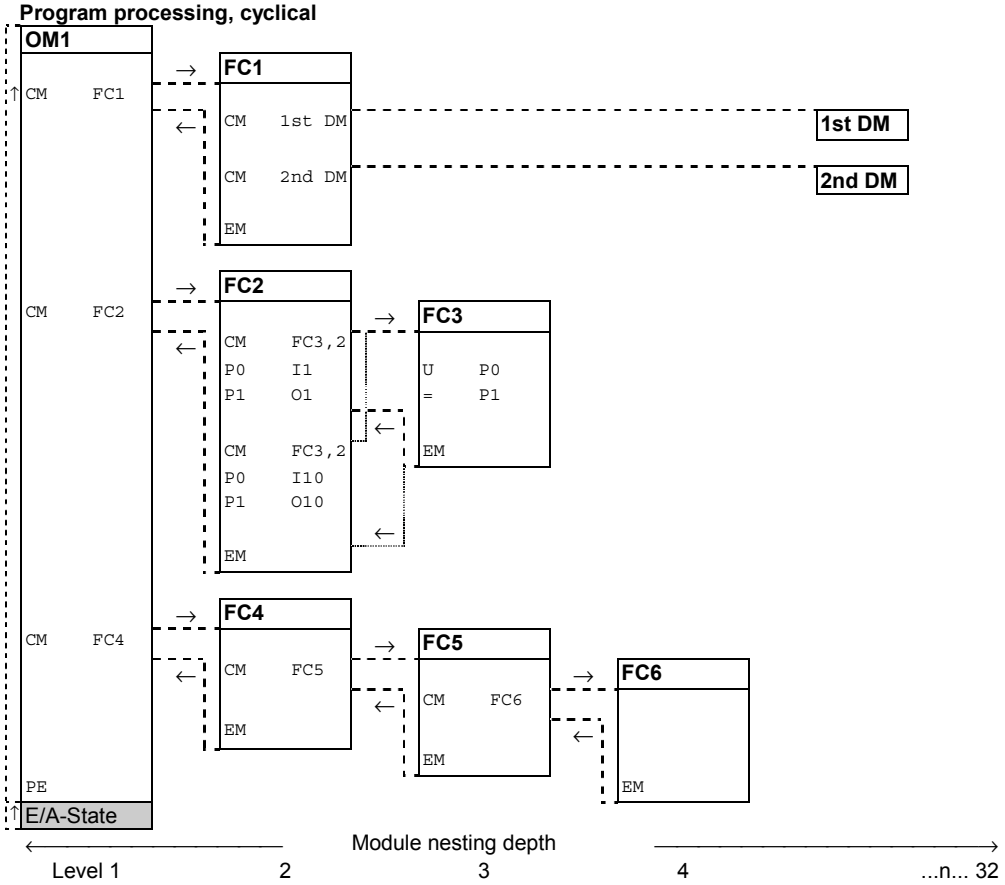
6.4 Application Program Structure

With the aim of providing a clear overview of the basic organization of program management, the following diagram shows an example of the program structure.

Program start-up, one-time only

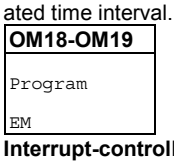


Program processing, cyclical



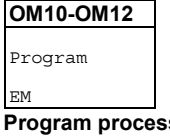
Time-controlled program processing

Processing always commences subsequent to the change of module (not module call) that follows the expiry of the associated time interval.



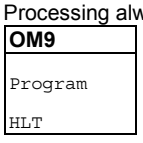
Interrupt-controlled program processing

Processing always commences immediately after the triggering criterion (branching flag) has been detected.



Program processing subsequent to PGM error

Processing always commences immediately after the program error has been detected.



6.5 OM2 Initialization Module

The OM2 initialization module comprises a system initialization table that is linked with the PLC program.

The OM2 determines the settings for the following:

- **Monitoring functions**
- **Remanence limits**
- **Time-controlled organization modules**
- **Onboard counter and onboard analog I/O**
- **Peripherals assignment**

Upon Power-ON and/or pressing the Stop/Run button, and prior to the execution of a startup OM which may be present, the settings reflected in the OM2 are accepted by the system and partially copied into the system area.

The following printout of an OM2 exemplifies all options of exercising control over the system initialisation:

```
;*****************************************************************************
;***                                                                       ***
;***           I N I T I A L I Z A T I O N   T A B L E                     ***
;***                                                                       ***
;***                              C L 2 0 0                                ***
;***                                                                       ***
;*****************************************************************************
;*** Last change: 03. 12. 1996                                            ***
;*****************************************************************************
;
;*****************************************************************************
; OM2 : CL200 Initialization table
;*****************************************************************************
;
;    - must be integrated in every user program which
;      uses different default settings
;
;    - if no OM2 entry in the symbol file is made,
;      the default settings will be used
;
;    I M P O R T A N T   N O T E , please observe in any case
;    ======================================================
;
;    EVERY change of data words (W) in forbidden address ranges
;    =====
;    can result in undefined sytem performance of the PLC.
;
;*****************************************************************************
;

;DW 1:      (reserved)
;-----------------
DEFW W       0

;DW 2:      Initialization flag (entries permitted)
;-------------------------------------------
;           Entry 0 = Function  n o t  checked or executed
;           Entry 1 = Function checked or executed
;
DEFW W       2#0000000000000000
;             **********|**|*|            *: not used
;                       |  | +------      Check configuration list
;                       |  |
;                       |  +--------      Check nominal cycle time
;                       |
;                       |
;                       +-----------       Disable cycle time monitoring
;                                   during start-up (OM5 or OM7)
;

;DW 3: (reserved)
;----------------
DEFW W       0
```

```
;DW 4:      Maximum cycle time (entries permitted)
;-------------------------------------------
;           Entries as multiples of the time base 1 ms of K1D and K2000D
;           (1 ms - 2000 ms) for cycle time monitoring.
;           Function execution at DW1 / Bit 2 = 1.
;
DEFW W      2000
;

;DW 5:      Number of highest timer loop (entry permitted)
;-----------------------------------------------------
;           Entries from K0D through K127D are possible.
;           K10D  = timer loops T0 - T10 exist in PLC program
;           K127D = any timer loops T0 - T127 exist
;
DEFW W      127
;

;DW 6:      Number of first remanent timer (entry permitted)
;-----------------------------------------------------
;           Entries from K0D through K128D are possible
;           K032D = Remanency for timer loops T32 - T127
;           K128D = no remanency
;
DEFW W      64
;

;DW 7:      Number of first remanent counter (entry permitted)
;-----------------------------------------------------
;           Entries from K0D through K64D are possible
;           K32D = Remanency for counters C32 - C63
;           K64D = no remanency
;
DEFW W      32
;

;DW 8:      Number of first remanent marker (entry permitted)
;-----------------------------------------------------
;           Entries from K0D through K192D are possible
;           K128D = Remanency from marker byte M128/marker bit M128.0,
;                   definition of remanency boundary via byte addresses
;           K192D = no remanency
;
DEFW W      96
;
;

;           Definition of Timer OMs (entries permitted)
;           ===========================================
;           Entries as multipliers of time base 10 ms of K1D - K65535D
;           e.g. K0   = no timer-based processing
;                K11D = 11 x 10 ms = 110 ms interval of processing time
;
;DW 9:      Timer OM18
;-----------------
DEFW W      0

;DW 10: Timer OM19
;-----------------
DEFW W      0

;DW 11: (reserved)
;-----------------
DEFW W      0

;DW 12: (reserved)
;-----------------
DEFW W      0
;
```

```
;           Definition of on-board counter (OC) (entries permitted)
;           ======================================================
;           Entry 0 = Function  n o t  existing or executed
;           Entry 1 = Function existing or executed
;DW 13: On-board Counter 0 settings (OC0)
;---------------------------------------
DEFW W      2#0000000000000000
;         *******|*****|||       *: not used
;                      |     |++------ Definition of transitions
;                      |     |              00 no       transition
;                      |     |              01 positive transitions
;                      |     |              10 negative transitions
;                      |     |              11 both     transitions
;                      |     +-------- allow external up/down switch-over
;                      |
;                      +-------------- Count downward

;DW 14/15: Actual value OC0  Low/High word
;-----------------------------------------
;
DEFW W      0
DEFW W      0

;DW 16/17: Nominal value1 OC0  Low/High word
;-----------------------------------------
;
DEFW W      16#FFFF
DEFW W      16#FFFF

;DW 18/19: Nominal value2 OC0  Low/High word
;-----------------------------------------
;
DEFW W      16#FFFF
DEFW W      16#FFFF
;

;DW 20: Onboard Counter 1 settings (OC1)
;---------------------------------------
;
DEFW W      2#0000000000000000
;         |******|*****|||         *: not used
;               |      |     |++------      Definition of transitions
;               |      |     |               00 no       transition
;               |      |     |               01 positive transitions
;               |      |     |               10 negative transitions
;               |      |     |               11 both     transitions
;               |      |     +--------      allow external up/down switch-over
;               |      |
;               |      +--------------      Count downward
;               |
;               +----------------------      incremental rotary position encoder

;DW 21/22: Actual value OC1    Low/High word
;-----------------------------------------
;
DEFW W      0
DEFW W      0

;DW 23/24: Nominal value1 OC1  Low/High word
;-----------------------------------------
;
DEFW W      16#FFFF
DEFW W      16#FFFF

;DW 25/26: Nominal value2 OC1  Low/High word
;-----------------------------------------
;
DEFW W      16#FFFF
DEFW W      16#FFFF

;DW 27: Number of output byte
;----------------------------
;    Outputs are set automatically when nominal values are reached.
;    This address can be used to define the output byte used for
;    this purpose.
;    e.g. K0010D
;                   Nominal value1 OC0  Bit 10.0
;                   Nominal value2 OC0  Bit 10.1
;                   Nominal value1 OC1  Bit 10.2
;                   Nominal value2 OC1  Bit 10.3
DEFW W      0

;DW 28 - DW 30 (reserved for fast counters)
;-----------------------------------------
DEFW W      0       ;DW 28
DEFW W      0       ;DW 29
DEFW W      0       ;DW 30
```

```
;DW 31 Number of analog inputs used (entry permitted)
;---------------------------------------------------
;
;    Entry of K0 through K4 possible
;      K0 = no analog input enabled
;      K2 = two analog inputs enabled (channel 0 and 1)

DEFW W      4

;DW 32 Selection of normalized analog channels (entries permitted)
;-----------------------------------------------------------------
;    Entry 0 = Function  n o t  existing or executed
;    Entry 1 = Function existing or executed
;
; The selected analog inputs are normalized to          2V  - 10V,
; and the enabled analog output is normalized to 2V  - 10V
; or to                                                    4mA - 20mA,
; respectively.

DEFW W      2#0000000000000000
;         *******|****||||       *: not used
;                      |   |||+------ Analog input       Channel 0
;                      |   ||+-------                     Channel 1
;                      |   |+--------                     Channel 2
;                      |   +---------                     Channel 3
;                      |
;                      +-------------- Analog output      Channel 0
;
; Entry K0 disables the normalization

;DW 33 - DW 35 (reserved)
;------------------------

DEFW W      0       ;DW 33
DEFW W      0       ;DW 34
DEFW W      0       ;DW 35

;DW 36: Entries of second serial interface (entries permitted)
;-------------------------------------------------------------
;    Entry 0 = Function  n o t  existing or executed
;    Entry 1 = Function existing or executed
;
DEFW W      2#0000000101110100
;          |*****||*|||*||*       *: reserved
;          |     || |||| +------- Data bits
;          |     || |||| |          0   8 bit
;          |     || |||| |          1   7 bit
;          |     || |||| |
;          |     || |||  +-------- Parity
;          |     || |||             0   odd
;          |     || |||             1   even
;          |     || |||
;          |     ||  +++---------- Baudrate
;          |     ||                111  19200 Baud
;          |     ||                110   9600 Baud
;          |     ||                101   4800 Baud
;          |     ||                100   2400 Baud
;          |     ||                011   1200 Baud
;          |     ||                010    600 Baud
;          |     ||                001  reserved
;          |     ||                000  reserved
;          |     ||
;          |     ++--------------- select protocol
;          |                         00  no protocol
;          |                         01  BUEP19E
;          |                         10  BUEP03E
;          |                         11  reserved
;          |
;          +---------------------- set new entries

;DW 37 - DW 40 (reserved)
;------------------------

DEFW W      0       ;DW 37
DEFW W      0       ;DW 38
DEFW W      0       ;DW 39
DEFW W      0       ;DW 40
```

```
;           Definition of peripheral configuration lists
;           (entries permitted)
;           ========================================
;           The configuration lists are used to enter the peripheral bytes
;           which are configured in the CL200 and for which a configuration
;           test shall be performed (see DW2, Bit 0). A test for mimimal
;           configuration is made upon start-up. Any additional
;           configuration is not checked.
;           Each I/O and EI/EO byte which is configured in the controller
;           and which shall be tested is marked with a "1" in the
;           corresponding data word. Byte which are not configured or shall
;           not be tested are to be marked "0".
;           16 I/O or EI/EO bytes are to be marked in each data word.
;

;           Input configuration list
;           ------------------------
;
;DW 41: I-Byte     15 ..... 0
;---------------------------
DEFW W       2#0000000000000000
;
;DW 42: I-Byte     23 .... 16
;---------------------------
DEFW W       2#0000000000000000
;
;DW 43 - DW 48 (reserved)
;------------------------

DEFW W       0       ;DW 43
DEFW W       0       ;DW 44
DEFW W       0       ;DW 45
DEFW W       0       ;DW 46
DEFW W       0       ;DW 47
DEFW W       0       ;DW 48


;           Output configuration list
;           -------------------------
;
;DW 49: O-Byte     15 ..... 0
;---------------------------
DEFW W       2#0000000000000000
;
;DW 50 - DW 56 (reserved)
;------------------------

DEFW W       0       ;DW 50
DEFW W       0       ;DW 51
DEFW W       0       ;DW 52
DEFW W       0       ;DW 53
DEFW W       0       ;DW 54
DEFW W       0       ;DW 55
DEFW W       0       ;DW 56
```

```
;           Extended Input Range configuration list
;           ---------------------------------------
;
;DW 57: EI-Byte  15 .... 0
;-------------------------
DEFW W      2#0000000000000000
;
;DW 58: EI-Byte  31 ... 16
;-------------------------
DEFW W      2#0000000000000000
;
;DW 59: EI-Byte  47 ... 32
;-------------------------
DEFW W      2#0000000000000000
;
;DW 60: EI-Byte  63 ... 48
;-------------------------
DEFW W      2#0000000000000000
;
;DW 61: EI-Byte  79 .... 64
;-------------------------
DEFW W      2#0000000000000000
;
;DW 62: EI-Byte  95 .... 80
;-------------------------
DEFW W      2#0000000000000000
;
;DW 63: EI-Byte 111 .... 96
;-------------------------
DEFW W      2#0000000000000000
;
;DW 64: EI-Byte 127 ... 112
;-------------------------
DEFW W      2#0000000000000000


;           Extended Output Range configuration list
;           ----------------------------------------

;DW 65: EO-Byte  15 .... 0
;-------------------------
DEFW W      2#0000000000000000
;
;DW 66: EO-Byte  31 ... 16
;-------------------------
DEFW W      2#0000000000000000
;
;DW 67: EO-Byte  47 ... 32
;-------------------------
DEFW W      2#0000000000000000
;
;DW 68: EO-Byte  63 ... 48
;-------------------------
DEFW W      2#0000000000000000
;
;DW 69: EO-Byte  79 .... 64
;-------------------------
DEFW W      2#0000000000000000
;
;DW 70: EO-Byte  95 .... 80
;-------------------------
DEFW W      2#0000000000000000
;
;DW 71: EO-Byte 111 .... 96
;-------------------------
DEFW W      2#0000000000000000
;
;DW 72: EO-Byte 127 ... 112
;-------------------------
DEFW W      2#0000000000000000
```

```
;           !!!     Internal system memory data   !!!
;           ==========================================
;
;        The following default settings must not be changed.
;        ===================================================

;Default for data words DW 73 - DW 128 = K0
;-----------------------------------------
DEFW W       0              ;DW73
DEFW W       0              ;DW74
DEFW W       0              ;DW75
DEFW W       0              ;DW76
DEFW W       0              ;DW77
DEFW W       0              ;DW78
DEFW W       0              ;DW79
DEFW W       0              ;DW80
DEFW W       0              ;DW81
DEFW W       0              ;DW82
DEFW W       0              ;DW83
DEFW W       0              ;DW84
DEFW W       0              ;DW85
DEFW W       0              ;DW86
DEFW W       0              ;DW87
DEFW W       0              ;DW88
DEFW W       0              ;DW89
DEFW W       0              ;DW90
DEFW W       0              ;DW91
DEFW W       0              ;DW92
DEFW W       0              ;DW93
DEFW W       0              ;DW94
DEFW W       0              ;DW95
DEFW W       0              ;DW96
DEFW W       0              ;DW97
DEFW W       0              ;DW98
DEFW W       0              ;DW99
DEFW W       0              ;DW100
DEFW W       0              ;DW101
DEFW W       0              ;DW102
DEFW W       0              ;DW103
DEFW W       0              ;DW104
DEFW W       0              ;DW105
DEFW W       0              ;DW106
DEFW W       0              ;DW107
DEFW W       0              ;DW108
DEFW W       0              ;DW109
DEFW W       0              ;DW110
DEFW W       0              ;DW111
DEFW W       0              ;DW112
DEFW W       0              ;DW113
DEFW W       0              ;DW114
DEFW W       0              ;DW115
DEFW W       0              ;DW116
DEFW W       0              ;DW117
DEFW W       0              ;DW118
DEFW W       0              ;DW119
DEFW W       0              ;DW120
DEFW W       0              ;DW121
DEFW W       0              ;DW122
DEFW W       0              ;DW123
DEFW W       0              ;DW124
DEFW W       0              ;DW125
DEFW W       0              ;DW126
DEFW W       0              ;DW127
DEFW W       0              ;DW128
;*****************************************************************************
EM
```

6.6 Reference List

In the program memory, three data words per module are reserved for the reference list.

The entries for a given module are structured as follows:

Word 0	Address offset of first instruction, and/or of first data word.
Word 1	Number of memory segment
Word 2	Module size in words, exc. module header

A module that is available in the CL200 is identified by these entries.
For modules that are not available, each word contains the entry FFFFH.

The reference list is structured as follows:

Start address (byte)		
Segment	Offset	Module
2	0050H	FC0
2	064AH	FC255
2	0650H	DM0
2	0C4AH	DM255
2	0C50H	OM0
2	0CC2H	OM19

In the PLC program, for example, the reference list entries can be used to check whether modules are present or available and, helpful in the case of data modules, to check the size of a module.

For the aforementioned purposes the CL200 uses the "LIMR" (Load IMage Range) instruction, which is not associated with any other function.

Example:

```
;To check if DM120 is present/available through a
;minimum of D420:

L    W  K120D,B   ;DM no. 120
L    W  K6D,A     ;Offset module/module in bytes
MUL  W  A,B       ;DM no.*  module offset
L    W  K0654H,A  ;Address offset module length, DM0
ADD  W  B,A       ;Address offset module length, DM120
L    W  K2H,B     ;Memory segment number
LIMR W  A,C       ;Read DB120 module size
CPLA W  K420,C    ;Available through D420?
JPM     -DM_nok   ;Jump if < ERROR
```

6.7 OM5 & OM7 Startup Modules

Two startup modules, OM5 and OM7, are available. If a startup module is linked with the PLC program, it will be automatically processed during the startup routine of the controller.

The start-up is governed by the following criteria:

- OM5: Startup module following restart, always processed subsequent to Power-On. This applies also if, upon Power-On, the ZE200 is in STOP mode. In this case, the OM5 is processed upon changing the operating mode via a Stop/Run command. OM5 is also processed after program loading.
- OM7: Startup module following restart. Unless the current restart comprises the first startup subsequent to Power-On, OM7 is always processed after a change of operating mode by means of Stop/Run.

6.7.1 Programming within Startup Modules

Within the startup modules, the entire instruction set and, as a consequence, also the I/O operation, can be activated through direct access.

As a close instruction for the startup modules, both the EM and/or the EP instruction can be used. Both have the same effect on the module.

In the event that, during the processing of startup OMs, program modules are called, the close instructions of such program modules will have the established meaning:

EM: Return to startup module that included the call.

EP: Cancel, continue with OM1.

6.7.2 Retriggering of Watchdog and Cycle Time

- In the OM2, the hardware watchdog function can be disabled for the duration of the startup modules. As a consequence, very long startup routines (initialization of peripheral modules) will not cause the controller to stop.
- The software watchdog time is set in the OM2, and cannot be changed once the program is operating.
- The cycle time is always measured from OM1 to OM2, and therefore also contains the time of the I/O state.

⇒ **In the case of peripheral operations with the hardware watchdog disabled, faulty programming (endless loops) may create dangerous system conditions!**

6.8 OM9 Error Module

OM9 is the error module. If this module is linked with the PLC program, any error occurrences which would normally cause an immediate Stop of the central processing unit, the OM9 will be called automatically.

The same happens in all cases of errors which are also designated by setting a bit in the system range.

Exception: If no cycle time limit was designated, and the hardware-dependent cycle time limit is reached due to a programming error, the CL200 will automatically enter Stop mode. In this case, enabling an error OM will no longer be possible.

The error module can be programmed with remedial measures to be launched in the event that an error occurs. For example, designated data, including the error bits in the system area, can be copied to nonvolatile memory areas.

A retriggering of programme execution with error acknowledgement is excluded. This causes the CL200 to enter Stop mode after each time the OM9 has been processed, regardless whether the EM or EP instruction was used as the close instruction for the module.

6.9 Remanence Characteristics

Unless other limits are specified within the OM2, the remanence characteristics of the CL200 are subject to the range limits described below. These limits cannot be changed by means of the PLC program.

6.10 Remanent Operation

In remanent operation, the statuses of the designated remanent operands are retained after a Stop/Run and Power-On/Off mode change. As a precondition, no battery failure may exist.

In the absence of specific designations in the OM2, this means that the following areas are remanent:

- The upper half of the marker range, M96 through M191
- The upper half of the counters, C32 through C63
- The upper half of the timers, T64 through T127
- The entire data field, the data modules and the fixation are always remanent. They will be deleted only in the case of a battery failure or, in the case of fixations, upon request by the programming unit (PG).

6.11 Non-remanent operation

The non-remanent operation is set by shifting the remanence limits in the OM2 to the highest possible address.

The entire data field, the data modules and the fixation are always remanent. They will be deleted only in the case of a battery failure or, in the case of fixations, upon request by the programming unit (PG).

6.12 Fixation

The ZE200 central processing unit offers the option of fixing the operands by means of the programming unit.

In contrast to the "Control" command of the PG, operands can be permanently set to specific bit statuses and/or values.

The following data areas in the CL200 system are fixable:

Fixable data range	Comment
Inputs	
Outputs	

6.12.1 Remanence of Fixation

An established fixation remains enabled under the following conditions:

- **After a Stop/Run change of operating mode**
- **After reloading, provided this is defined in the OM2**
- **After Power-Off/On**

6.13 Interrupts

The ZE200xx central processing unit utilizes several groups of interrupts:

TI	Program interruption by means of time-controlled OM
PI	Program interruption by means of a peripheral event (interrupt inputs)

When an interrupt occurs, normal program execution is interrupted, and the associated interrupt module is activated.

The lowest priority is assigned to the group of timed interrupts, and the highest to the group of peripheral interrupts. Within individual groups, the interrupt assigned to the lowest OM number has the highest priority.

6.13.1 Time Interrupts (time-controlled processing)

For each time the time OM is called, the following must be true:

1. The designated time interval has expired, **and**
2. sequential processing has reached a change of module.

   Neither a DM call-up nor an EP instruction is considered a change of module!

   The time interrupts are always enabled. Interrupt disabling / enabling functions are controlled by interrupt mask programming.

⇒ **Due to programmed module nesting within time OMs, additional time OMs can occur and be processed, with the understanding that active time modules are incapable of causing their own interruption.**

6.13.2 Peripheral Interrupts (interrupt inputs)

The standard hardware configuration of all CL200 central processing units features three interrupt inputs which are wired to the X71 interface connector, and which are used to trigger peripheral interrupts.

A peripheral interrupt is triggered by a 0→1 (LOW → HIGH) signal change on the associated input of the ZE200 central processing unit, and is not linked to a module change. Instead, it branches into the respective interrupt OM immediately after processing a suitable instruction in the PLC program.

In this process, the flag register, i.e., RES, etc., and the system register contents are rescued.

The user is responsible for effecting a possible rescue of scratch markers, etc.

The peripheral interrupts are always enabled. Interrupt disabling / enabling functions are controlled by interrupt mask programming.

⇒ **Active peripheral interrupts are neither capable of interrupting themselves, nor can they be interrupted by time interrupts.**

6.13.3 Interrupt Handling Instructions

The central processing control unit (ZE) internally assigns one interrupt mask each to all interrupt groups, time interrupts (TI) and peripheral interrupts (PI). The TIM and LIM instructions are used to read from and write to these masks.

The mask associated with a given interrupt group contains one bit for each interrupt.

Bit set HIGH: the respective interrupt is enabled.

Bit set LOW: the respective interrupt is disabled.

To effectively enable the interrupts assigned in the mask, the additional EAI (Enable All Interrupts) instruction is required!

To generally disable an interrupt group without influencing the mask entries, the DAI (Disable All Interrupts) instruction is required.

Incoming interrupt signals cause an entry in the corresponding interrupt register, even though the corresponding interrupts are masked. Again, each interrupt is assigned one bit.

If the interrupt is executable, i.e., enabled, the bit in the interrupt register will be automatically cancelled by the call-up of the interrupt OM.

If the interrupt is disabled, the bit will remain in the interrupt register while the interrupt is waiting to be enabled.

The interrupt register can be read with the LAI (Load All Interrupts) instruction, and waiting interrupts can be cancelled with the RAI (Reset All Interrupts) instruction.

During a change of operating mode with the use of Stop/Run and Power-Off/On, all waiting interrupts are cancelled.

The PI masks are set to zero, and the peripheral interrupts disabled. Any required interrupts must be enabled be the user with the TIM and EAI instructions!

Time interrupts are enabled by default.

During startup, i.e., while processing OM5 and OM7, all interrupts remain disabled.

6.14 Application Stack

The application stack (AST) comprises a pushdown-pop-up memory stack with a storage depth of 128 words, using FILO (first-in-last-out) processing.

The PUSH and POP instructions facilitate a word-by-word data transfer between the registers and the contents of the application stack.

Example:

```
PUSH W  A   ; Shift contents of register A to applic. stack
PUSH W  B   ; Shift contents of register B to applic. stack
PUSH W  C   ; Shift contents of register C to applic. stack
PUSH W  D   ; Shift contents of register D to applic. stack

POP  W  D   ; Load contents of applic. stack into Register D
POP  W  C   ; Load contents of applic. stack into Register C
POP  W  B   ; Load contents of applic. stack into Register B
POP  W  A   ; Load contents of applic. stack into Register A
```

In the event of an application stack underflow, bit S28.4 in the system range will be set to HIGH. In the case of an application stack overflow, bit S28.5 in the system range will be set to HIGH.

Both application stack (AST) underflow and overflow conditions will cause the central processing module to enter Stop mode.

The application stack is flushed after each EP!

6.15 Setting the System Clock

⇒ **When setting the system clock, no crosschecks with the calendar month are made for the "Day" entry. As a result, incorrect entries may occur (e.g., April 31). Neither is the "Day of the Week" referenced to the date. Therefore, when setting the system clock, the user is required to ensure that all data information has been entered correctly.**

Procedure:
The system clock is set by writing into the system area defined by S128 through S134, whereby the write-access must occur transition-controlled (pulse). Otherwise, the system time will be reset in each PLC program cycle. Setting the system clock may be effected not only by the PLC program but also via the communication protocols.

⇒ **In the event that, when setting the system clock, the respective permitted value range is exceeded, the existing clock settings will remain unchanged.**

Value ranges:

Minutes	0-59	Seconds	0-59
Day	1-31	Hours	0-23
Year	0-99	Month	1-12
		Day of Week	0-6 0=Sun ... 6=Sat

7 CL200 Addressing Conventions

7.1 Operand & Module Identifiers

Abbr.	Operand	Peripheral access / data width	Image update
I II EI	Input with image Interface inputs I physically equal to II Extended input	Image/ bit, byte, word direct/ byte, word direct/ byte, word	in I/O state during PGM execution ./.
O IO EO	Output Interface outputs O physically equal to IO Extended output	Image/ bit, byte, word direct/ byte, word direct/ byte, word	in I/O state during PGM execution ./.
M	Marker		
T	Timer		
C	Counter		
D DX	Data word, 1st curr. DM Data word, 2nd curr. DM		
DF	Data field		
OC	Onboard counter		
S	System area		
K	Constant		
DM	Data module	CM DMnn ; calls 1st active DM CX DMnn ; calls 2nd active DM	
FC	Program module		

7.2 Module List

The CL200 manages the following program modules:

- Organization modules
- Program modules
- Data modules

Name	Function	Remark
OM1	Cyclical program execution	
OM2	Initialization table	refer to Section, "OM2 Initialization module"
OM5	Startup module after Power-ON	
OM7	Startup module after Stop/Run	
	:	
OM9	Error module	e.g., cycle time error
OM10	Interrupt module	assigned interrupt = I 0, priority 1
OM11	Interrupt module	assigned interrupt = I 1, priority 2
OM12	Interrupt module	assigned interrupt = I 2, priority 3
	:	
OM18	Time-controlled module	Raster definition in OM2 or S10, priority 1
OM19	Time-controlled module	Raster definition in OM2 or S12, priority 2
FC0-FC255	Program modules	
DM0-DM255	Data modules	

7.3 System Area

The ZE200 central processing unit features a system area encompassing 256 data words (S0 through S255).

This is the location of the configuration files of the CL200 system.

Essential specifications defined in the OM2 are copied into the system area, where they can be read by the PLC program.

To the extent deemed useful, system conventions related to cycle time can be also changed. This includes also the time intervals of the time-controlled organization modules and the system clock.

In addition to the data relating to the ZE, the system area also contains configuration data of all intelligent modules encompassed by the CL200 system.

Certain segments of the system range are utilized by standard function modules providing data that is also useful to other PLC program segments.

The unassigned addresses in the system area are reserved for internal purposes, and may not be modified.

7.3.1 System Area Assignment

Address	Contents	Comments
S0	Initialization flags, e.g., OM2_DW2	Read-only
S2	Reserved	
S4	Reserved	
S6	Reserved	
S8	Reserved	
S10	Time value for time-controlled processing, OM18	Read and write-access, Accordingly, time values can also be changed via PLC program
S12	Time value for time-controlled processing, OM19	
S14	Reserved	
S16	Reserved	
S18	Reserved	
S20	Counter, actual cycle time, factor = 1ms	Interval length, OM1-OM1, Reset on Stop/Run, Time refresh on error-based jump or in I/O state.
S22	Max. cycle time, factor = 1ms	Interval length, OM1-OM1, Reset on STOP/RUN
S24	Max. cycle time, factor = 1ms	

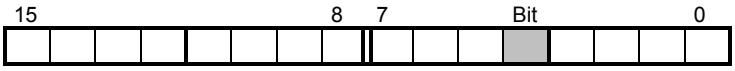
Address	Contents	Comment
S26	Error word 1 Bit: S26.0 Addressing error S26.1 Parameter error S26.2 S26.3 Module stack overflow S26.4 S26.5 S26.6 DM too small S26.7 Error "Jump direct“ (JP [R]) S27.0 Illegal write-access S27.1 Opcode error S27.2 Battery failure/ battery missing S27.3 Timer no. too large S27.4 S27.5 DM not active S27.6 Transfer error, peripheral bus S27.7	
S28	Error word 2 Bit: S28.0 S28.1 S28.2 Nonexistent module called S28.3 S28.4 Underflow, application stack S28.5 Overflow, application stack S28.6 S28.7 Cycle time error S29.0 S29.1 S29.2 S29.3 S29.4 S29.5 S29.6 S29.7	
S30	Bit field Bit: S30.0 Log. 0 S30.1 Log. 1 S30.2 Flashing marker (2 Hz) S30.3 Stop/Run trigger pulse S30.4 Power-On/Load Pgm trigger pulse S30.5 I/O fixed S30.6 Outputs disabled S30.7 Low Battery warning S31.0 Fault, diagnosable periph. module S31.1 Cable break, analog inputs S31.2 S31.3 S31.4 S31.5 S31.6 S31.7	Summary message (modules under development) Summary message
S32	Reserved	

Address	Contents	Comment
S34 S36 S38 S40 S42 S44	**OC0 Onboard counter** Actual value LOW word HIGH word Spec'd value 1 LOW word HIGH word Spec'd value 2 LOW word HIGH word	Upon reaching spec'd values, outputs defined in OM2/DEFW27 will be set.
S46 S48 S50 S52 S54 S56	**OC1 Onboard counter** Actual value LOW word HIGH word Spec'd value 1 LOW word HIGH word Spec'd value 2 LOW word HIGH word	Upon reaching spec'd values, outputs defined in OM2/DEFW27 will be set.
S58	**OC0 Onboard counter, control bits** S58.0 OC0 counting direction 0 = upward 1 = downward S58.1 Set OC0 actual value S58.2 Set OC0 specified value S58.3 S58.4 S58.5 S58.6 S58.7 **OC1 Onboard counter, control bits** S59.0 OC1 counting direction 0 = upward 1 = downward S59.1 Set OC1 actual value S59.2 Set OC1 specified value S59.3 S59.4 S59.5 S59.6 S59.7	After the transfer, the CL200 will reset the bits. After the transfer, the CL200 will reset the bits.
S60-S62	Reserved	
S64 S66 S68 S70	**Analog inputs** Analog input, channel 0 Analog input, channel 1 Analog input, channel 2 Analog input, channel 3	
S72	Reserved	
S74	Reserved	
S76	Reserved	
S78	Reserved	
S80	**Cable break reporting bits** S80.0 Analog input, channel 0 S80.1 Analog input, channel 1 S80.2 Analog input, channel 2 S80.3 Analog input, channel 3	
S82	**Analog output**	
S84-S127	Reserved	

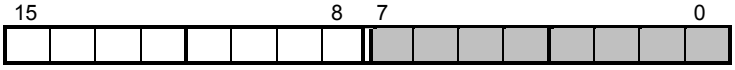
S128 S130 S132 S134	**System clock** (High byte / Low byte) Minutes / Seconds Day / Hours Year / Month	*Value range* 0-59 / 0-59 1-31 / 0-23 0-99 / 0-12
	Reserved / Day of Week	/ 0-6 (0=Sun)
S136-S142	Reserved	
	Initialization values for ZE200	
S145/144	---------- / Typ ID	00=ZE200; 01=ZE200A 02=ZE200M, 03=ZE200AM
S147/146	Boot firmware version / Hardware ver.	
S149/148	---------- / System firmware version	Firmware loadable with PG
S150	Reserved	
S152	Reserved	
S154	Reserved	
S156	Reserved	
S158	Reserved	
	Initialization values for intelligent modules	
S160-S175	Module 1	
S176-S191	Module 2	
S192-S207	Module 3	
S208-S255	Reserved	

7.4 Data Formats

Bit = X



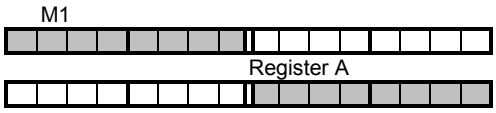
Byte = B



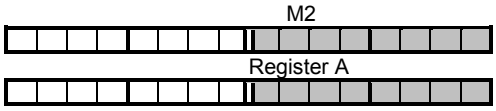
This addressing mode differentiates between load and transfer instructions:

Load instruction: The source operand may be either the even-numbered (LOW) byte or the odd-numbered (HIGH) byte.
In the case of the destination operand (register), the LOW byte is always addressed.

Examples: L BY M1,A



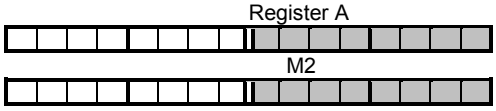
L BY M2,A



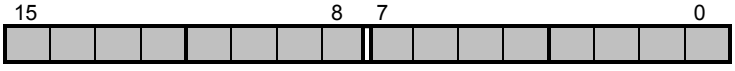
Transfer instr.: The low byte in the source operand (SRC_OPD - register) is addressed. The specified DEST_OPD may be both the even-numbered (LOW) byte and the odd-numbered (HIGH) byte.

Examples: T BY A,M1

T BY A,M2



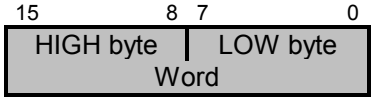
Word = W



7.5 Register Structure

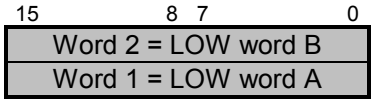
The CL200 features 4 working registers, which can be addressed in a bit-wise, byte-wise or word-by-word fashion. In this context, it should be noted that byte/word addressing always addresses the LOW-byte word.

Working registers A, B, C and D

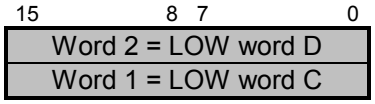


For operations that exceed the 16-bit format, the registers are combined to form permanent register pairs.

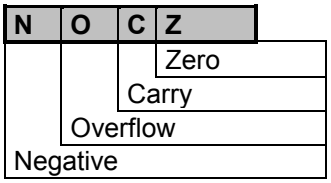
Working register pair A + B



Working register pair C + D



Status bits



⇒ **The negative flag always corresponds to the MSB (most significant bit) of the specified data format. Therefore, for byte operations, this is Bit 7, and for word operations, it is Bit 15.**

7.6 Representing Constants

The representation of constants is contingent upon the programming unit software being utilized. It has no functional bearing on the CL200.

Data Type		PLC utility programs	
Explanation	Notation	PROFI	WinSPS
UINT (unsigned integer)	Binary / Dual	K00000000 00000000B K11111111 11111111B	2#0000000000000000 2#1111111111111111
	Decimal, word	K00000D - K63535D	00000 - 65535
	Decimal, byte/byte	K000/000 - K255/255	Not defined in IEC1131 Part 3
	Hexadecimal	K0000H - KFFFFH	16#0000 - 16#FFFF
INT (signed integer)	Decimal, word	K-32768 - K+32767 K-32768D - K+32767D	-32768 - +32767
Text, STRING(2)	ASCII	K'AB'	'AB'
Time value, TVALUE	Time val. (+timebase r) r: 0=10ms, 1=100ms 2=1s, 3=10s	K0.r - K1023.r	T#10ms - T#10230s T#0.r - T#1023.r

7.7 Program Module Calls

	PLC utility programs			
	PROFI		WinSPS	
Program module / function call (IEC1131-3)	CM	PM	CM	FC

7.8 Jump Instructions

	PLC utility programs			
	PROFI		WinSPS	
Jump instruction	JPx	-label	JPx	label
Jump destination	-label		label:	

7.9 Bit & Module Addresses

Operand	Addresses (decimal)
I	0.0-23.7
O	0.0-15.7
M	0.0-191.7
T status	0-127
C status	0-63
DM	0-255
PM and/or FC	0-255

7.10 Byte Addresses

Operand	Address (decimal)	Comment
I, II EI	0-23 0-127	I is physically identical to II! external
O, IO EO	0-15 0-127	O is physically identical to IO! external
T actual value T status	0-127 0-127	Time range, 10 ms-1023 s (resolution 0.01; 0.1; 1; 10 s)
C actual value C status	0-63 0-63	Counter range, 0-8191
M	0-191	
S	0-255	The system area manages the following: - System initialization values - Analog onboard I/O - Onboard high-speed counters - System clock - Aux. bits (log1 / 0, RI, flasher, etc.)
P	0-31	
DF	0-8191	
D	0-511	
DX	0-511	

The even-numbered byte addresses are used as word addresses.

7.11 Addressing Modes

7.11.1 Direct Addressing

Operands for absolute addressing

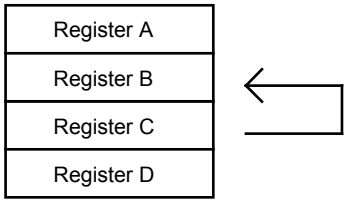
Byte/word readable	I, O, M, T and C K, DF, D, DX, S, II, EI	Actual values apply to T/C
Byte/word writable	O, M DF, D, DX, S, IO, EO	

Direct addressing of all absolute-addressable operands

Examples:

```
L     B     I10,B    ; Loads the status of input byte I10 into B.
L     W     100,C    ; Loads the value 100 into register C.
```

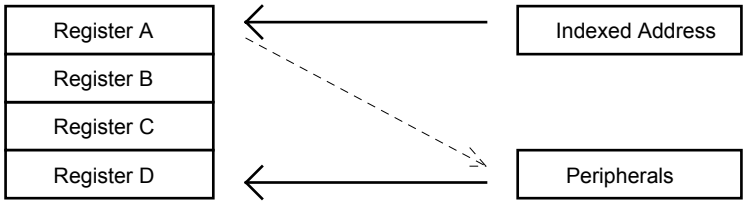
7.11.2 Register-to-Register Addressing



Example:

```
L     W     C,B      ; Loads the contents of register C into register B.
```

7.11.3 Register-indirect addressing



Example:

```
L     W     10,A     ; Loads index address into A as a byte number.
L     W     I[A],D   ; Loads the status of I10 (addr. in A) into register D.
```

7.11.4 Indirect addressing

The indirect addressing method – whether "word/byte or bit-oriented" – uses an operand prefix containing the operand identifier and the operand address. This greatly facilitates the handling and monitoring of operand addresses.

In addition, all data and program modules can be called up indirectly.

The operand prefix is structured as follows:

OPD[R] OPD = Operand identifier
[R] = Operand address in register A, B, C or D

The following is a demonstration of the indirect addressing principle, using the example of a block transfer:

Objective:
Five input words on address I10 are to be transferred to marker words starting with address M50.

```
L       W       5,A             ; Loading loop counter
L       W       10,B            ; Loading I10 byte base address
L       W       50,C            ; Loading M50 byte base address
continue:                       ; Loop entry label
L       W       I[B],D          ; Reading of contents (operand status)
T       W       D,M[C]          ; Writing of loaded status
INC     W       B,2             ; Next I-word (byte address + 2)
INC     W       C,2             ; Next M-word
DEC     W       A,1             ; Loop counter -1
JPN     continue                ; not all words processed so far
```

7.11.4.1 Indirect Byte Addresses

OPD ID	Byte address (dec.)	Instructions ... [Reg]	Examples OPD: see column 1
I, II	0-23	L	
EI	0-127		
O, IO	0-15	L, T	L 10,A
EO	0-127	IO and EO, T only	L OPD[A],B
T actual val.	0-127	L	
C actual val.	0-63	L	L 10,A
M	0-191	L, T	T B,OPD[A]
S	0-255	L, T	
DF	0-8191	L, T	
D	0-511	L, T	
DX	0-511	L, T	

To address the next byte and/or the next T/C when starting from an address, the address must be increased by 1. To address the next word, the address must be increased by 2.

In the event that an attempt is made to access a word by using an odd-numbered address (operand attribute = W), the controller will enter the Stop mode.

The cause of the fault can then be displayed by means of the PG.

⇒ **The CL200 does not perform range monitoring. Accordingly, the programmer is responsible for staying within range limits. In the case of write-access range violations, data will be destroyed, and the controller will enter the STOP mode.**

7.11.4.2 Indirect Bit Addresses

OPD ID	**Bit address** (dec.)	**Instructions**	**Examples** OPD: see co. 1.
I	0-191	A, AN, O, ON	
O	0-127	A, AN, O, ON S, R, =	L 10,A A B OPD[A]
M	0-1535	A, AN, O, ON S, R, =	= B OPD[A]
S	0-2047	A, AN, O, ON	
DF	0-65535	A, AN, O, ON S, R, =	
T status	0-127	A, AN, O, ON	
C status	0-63	A, AN, O, ON	

To address the next bit when starting from an address, the address must be increased by 1.

If a range limit violation is detected, the controller will enter the Stop mode. The cause of the fault can then be displayed by means of the PG.

7.11.4.3 Indirect Module Addresses

Operand	**Module Number**	**Instructions** ... [Reg]	**Example**
DM	0-255	CMx CXx	L W 10,A CM DM[A]
FC	0-255	CMx CMx	L W 100,A CM FC[A]

To address the next module starting from a module number, the module number must be increased by 1.

If a range limit violation is detected, or if the module is not available, the controller will enter the Stop mode. In both cases, the fault can subsequently be displayed by means of the PG.

7.12 Parameterized Modules

When a program module is called up, up to 32 parameter values can be transferred. The number of parameter values to be transferred is stated as part of the module call-up instruction, followed by the actual parameters, starting with the number P0.

All parameters that are to be used as a byte or word in the program module being called up are transferred without operand attribute.

(Depending on the version of the programming unit (PG) being used, the operand attribute **B** or **W** may be included, depicting that no operand attribute is being used.)

All parameters to be used as bits in the module being called up are transferred with the operand attribute **B**!

Exception:
If times and counters are transferred in the form of parameters without operand attribute, they may be utilized as both a word function, i.e., time/counter value, and/or a bit function, i.e., time/counter status in the module being called up.

Example of parameter transfer:

```
CM      FC100,7    ;Call up FC100 and transfer 7 parameters
P0      43         ;Parameter P0: FC no. as decimal constant K43
P1      4          ;Parameter P1: DM no. as decimal constant K4
P2      O56        ;Parameter P2: Output word with byte address O56
P3      I7.3       ;Parameter P3: Input bit I7.3
P4      T2         ;Parameter P4: Time T2
P5      C13        ;Parameter P5: Counter C13
P6      O10.0      ;Parameter P6: Output bit O10.0
```

Utilization of parameters in called-up module FC100:

```
L       P1,A       ;Load DM no. 4
CM      DM[A]      ;Open DM4
CX      DM5

L       P0,A       ;Load FC no. 43
CM      FC[A],2    ;Call up FC43 and transfer two parameter values
P0      D2         ;Parameter P0: D2 of active 1st DM, this being DM4
P1      DX6        ;Parameter P1: DX6 of active 2nd DM, this being DM5

L    W  P2,A       ;Load output word O56

L    W  P4,B       ;Load time value from T2 to B

A       P3         ;I7.3
A       P4         ;Status of T2
A       P5         ;Status of C13
=       P6         ;O10.0
```

8 Interfaces & Connectors

8.1 X31 – Connector for Programming Unit or External Operator Terminal

The X31 connector provides a combined V.24 / 20mA interface for connecting the programming unit (PG) or other peripheral devices, such as the BT20 operator terminal. This interface does not feature control lines, and only the 20mA section is electrically isolated.

The interface uses a female DB-25 connector.

Explanation	Designation	Pin no.	Signal Direction
V.24			
Transmit data	TxD	2	→
Receive data	RxD	3	←
Ref. & signal GND	Sig. GND	7	
20mA			
12 V out (active)	12 V out	10	For active mode, 9 ↔10 are bridged.
12 V in (active)	12 V in	9	
12 V Ref./GND	12 V GND	21	
Receive data + (passive/active)	RxD+	22/12	←
Receive data - (passive/active)	RxD-	12/24	
Transmit data + (passive/active)	TxD+	23/13	→
Transmit data - (passive/active)	TxD-	13/25	
Shield	Shield	Housing	

Cable lengths:

Baud rate	V.24	20mA
9600	15m	300m
19200	15m	150m
38400	15m	50m
57600	15m	-

The interface uses the BUEP19E (PST) **transmission protocol**.

The baud rates are set with a four-segment DIP switch on the ZE200xx as follows:

Switch segment 4	3	2	1	Baud	Fixed setting
OFF	OFF	OFF	OFF	9600	Parity Even
OFF	OFF	OFF	ON	19200	1 Stop bit
OFF	OFF	ON	OFF	38400	8 Data bits
OFF	OFF	ON	ON	57600	
reserved					

8.2 X32 – Second Serial Interface Connector (ZE201 only)

The second serial interface on the ZE201 central processing unit is used, for example, to connect the PG programming unit in the event that a BT20 operator terminal is connected to the standard X31 connector.

The interface features a male DB-9 connector which uses the following **pin assignment**:

Pin	Signal	Designation
1	-	
2	RxD	Receive Data
3	TxD	Transmit Data
4	-	
5	GND	Ground
6	-	
7	-	
8	-	
9	-	
Housing	Shield	

The maximum **cable length** for all baud rates is 15 metres.

The X32 serial interface supports the following **transmission protocols**:

BUEP19E (PST, similar to PG connector X31), and

BUEP03E.

In the same fashion as with the computer modules, the BUEP03E protocol is handled via the R2REQ and R2CONV function modules. To transmit task requests via the second serial interface of the ZE201, only the fictional CXN (computer interface) number "**FF**" must be addressed.

The BUEP19E transmission protocol of the 2nd interface (X32) connector is fully compatible with the X31 interface for the PG programming unit. The RST function is not available.

Basic transmission protocol and baud rate settings

Baud rate:	19200 Baud
Number of stop bits:	1 Stop Bit
Parity:	EVEN
Number of information bits:	8
Protocol:	BUEP19E

In the event that the **protocol and transmission speed settings** must depart from the default setting, the desired declaration is made in the OM2, DW36.

8.3 X71 – Interrupt & Counter Inputs

The interrupt and counter inputs are part of the standard equipment of every ZE200xx, and are connected by means of a female DB-9 connector.

In both cases the inputs operate with 24 V signals.

Explanation	Designation	Pin no.	Signal direction
Interrupt I0	II0	1	←
GND		2	
Interrupt I1	II1	3	←
GND		4	
Interrupt I2	II2	5	←
Counter input, OC0	OCI0	6	←
Counting direct., OC0	OCD0	7	←
Counter input, OC1	OCI1	8	←
Counting direct., OC1	OCD1	9	←
Shield		Housing	

8.3.1 Interrupt Inputs (ZE200xx standard version)

As described in the foregoing, the interrupt inputs are wired to the X71 interface connector and, in the case of a transition from 0→1, trigger the assigned peripheral interrupt. The response to the interrupt is programmed in the corresponding interrupt OM.

Interrupt I 0	OM10	Priority 1
Interrupt I 1	OM11	Priority 2
Interrupt I 3	OM12	Priority 3

Minimal programming required for interrupt detection and processing:

```
1.  L     W   7H,A     ; Prepare all three interrupts for enable.
    TIM   W   A,PI     ; Write peripheral interrupt mask.

2.  EAI       PI       ; Enable interrupts.
```

In the event that a signal transition occurs on one of the interrupt inputs, the associated OM will be called. If this module has not been integrated into the program, the controller will enter STOP mode while returning an appropriate error message. If several interrupts occur at the same time, they will be processed in accordance with the above listed priority ranking.

8.3.2 High-speed Counters (onboard counters, standard on ZE200xx)

For operations requiring high-speed counting, the CL200 provides two independent 32-bit counters, which can be operated in both upward and downward-counting modes. The reversal of the counting direction may be effected either by the PLC program or by external signals via special directional inputs.

Effective with v1.6, the Incremental Rotary Transducer" counter mode is available.

This function is subject to the following preconditions:

- **There is a new counting mode, termed "Incremental Rotary Transducer". This mode is available for counter 1 but not for counter 0.**
- **This mode is enabled by setting the MSB in word 20 of OM2. This renders all other bits in this word meaningless.**
- **The counter performance is max. 10 kHz and/or a line count of 10,000 per second.**
- **At the same time, counter 0 may be used without directional change.**
- **In the event that the maximum counting rate of 10 kHz is used in the "Incremental Rotary Transducer" mode, an approximate 35% slow-down in the PLC cycle rate may be expected.**
- **The "Transducer" mode does not permit spec'd-value monitoring!**
- **The "Transducer" mode provides the PLC with a dual interpretation, i.e., both the rising and falling edge of pulses are counted. Accordingly, a rotary transducer with 1000 lines per revolution produces a counter value of 2000 with each revolution.**
- **The permissible limit frequency of 10 kHz is not monitored. If the frequency is exceeded, counting faults will result.**
- **In the event that the zero-pulse is to be used as an interrupt input signal, it must have a minimum duration of 80 µs.**

When using these counters, the integration of the OM2 is a mandatory requirement. All parameter values required for the counter are predefined in data words DW13-DW27 of this module.

For utilization within the PLC program, the counter values and the required control bits can be accessed in S34-S58 of the system area.

⇒ **Effective with v1.2 of the system firmware, the "Set Specified Values" command (see system range, S148) is available.**

OM2 Preset values:

Onboard counter settings (OC0: DW 13 / OC1: DW 20)

```
13/20   DEFW W  2#0000000000000000
                 *******|*****|||       *: reserved
                        |     |++------ Definition of transitions
                        |     |           00 no       transition
                        |     |           01 positive transitions
                        |     |           10 negative transitions
                        |     |           11 both     transitions
                        |     +-------- Allow external up/down switchover
                        |
                        +-------------- Downward-counting
      not with DW 13    +--------------- increment. Rotary Trans. (V1.6)
```

Presetting actual values, LOW / HIGH word

OC0: DW 14/15 / OC1: DW 21/22

Values can be changed via the PLC program.

```
14/21   DEFW W  16#0000
15/22   DEFW W  16#0000
```

Presetting specified value 1, LOW / HIGH word (with DW17/24 not for Transducer mode).

OC0: DW 16/17 / OC1: DW 23/24

Values can be changed via the PLC program.

```
16/23   DEFW W  16#FFFF
17/24   DEFW W  16#FFFF
```

Presetting specified value 1, LOW / HIGH word (with DW19/26 not for Transducer mode).

OC0: DW 18/19 / OC1: DW 25/26

Values can be changed via the PLC program.

```
18/25   DEFW W  16#FFFF
19/26   DEFW W  16#FFFF
```

Output byte definition (no extended output permitted), not for Transducer mode.

Upon achieving the specified values, the outputs are set automatically. After the response has been effected, outputs must be again reset by the PLC program.

⇒ **Since the output byte defined here is output via direct access, bits 4-7 are also always output in their current image status (during program processing). Accordingly, when using high-speed counters, these outputs should always remain unused in order to prevent the return of unwanted statuses!**

Example: Entry 10 in word 27

```
   OC0 Spec'd value1 = Bit O10.0     OC0 Spec'd value2 = Bit O10.1
   OC1 Spec'd value1 = Bit O10.2     OC1 Spec'd value2 = Bit O10.3
```

```
27   DEFW W  0
```

System area

Address	Contents		Comment
	OC0/OC1 onboard counter		
S34/S46 S36/S48	Actual value	LOW word HIGH word	Actual values can be modified via the PLC program (see Control Bits)
S38/S50 S40/S52	Spec'd value 1	LOW word HIGH word	Specified values can be modified via the PLC program (see Control Bits)
S42/S54 S44/S56	Spec'd value 2	LOW word HIGH word	Upon reaching the spec'd values, defined outputs are set in DEFW 27 of the OM2.
S58/S59	**OC0/OC1 onboard counter Control bits** Bit0 Counting direction 0 = upward 1 = downward Bit1 Set actual value Bit2 Set specified value Bit3 Bit4 Bit5 Bit6 Bit7		After the transfer, the CL200 will reset the bits.

In the event that, during counting up (or counting down) the maximum value of FFFF FFFF$_H$ (or minimum value of 0$_H$) is attained, the counter will again start at 0 (or FFFF FFFF$_H$).

Notes:

The process of setting new current/specified values is transition-controlled, and occurs in the following sequence:

- **The "Set current/specified value" bit must be reset.**
- **The system word provides the new current/specified value, and the "Set current/specified value" control bit is set once (never cyclically).**
- **In the subsequent cycle, the value is transferred and the control bit reset. In the case of manipulations using the actual value, the system variable again serves the display of current actual values.**

Without exception, the updating of actual values in the system range occurs in the I/O state. Independent of program processing, the outputs assigned to the specified values are affected immediately.

⇒ **To prevent the loss of pulses, the default entry for change of direction and the counting pulses must not occur simultaneously.**

8.3.2.1 Sample Program – High-speed Counters onboard CL200 Basic Unit

```
; ************************************************************************
; *                          High-speed Counters                         *
; ************************************************************************

; Examples of onboard counters (high-speed counters) in the CL200
; basic unit.

; The basic unit provides 2 high-speed counters, termed
; "OC0 onboard counter" and "OC1 onboard counter" hereunder.

; The OM2 initialization module must be linked, and data words 13 to 32
; in this module must be suitably modified.


; The OM2 entries for the high-speed counters are described below.
;
;       Definition of onboard counters (OC) (entries permitted)
;       =======================================================
;       Entry 0 = Function NOT available, and/or DO NOT execute
;       Entry 1 = Function available, and/or execute
;
;DW 13: Settings, on-board counter 0 (OC0)
;-----------------------------------------
; 13   DEFW W  2#0000000000000110
;                *******|*****|||       *: reserved
;                       |     |++------ Definition of transitions
;                       |     |           00 no       transition
;                       |     |           01 positive transitions
;                       |     |           10 negative transitions
;                       |     |           11 both     transitions
;                       |     +-------- allow external up/down switchover
;                       |
;                      +-------------- Downward-counting

;DW 14/15: OC0 actual value, LOW/HIGH word
;-----------------------------------------
; 14   DEFW W  16#0000
; 15   DEFW W  16#0000

;DW 16/17: OC0 spec'd value1   LOW/HIGH word
;-----------------------------------------
; 16   DEFW W  16#FFFF
; 17   DEFW W  16#FFFF

;DW 18/19: OC0 spec'd value2   LOW/HIGH word
;-----------------------------------------
; 18   DEFW W  16#FFFF
; 19   DEFW W  16#FFFF

;DW 20: Settings for onboard counter 1 (OC1)
;-------------------------------------------
; 20   DEFW W  2#0000000000000000
;                *******|*****|||       *: reserved
;                |      |     |++------ Definition of transitions
;                |      |     |           00 no       transition
;                |      |     |           01 positive transitions
;                |      |     |           10 negative transitions
;                |      |     |           11 both    transitions
;                |      |     +-------- allow external up/down switchover
;                |      |
;                |      +-------------- Downward-counting
;                +--------------------- Incremental rotary transducers (v1.6)
```

```
;DW 21/22: OC1 actual value, LOW/HIGH word
;-----------------------------------------
; 21   DEFW W  16#0000
; 22   DEFW W  16#0000

;DW 23/24: OC1 spec'd value, LOW/HIGH word
;-----------------------------------------
;
; 23   DEFW W  16#FFFF
; 24   DEFW W  16#FFFF

;    ;DW 25/26: OC2 spec'd value, LOW/HIGH word
;-----------------------------------------------
; 25   DEFW W  16#FFFF
; 26   DEFW W  16#FFFF

;DW 27: Number of output byte
;-----------------------------
;    Outputs are set automatically upon reaching specified values.
;    This address can be used to set the output byte used for
;    this purpose.
;                e.g., K0010D
;                 OC0 spec'd value1 = Bit 10.0      OC0 spec'd value2 = Bit 10.1
;                 OC1 spec'd value1 = Bit 10.2      OC1 spec'd value2 = Bit 10.3
; 27   DEFW W  K0000D


;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

; Assignments in system area
; --------------------------
DEF     S34,-OC0_IwL      ;   OC0 onboard counter, actual value    LOW word
DEF     S36,-OC0_IwH      ;   OC0 onboard counter, actual value    HIGH word
DEF     S38,-OC0_Sw1L     ;   OC0 onboard counter, spec'd value1   LOW word
DEF     S40,-OC0_Sw1H     ;   OC0 onboard counter, spec'd value1   HIGH word
DEF     S42,-OC0_Sw2L     ;   OC0 onboard counter, spec'd value2   LOW word
DEF     S44,-OC0_Sw2H     ;   OC0 onboard counter, spec'd value2   HIGH word

DEF     S46,-OC1_IwL      ;   OC1 onboard counter, actual value    LOW word
DEF     S48,-OC1_IwH      ;   OC1 onboard counter, actual value    HIGH word
DEF     S50,-OC1_Sw1L     ;   OC1 onboard counter, spec'd value1   LOW word
DEF     S52,-OC1_Sw1H     ;   OC1 onboard counter, spec'd value1   HIGH word
DEF     S54,-OC1_Sw2L     ;   OC1 onboard counter, spec'd value2   LOW word
DEF     S56,-OC1_Sw2H     ;   OC1 onboard counter, spec'd value2   HIGH word

DEF     S58,-OC_StB       ;   Onboard counter, control bits
                          ;
                          ;       S58.0   OC0 counting direction
                          ;                 0 = upward
                          ;                 1 = downward
                          ;       S58.1   Set OC0 actual value
                          ;       S58.2   Set OC0 spec'd values
                          ;
                          ;       S59.0   OC1 counting direction
                          ;                 0 = upward
                          ;                 1 = downward
                          ;       S59.1   Set OC1 actual value
                          ;       S58.2   Set OC0 spec'd values
```

```
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

; Connectors, pin assignments
; ---------------------------
        ; OC0 counter input        X71 connector, pin 6
        ; OC0 counting direction   X71 connector, pin 7
        ; OC1 counter input        X71 connector, pin 8
        ; OC0 counting direction   X71 connector, pin 9


;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

; Output byte assignment (specified in OM2, w27)
; ----------------------------------------------
; Outputs are set automatically upon reaching specified values.
; After the appropriate response has been received, outputs must again
; be reset by the PLC program.
        ; Bit Axx.0       OC0 spec'd value1
        ; Bit Axx.1       OC0 spec'd value2
        ; Bit Axx.2       OC1 spec'd value1
        ; Bit Axx.3       OC1 spec'd value2
        ; Bit Axx.4 - 7   DO NOT USE



;============================================================================

; Example


; *** Set Specified Value ***

; Transition control for 'Set Specified Value' instruction - the loading
;       procedure for specified value will be enabled only
;       if a 0-to-1 transition is detected on input I0.0.

1    AN      I0.0              ; 'Set Specified Value' input bit
2    R       M190.0            ; reset help marker
3    A       I0.0
4    AN      M190.0
5    S       M190.0
6    JPCI    noload1

     ; Loading procedure for specified value
     ; Load 0C0 specified value (on-board counter0) and enable control bit for
     ; 'Set Specified Value' command. The control bit will be reset automatically
     ; by the PLC once the value has been transferred.
     ; Write spec'd value1  (S40, S38)
7    L    W  10,D
8    L    W  0,C
9    T    W  D,S38             ;  OC0 onboard counter, spec'd value1   LOW word
10   T    W  C,S40             ;  OC0 onboard counter, spec'd value1   HIGH word
     ; Write spec'd value2  (S44, S42)
11   L    W  100,D
12   L    W  0,C
13   T    W  D,S42             ;  OC0 onboard counter, spec'd value2   LOW word
14   T    W  C,S44             ;  OC0 onboard counter, spec'd value1   HIGH word
     ; Enable 'Set Specified Value' control bit
     ; Note: May be active during one cycle only (see Transition Control, above)
15   L    W  2#00000000000100,A
16   T    W  A,S58             ;  Onboard counter, control bits

     noload1:
```

```
     ; *** Set Actual Value ***

     ; Transition control for 'Set Actual Value' instruction - the loading
     ;       procedure for actual value will be enabled only
     ;       if a 0-to-1 transition is detected on input I0.1.

17   AN      I0.1              ; 'Set Actual Value' input bit
18   R       M190.1            ; Reset help marker
19   A       I0.1
20   AN      M190.1
21   S       M190.1
22   JPCI    noload2

     ; Loading procedure for actual value
     ; Load 0C0 actual value (onboard counter0) and enable control bit for
     ; 'Set actual value'. The control bit will be reset automatically
     ; by the PLC once the value has been transferred.
     ; Write actual value (S36, S34)
23   L    W  50,D
24   L    W  0,C
25   T    W  D,S34              ;  OC0 onboard counter, actual value   LOW word
26   T    W  C,S36              ;  OC0 onboard counter, actual value   HIGH word
     ; Enable 'Set Actual Value' control bit
     ; Note: May be active during one cycle only (see Transition Control, above)
27   L    W  2#00000000000010,A
28   T    W  A,S58              ;  Control bits, onboard counter

     noload2:


     ; Read and reset output bits which are set by direct access by
     ; the counter (specified in initialization module OM2, w27).

29   A       O0.0        ; OC0 Spec'd value1 was reached
                         ; This location for programming additional responses as required.
30   R       O0.0        ;

31   A       O0.1        ; OC0 Spec'd value2 was reached
                         ; This location for programming additional responses as required.
32   R       O0.1        ;



33   L    W  M180,A
34   INC  W  A,1
35   T    W  A,M180
36   A       A.4
37   =       O0.4
38   A       I0.3
39   =       O0.5
40   L    W  S34,A              ;  OC0 onboard counter, actual value    LOW word
41   L    W  S36,A              ;  OC0 onboard counter, actual value    HIGH word
42   L    W  S38,A              ;  OC0 onboard counter, spec'd value1   LOW word
43   L    W  S40,A              ;  OC0 onboard counter, spec'd value1   HIGH word
44   L    W  S42,A              ;  OC0 onboard counter, spec'd value2   LOW word
45   L    W  S44,A              ;  OC0 onboard counter, spec'd value2   High word

46   EP
```

8.4 X72 – Analog Inputs & Analog Output

For "simple" processing of analog values, the CL200 with its ZE200Ax central processing control units provides four analog inputs and one analog input. They are connected by means of a female DB-15 connector.

The analog values and the cable-break messages of the analog inputs are accessible in system range S64-S82.

Explanation	Designation	Pin no.	Signal direction
Analog input 0	AI0	1	←
Analog input 1	AI1	2	←
Analog input 2	AI2	3	←
Analog input 3	AI3	4	←
Analog output OP mode	OM	5	open: Output I AOI bridged: output U
Analog output, voltage	AOU	10	→
Analog output, current	AOI	14	→
reserved		6, 13	
GND		7, 8, 9, 11, 12, 15	
Shield		Housing	

When the controller is powered up without the OM2, the processing of all analog inputs within a voltage range of 0-10 V is enabled.

OM2 Settings (OM2 is not an indispensable requirement)

To enable the control processor to restrict processing to channels actually in use, assigned analog inputs are specified in DW 31.

```
;    Entries are possible from K00D through K04D
;      0 = no analog input enabled
;      2 = two analog inputs enabled (channels 0 and 1)
;
31   DEFW W  4
```

To select whether or not which analog input channels are to be enabled, and/or whether the analog output is to be standardized.

```
; The selected analog inputs are standardised to  2 V  - 10 V, and
; the enabled analog output is standardised to    2 V  - 10 V
; and/or to                                       4 mA - 20 mA
;
;
32   DEFW W  2#0000000000000000
;          *******|****|||       *: reserved
;                 |    |||+------ Analog input, channel 0
;                 |    ||+-------               channel 1
;                 |    |+--------               channel
;                 |    +---------               channel 3
;                 |
;                +--------------- Analog output, channel 0
;
; Entry 0 switches off standardization.
```

8.4.1 Analog Inputs

The voltage value present at the analog inputs is converted and subsequently written into the system area for further processing in digital form.

If a cable break is detected during normal (default) operation (value < 4 mA and/or < 2 V), this will be reported in word S80 of the system area.

Specifications

Number of analog inputs	4, pursuant to IEC 1131-2
Electrical isolation	no
Input range	0 – 10 V 2 – 10 V, when standardized via OM2
Permissible input voltage	-10 V through 30 V
Digital mapping	16-bit
Resolution	10-bit
Input resistance	20.4 kΩ
Conversion time	20 µs
Scanning interval	10 ms, value updating in system area during subsequent I/O state
Error tolerance	1% at 0 through 55 °C
Cable length	max. 100 metres, screened

System area

	Analog inputs	
S64	Analog input value, channel 0	
S66	Analog input value, channel 1	
S68	Analog input value, channel 2	
S70	Analog input value, channel 3	
S72	Reserved	
S74	Reserved	
S76	Reserved	
S78	Reserved	
S80	**Cable break reporting bits** S80.0 Analog input, channel 0 S80.0 Analog input, channel 1 S80.0 Analog input, channel 2 S80.0 Analog input, channel 3	

Priority representation

Priority Bit										**without significance**					
9	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**x**	**x**	**x**	**x**	**x**	**x**
MSB									**LSB**						

LSB voltage value:

- Default operation: 10 V / 1024 = 9.8 mV
- With standardization: 8 V / 1024 = 7.8 mV

8.4.2 Analog Output

The PLC program writes into the system area the digital representation of the voltage or current output value that will be sent to the analog output, and utilizes the I/O state to transfer it to the output.

Specifications

Number of analog outputs	1, pursuant to IEC 1131-2
Electrical isolation	no
Short-circuit protection	yes, unlimited
Short-circuit current	32 mA
Error tolerance	1% at 0 - 55 °C
Cable length	max. 100 metres, screened
Digital mapping	16 bit
Resolution	12 bit
Conversion time	1 ms
Value output	after writing to system area, value output occurs during subsequent I/O-State.

Output voltage

Output range	0 – 10 V 2 – 10 V, with standardization via OM2
Load resistance	≥ 1 kΩ

Output current

	0 – 20 mA 4 – 20 mA, with standardization via OM2
Load resistance	≤ 600 Ω

System area

S82	**Analog output value**	

Priority representation

Priority bit												**without significance**			
11	10	9	8	7	6	5	4	3	2	1	0	x	x	x	x
MSB											**LSB**				

LSB voltage value

- Default operation: 10 V / 4096 = 2.4 mV
- With standardization: 8 V / 4096 = 2.0 mV

LSB current value

- Default operation: 20 mA / 4096 = 4.9 µA
- With standardization: 16 mA / 4096 = 3.9 µA

8.4.2.1 Sample Program – Analog Processing on the CL200 Basic Unit

```
; Examples of analog inputs and the analog output
    ; on the CL200A and CL200AM central processing units.
    ;
    ; Measuring ranges:   analog inputs       0 - 10 V
    ;                                         2 - 10 V
    ;                 analog output           0 - 10 V
    ;                                         0 - 20 mA
    ;                                         2 - 10 V
    ;                                         4 - 20 mA
    ; Note: Whenever an example specifies standardized operation,
    ; the OM2 initialization module must be linked to the program,
    ; and data word 32 in the OM2 must be appropriately modified.


    ; ************************************************************************ag
    ; *             Analog inputs, CL200 Central Processing Unit              *
    ; ************************************************************************

    DEF     S64,-AnaKan0      ; = Analog value, channel 0, X72 connector, pin 0
    DEF     S66,-AnaKan1      ; = Analog value, channel 1, X72 connector, pin 1
    DEF     S68,-AnaKan2      ; = Analog value, channel 2, X72 connector, pin 2
    DEF     S70,-AnaKan3      ; = Analog value, channel 3, X72 connector, pin 3
                              ;   GND           X72 connector, pins 9,11,12,15
    DEF     S80,-KaBruch      ;   Cable break message, channel 0 = Bit 0
                              ;   Cable break message, channel 1 = Bit 1
                              ;   Cable break message, channel 2 = Bit 2
                              ;   Cable break message, channel 3 = Bit 3
                     ;  Cable break is reported only during standardized
                     ;  2-10 V operation (as specified in OM2, w 32).

    ;    Bit assignment:
    ;    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    ;    | 15| 14| 13| 12| 11| 10| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
    ;    +MSB+---+---+---+---+---+---+---+---+LSB|---+---+---+---+---+---+
    ;    |<============ Analog value ============>|******* not used ******|

    ;    Example:    1111111111xxxxxx  = Analog value, 10 V
    ;                 0000000000xxxxxx = Analog value,  0 V
```

```
      ; ======================================================================ag
      ; *     Start of Sample Programs "Reading and Scaling Analog Values"     *
      ; ======================================================================

      ;Example 1
      ;
      ;  Loading analog value from channel 0 (default operating mode / non-
      ;  standardized) on CL200 central processing unit, using 0-1023 scaling
      ;  over 0-10000 (mV). The value is filed in data module DM0, word 0.

1     CM      DM0
2     L    W  S64,A      ; = Analog value, channel 0, X72 connector, pin 0,
3     SLR  W  A,6        ; and load into bits 0-11.
                         ; Scale value 0-1023 over 0-10000 mV
4     L    W  9775,C     ; at weighting: 1 bit = 0.9775 mV,
5     MUL  W  C,A        ; and multiply with loaded value.
6     L    W  1000,C     ; --"--
7     DIV  W  C,A        ; --"--
8     T    W  A,D0       ; Output scaled analog value in data word 0 of DM.


      ;Example 2

      ;  Reading of analog value from channel 1 (standardized operating mode /
      ;  entry in OM2 DW32) on CL200 central processing unit, using 0-1023 scaling
      ;  over 2000-10000 (mV). The value is filed in data module DM0, word 2.

9     CM      DM0
10    L    W  S66,A      ; = Analog value, channel 1, X72 connector, pin 1,
11    SLR  W  A,6        ; and load into bits 0-11.
                         ; Scale value 0-1023 over 0-10000 mV
12    L    W  7820,C     ; at weighting 1 bit = 0.7820 mV,
13    MUL  W  C,A        ; and multiply with loaded value.
14    L    W  1000,C     ; --"--
15    DIV  W  C,A        ; --"--
16    L    W  2000,D     ; Scale value 0-8000 over 2000-10000 mV.
17    ADD  W  D,A        ; --"--
18    T    W  A,D2       ; Output scaled analog value in data word 2 of DM.

19    L    W  S80,B      ; Cable break message, channel 0 = Bit 0
20    A    B  B.1        ; --"--
21    =    B  O0.0       ; Response to cable break


      ; *     End of sample programs "Reading and Scaling Analog Values"     *
```

```
; ********************************************************************ag
; *          CL200 Central Processing Unit, Analog output              *
; ********************************************************************

DEF     S82,-AnaAus       ; =  Analog output
                          ;    Current:             X72 connector, pin 14
                          ;    - alternatively
                          ;    Voltage:             X72 connector, pin 10
                          ;                          bridge pins 5 and 14

                          ;    GND         X72 connector, pins 9,11,12,15


; Default setting  0 - 10 V  and/or  0 - 20 mA
; or
; Standardized operation, 2-10 V and/or 4-20 mA (as defined in w32 of OM2).
;


;    Bit assignment:
;    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
;    | 15| 14| 13| 12| 11| 10| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
;    +MSB+---+---+---+---+---+---+---+LSB|---+---+---+---+---+---+---+
;    |<================ Analog value ================>|*** not used **|

;    Example:  111111111111xxxx  = Analog value 10 V
;               000000000000xxxx = Analog value  0 V (standardized: 2 Volt)


; ====================================================================ag
; *     Start of Sample Program "Scaling and Outputting Analog Value"    *
; ====================================================================


;Example 1
;
;  Outputting an analog value after prior scaling.
;  The value of 0-10000 (mV) in data module DM0, on word 20 is to be
;  output as a voltage of 0-10 V.


22   CM      DM0        ; Open data module
23   L    W  D20,A      ; Load analog value
                        ; Scale value of 0-10000 mV over 0-4095
                        ; at weight: 1 bit = 0.2442 mV
24   L    W  1000,C     ; --"--
25   MUL  W  C,A        ; --=--
26   L    W  2442,C     ; --"--
27   DIV  W  C,A        ; --"--
28   SLL  W  A,4        ; Load value into bits 4-15
29   T    W  A,S82      ; =  Analog output

; *    End of sample program "Scaling and Outputting Analog Value"   *

30   EM
```

9 Instruction List

9.1 Structure of Controller Instructions

Controller Instruction			
Operation part	Operand attribute	Source operand	Destination operand
OPP	OPA	SRC ,	DEST

Examples:

```
A                   I0.0
A       W           -Name       ,   A
L       B           O0          ,   B
T       W           C           ,   M10
MUL     W           K1234D      ,   D
```

9.2 Status Bits (Flags)

The status bits are influenced by the following instruction groups:

- **Compare**
- **Convert**
- **Swap**
- **Increment**
- **Decrement**
- **Shift**
- **Rotate**
- **Add**
- **Subtract**
- **Multiply**
- **Divide**

Instructions belonging to these groups are equally applicable to program processing instructions (jumps, module instructions) and logical links (flag queries).

Status bits			Flag query	Explanation
Flags	PG display readout	JP... CM...		
CY=1 CY=0	C	...C ...CN	A CY AN CY	Carry Carry Not
O=1 O=0	O	...O ...ON	A O AN O	Overflow Overflow Not
Z=1 Z=0	Z	...Z ...N	A Z AN Z	Zero Not Zero
N=1 N=0	N	...M ...P	A N AN N	Negative / Minus Positive
AG=1 AG=0	 N∨Z	...AG ...MZ	AN Z (AN O AN N O O A N) A Z O N AN O ON N A O	Arithmetical greater Minus / Zero
LG=1		...LG	AN Z AN CY	Logical greater
LG=0	C∨Z	...CZ	A Z O CY	CarryZero

9.3 Key to Abbreviations

OPP		Operation / operator
OPA		Operand attribute
	X	Bit (X may be omitted)
	B	Byte
	W	Word
SRC		Source operand
DEST		Destination operand
	I	Input
	II	Interface input
	EI	Extended input
	O	Output
	IO	Interface output
	EO	Extended output
	M	Marker
	T	Time (timer)
	C	Counter
	D	Data word (within data modules)
	DF	Data field
	OC	Onboard counter
	S	System area
	K	Constant
	DM	Data module
	DX	2nd active data module
	FC	Program module (function call)
	SYM	Symbolic (max. 8 characters)
	R.bit	Register bit, with R = A, B, C, D, and bit = 0 thru 15
	OPD[R]	Register indirect, with operand prefix
	TI	Timed interrupt (time-controlled processing)
	PI	Peripheral interrupt
RG		Program rung
	A	Permitted operation at RG start
	E	Operation concluding RG
AddrMode		Addressing mode
	D	Direct
	R	Register (A, B, C or D)
	[R]	Register indirect, with operand prefix

Status bits / Special markers

V	Result RES
C	Carry
O	Overflow
Z	Zero
N	Negative

9.4 Binary Links & Parenthesized Instructions

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example		Comment
OPT	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)			
U		I/O/M		•		•			•					4*	0.3	A	I0.0	AND link, query for status '1'
		T/C/SYM		•		•			•							A	T0	
		R.bit		•			•		•							A	A.0	4*: Length is valid for branch centre only; for start of branch, + 2 bytes.
		P		•		•			•					10	11.0	A	P0	
		OPD[R]		•				•	•					10	8.3	A	M[O]	
		CY,Z,N,O		•		•			•					4*	0.3	A	CY	
UN		I/O/M		•		•			•					6*	0.6	AN	O0.0	AND link, query for status '0'
		T/C/SYM		•		•			•							AN	Z0	
		R.bit		•			•		•							AN	B.0	6*: Length is valid for start of branch only; for branch centre, + 2 bytes
		P		•		•			•					10	11.0	AN	P1	
		OPD[R]		•				•	•					10	8.3	AN	M[A]	
		CY,Z,N,O		•		•			•					6*	0.6	AN	Z	
O		I/O/M		•		•			•					8	0.6	O	M0.0	OR link, query for status '1'
		T/C/SYM		•		•			•							O	-SYMBOL	
		R.bit		•			•		•							O	C.0	
		P		•		•			•					10	11.0	O	P10	
		OPD[R]		•				•	•					10	8.3	O	M[A]	
		CY,Z,N,O		•		•			•					8	0.6	O	N	
ON		I/O/M		•		•			•					8	0.6	ON	M31.7	OR link, query for status '0'
		T/C/SYM		•		•			•							ON	-name	
		R.bit		•			•		•							ON	D.0	
		P		•		•			•					10	11.0	ON	P1	
		OPD[R]		•				•	•					10	8.3	ON	M[A]	
		CY,Z,N,O		•		•			•					8	0.6	ON	O	
=		O/M/SYM			•	•								8	0.6	=	O0.0	Assign result when RES = 1
		P			•	•								10	11	=	P0	
		R.bit			•		•							8	0.6	=	O.0	
		OPD[R]			•			•						10	8.3	=	M[A]	
S		O/M/SYM			•	•								10	0.75	S	O0.0	Set bit when RES = 1
		P			•	•								10	11.0	S	P0	
		R.bit			•		•							10	0.75	S	O.0	
		OPD[R]			•			•						10	8.3	S	M[A]	
R		O/M/SYM			•	•								10	0.75	R	O0.0	Reset bit when RES = 1
		P			•	•								10	11.0	R	P0	
		R.bit			•		•							10	0.75	R	O.0	
		OPD[R]			•			•						10	8.3	R	M[A]	
																		Parentheses facilitate 7-fold nesting
(														6	0.45	(		AND open parenthesis
)														10	0.75	)		close parenthesis
O(														10	0.75	O(		OR open parenthesis
)N														14	1.2	)N		Negation of contents enclosed in parentheses

9.5 Time Programming

The ZE200 provides 128 time circuits, T0 through T127.

These can be utilized in the following modes:

- SP Start time as pulse
- SPE Start pulse extended
- SR Start time as raising delay
- SF Start time as falling delay
- SRE Start time as raising delay extended

Starting non-remanent times (SP, SPE, SR and SRE) require a positive transition of the time start condition to effect the timed start. The start will not occur however, if the start condition is equal to "1" already during the first addressing (1st PLC cycle) subsequent to a start or restart.

Remanent times cause the transition marker to be retained; i.e., whether or not a "1" in the first addressing (1st PLC cycle) after a start or restart will start the timer will depend on the start condition prior to the STOP and/or Power-Off.

The timers are decremented in the I/O state. This means that a timeout will be recognized only during the I/O state, and not during the program cycle!

Because during each I/O state, a timer is decremented by multiples of the defined time matrix, it is good practice to select a resolution that is as fine as possible.

⇒ **To effect a timed start, a positive transition and/or a negative transition for the falling delay of the time start condition is required.**

⇒ **A time start condition that is satisfied immediately upon Power-On does not qualify as a transition!**

Time start condition 0 and 1 may be programmed in different modules or also in immediate succession.

Example:

```
A    B  -log0     ;Time start condition 0
SR      A,T5

A    B  -log1     ;Time start condition 1
SR      A,T5      ;Time start correct
```

9.5.1 Time Instructions

Time starts are activated only when the RES signal undergoes a transition from 0↑1. In advance of the time start, the time value is loaded into the register being used. Reset and stop functions of times are always RES signal-dependent. The time status for logical links is instruction-dependent, and may be taken from the time diagrams.

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example		Comment
OPT	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)			
SP		R	, T		•	•								8	5.1[1]	SP	A,T0	Start time as pulse
			, SYM												8.0[2]	SP	A,-Symbol	
			, P			•									8.9[1] 12.0[2]	SP	A,P0	[1] when RES = stable [2] when RES = 0↑1
SPE		R	, T		•	•								8	5.1[1]	SPE	A,T0	Start pulse extended
			, SYM												8.0[2]	SPE	A,-Symbol	
			, P			•									8.9[1] 12.0[2]	SPE	A,P0	[1] when RES = stable [2] when RES = 0↑1
SR		R	, T		•	•								8	5.1[1]	SR	A,T0	Start time as raising delay
			, SYM												8.0[2]	SR	A,-Symbol	
			, P			•									8.9[1] 12.0[2]	SR	A,P0	[1] when RES = stable [2] when RES = 0↑1
SF		R	, T		•	•								8	5.1[1]	SF	A,T0	Start time as falling delay
			, SYM												8.0[2]	SF	A,-Symbol	
			, P			•									8.9[1] 12.0[2]	SF	A,P0	[1]when RES = stable [2] when RES = 0↑1
SRE		R	, T		•	•								8	5.1[1]	SRE	A,T0	Start time as raising delay extended
			, SYM												8.0[2]	SRE	A,-Symbol	
			, P			•									8.9[1] 12.0	SRE	A,P0	[1] when RES = stable [2] when RES = 0↑1
RT		T			•	•								10	4.4[1]	RT	T0	Reset time when RES = 1
		SYM													5.4[2]	RT	-Symbol	
		P				•									9.0[1] 10.0	RT	P0	[1] when RES = 0 [2] when RES = 1
TH		T			•	•								10	4.8	TH	T0	Timer halt when RES = 1, Time continues when RES = 1
		SYM														TH	-Symbol	
		P				•									8.5	TH	P0	

9.5.2 Time Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	R	R	W	W	W	W	W	W	W	W	W	W
				Time matrix		Time value: 1 - 1023									

Time matrix		
0	0	0: 10 ms
0	1	1: 100 ms
1	0	2: 1 s
1	1	3: 10 s

:

Program entry of time constant:
Kw.r with time value w = 1 -1023
and time matrix r = 0 - 3

Example: Timer T100 shall be started at 15 sec:

```
L    W  t#15s,A  ;15s declaration in 1-sec time matrix of CL200
A       -start
SPE     A,T100
```

Same function with higher matrix resolution, i.e., higher accuracy:

```
L    W  t#1500ms,A  ;15s declaration in 1oo-ms time matrix of CL200
A       -start
SPE     A,T100
```

Timed start with the assistance of the PG time matrix:

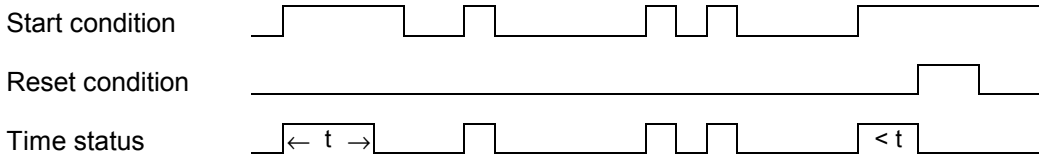
```
L    W  t#15.2,A   ;15s declaration in 1-ms time matrix of PG
A       -start
SPE     A,T100
```

Same function with higher matrix resolution, i.e., higher accuracy:

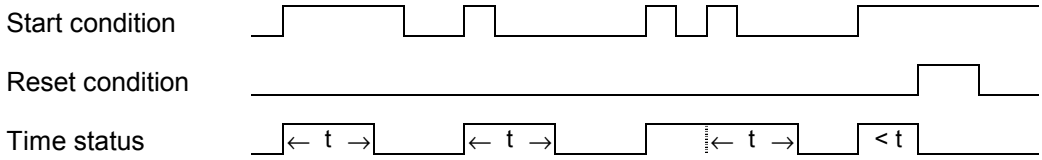
```
L    W  t#150.1,A  ;15s declaration in 1oo-ms time matrix of PG
A    B  -start
SPE    A,T100
```

9.5.3 Time Diagrams

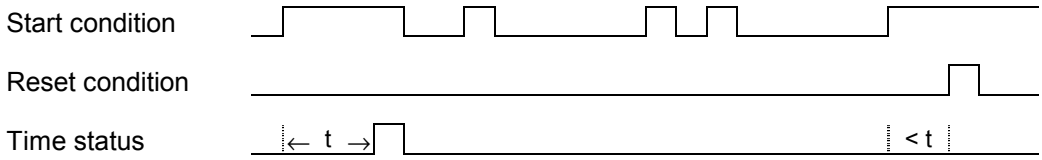
SP – Start time as pulse



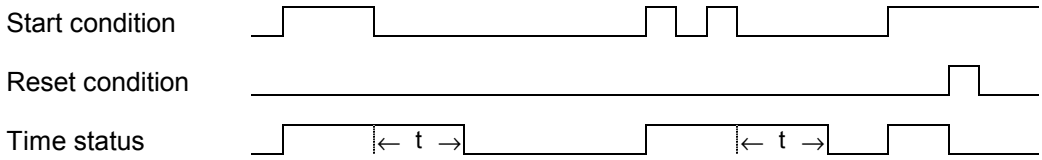
SPE – Start pulse extended



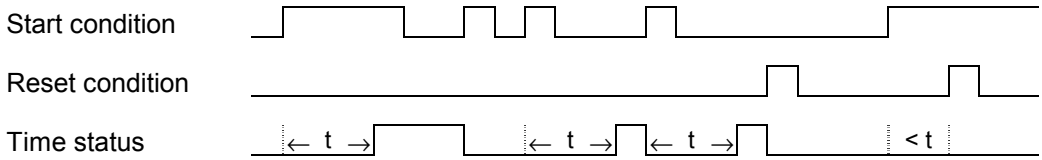
SR – Start time as raising delay



SF – Start time as falling delay



SRE – Start time as raising delay extended



9.6 Counter Instructions

9.6.1 Software Counter

Timer resets and changes of direction of count (up/down) are activated only when the RES signal undergoes a transition from $_0\uparrow^1$.

In advance of the reset, the required counter content is loaded into the register being used.

Counter reset functions are statically RES signal-dependent.

The **counter status** for logical links depends on the counter content. For counter contents > 0, the status is = 1 (HIGH); counter content = 0 will have status = 0 (LOW).

Counter range: 0 – 8191

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example		Comment
OPT	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)			
SC		R	, Z		•	•								8	4.3[1]	SC	A,Z0	Set counter
			, SYM												6.2[2]	SC	A,-Symbol	[1] when RES = stable
			, P			•									81[1] 9.0[2]	SC	A,P0	[2] when RES = $_0\uparrow^1$
CU		C			•	•								10	5.3[1]	CU	Z0	Count upward
		SYM													6.3[2]	CU	-Symbol	[1] when RES = stable
		P				•									9.2[1] 10.2	CU	P0	[2] when RES = $_0\uparrow^1$
CD		C			•	•								10	5.3[1]	CD	Z0	Count downward
		SYM													6.3[2]	CD	-Symbol	[1] when RES = stable
		P				•									9.2[1] 10.2	CD	P0	[2] when RES = $_0\uparrow^1$
RC		C			•	•								10	4.0[1]	RC	Z0	when RES = 1, reset counter
		SYM													6.0[2]	RC	-Symbol	[1] when RES = 0
		P				•									7.8[1] 9.8	RC	P0	[2] when RES = 1

9.6.2 High-speed Counter (onboard counter)

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example		Comment
OPT	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)			
CS		OC			•	•								6	7.5	CS	OC0	Onboard counter Stop when RES = 1

9.7 Digital Links

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example			Comment
OPT	OPE	Q-OPD	Z-OPD	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)				
A	W,B	R	, R				•			0	0	•	•	6	0.45	A	W	A,B	Digital AND link between source and destination. The result is written to destination.
	W,B	K				•				0	0	•	•	8	0.6	A	B	10,A	
AN	W,B	R	, R				•			0	0	•	•	8	0.6	AN	W	A,B	Digital AND NOT link between source and destination. The result is written to destination.
	W,B	K				•				0	0	•	•	10	0.75	AN	B	10,A	
O	W,B	R	, R				•			0	0	•	•	6	0.45	O	W	A,B	Digital OR link between source and destination. The result is written to destination.
	W,B	K				•				0	0	•	•	8	0.6	O	B	10,A	
ON	W,B	R	, R				•			0	0	•	•	8	0.6	ON	W	A,B	Digital OR NOT link between source and destination. The result is written to destination.
	W,B	K				•				0	0	•	•	10	0.75	ON	B	10,A	
XO	W,B	R	, R				•			0	0	•	•	6	0.45	XO	W	A,B	EXCLUSIVE OR link between source and destinationl. The result is written to destination.
	W,B	K				•				0	0	•	•	8	0.6	XO	B	10,A	
XON	W,B	R	, R				•			0	0	•	•	8	0.6	XON	W	A,B	EXCLUSIVE OR NOT link between source and destinationl. The result is written to destination.
	W,B	K				•				0	0	•	•	10	0.75	XON	B	10,A	

9.8 Swap Instruction

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example			Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)				
SWAP	W	R					•							4	0.3	SWAP	W	A	Byte swap in registers High-Byte ↔ Low-Byte

9.9 Compare Instruction

The CPLA (Compare Logical and Arithmetical) instruction is available for Compare operations. As the name implies, the instruction facilitates both logical and arithmetical compare operations.

The logical compare operation regards the bytes and/or words to be compared as unsigned integers, i.e., as "unsigned 8" or "unsigned 16".

The arithmetical compare operation regards the bytes and/or words to be compared as signed integers, i.e., as "integer 8" or "integer 16".

Subsequent to a Compare operation, the flags provide information about the Compare result.

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example			Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)				
CPLA	W,B W,B	K R	, R			•	•			• •	• •	• •	• •	8 6	0.6 0.45	CPLA CPLA	W B	255,B B,C	Arithmetical Compare function. The result may be used for logical and arithmetical purposes. The result can be interpreted logically and arithmetically. **Compare values:** logical: positive, integer arithmet.: two's complement, integer

Binary result evaluation of compare results occurs by means of jump functions dependent upon status bits, and/or through status bit query.

Examples:

Compare Destination (A) w/ Source (B)		CPLA				
		Logical			Arithmetical	
CPLA	B,A	Flag	Status bit		Flag	Status bit
Equal	**A=B**	**JPZ**	**A**	**Z**	**JPZ**	**A Z**
Unequal	A≠B	JPN	AN	Z	JPN	AN Z
Less than	**A<B**	**JPCY**	**A**	**CY**	**JPM**	**AN N A O O N AN O**
Less than or equal	A≤B	JPCZ	A O	Z CY	JPMZ	A Z O N AN O ON N A O
Greater	**A>B**	**JPLG**	**AN AN**	**CY Z**	**JPAG**	**AN Z (AN O AN N O O A N)**
Greater than or equal	A≥B	JPCN	AN	CY	JPP	AN N AN O O N A O

9.10 Load Instructions

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example			Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)				
L	W,BY	I/O/M T/Z/K/SYM DF D/DX II/EI R OPD[R] P	, R			• • • • • •	 • 	 • 						see below *	see below *	L W I0,A L BY 0,B L W 0,C L BY D0,D L W II0,A L BY B,C L W M[C],D L BY P0,A			Load contents of SRC operand into DEST operand. Read value
L	DW	K	, R,R+1			•								12	0.75	L DW K20000H,A			Load constant > 64k into registers R/R+1
LIMR	W	O	, C				•							6	5.8	LIMR W A,C			Load contents of address in A/B to C.
This instruction is used exclusively for the purpose of reference list verification. Example: Verification of whether or not a module is available, or whether a data module of sufficient size is linked!																			

* instruction lengths and processing times

	direct				indirect				as parameter			
	W		BY		W		BY		W		BY	
SRC	Byte	µs	Byte	µs	Byte	µs	Byte	µs	Byte	µs	Byte	µs
K	4	0.3	4	0.3					8	8.8	8	8.8
R	2	0.15	2	0.15								
I,O,M	4	0.3	4	0.3	8	0.6	8	0.6	8	9.0	8	9.0
T	8	3.8	8	3.8	6	3.7	6	3.7	8	11.6	8	11.6
C	4	0.3	4	0.3	12	0.8	12	0.8	8	9.0	8	9.6
S	4	0.3	4	0.3	8	0.6	8	0.6	8	9.0	8	9.6
DF	4	0.3	4	0.3	8	0.6	8	0.6	8	9.0	8	9.6
D,Dx	8	5.0	8	4.7	6	4.8	6	4.5	8	12.9	8	11.6
II	8	51.6	8	34.8	6	50.1	6	35.9	8	52.9	8	42.5
EI	8	51.6	8	34.8	6	50.1	6	35.9	8	52.9	8	42.5

9.11 Transfer Instructions

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example			Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)				
T	W,BY	R	, O/M/SYM , S/DF , D/DX , IO/EO , R , OPD[R] , P			• • • • •	 • 	 • 						see below *	see below *	T W A,M0 T BY B,DF0 T W C,D0 T BY D,IA0 T W A,B T BY B,M[C] T W D,P0			Transfer contents of SRC operand into DEST operand. Write value

* Instruction lengths and processing times

	direct				indirect				as parameter			
	W		BY		W		BY		W		BY	
SRC	Byte	µs	Byte	µs	Byte	µs	Byte	µs	Byte	µs	Byte	µs
R	4	0.3										
O, M, S, DF	4	0.3	8	0.6	8	0.6	8	0.6	8	9.0	8	9.6
D, Dx	8	5.9	8	5.8	6	5.5	6	4.5	8	13.3	8	13.2
IO, EO	8	49.4	8	33.3	6	46.1	6	30.0	8	52.2	8	39.0

9.12 Convert Instructions

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example			Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)				
BID	W,BY	R					•			0	•	0	•	10	17.8 8.9	BID BID	W B	A B	Binary → BCD (decimal) Result > 9999 sets the overflow bit.
DEB	W,BY	R					•			0	•	0	•	10	16.7 10.3	DEB DEB	W B	C D	BCD (decimal) → binary Incorrect BCD encoding sets the overflow bit.
TC	W,BY	R					•			•	•	•	•	6	0.45	TC TC	W B	A B	Converts the register contents to the two's complement.
N	W,BY	R					•			0	•	0	•	8	0.6	N N	W B	C D	Negates the register contents (one's complement).

Representing of positive and negative numbers

A negative number corresponds to the two's complement of the positive number.

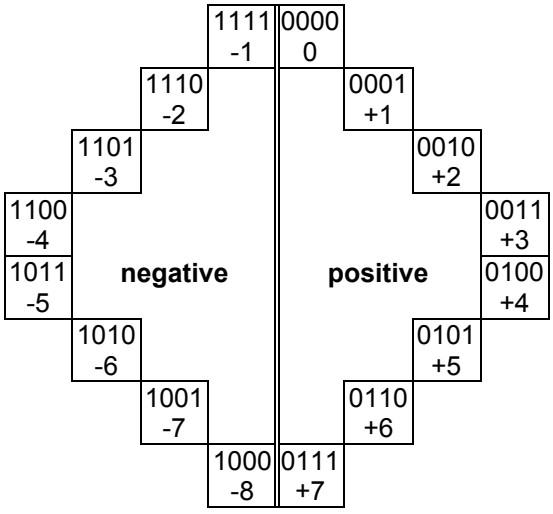
Example: 0 1 1 0 positive integer 6
1 0 0 1 Negation and/or one's complement
\+ 1
1 0 1 0 two's complement = negative integer 6

In the case of word operations, the differentiation between positive and negative integers is determined by Bit 15. In the case of byte operations, this is Bit 7.

Word: Bit 15 = 0 Byte: Bit 7 = 0 positive integer
Bit 15 = 1 Bit 7 = 1 negative integer

Integer value range

Positive integers	Word: 0 - 32767	Byte: 0 – 127
Negative integers	0 - 32768	0 - 128



9.13 Increment / Decrement Instructions

Increment / decrement the source operand (SRC_OPD, 1 to 7 in number):

- **by the number n**
- **when n=0 and when [C], by the number stored in C (max. 7).**

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example			Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)				
INC	W,BY	R	, n			•				•	•	•	•	6	0.45	INC	BY	A,5	Raise (increment) the contents of the SRC_OPD.
			, 0				•			•	•	•	•	10	0.75	INC	W	B,0	
			, [C]					•		•	•	•	•	10	0.75	INC	W	B,[C]	
DEC	W,BY	R	, n			•				•	•	•	•	6	0.45	DEC	BY	A,5	Lower (decrement) the contents of the SRC_OPD.
			, 0				•			•	•	•	•	10	0.75	DEC	W	B,0	
			, [C]					•		•	•	•	•	10	0.75	DEC	W	B,[C]	

9.14 Stack Instructions

⇒ **The available stack size comprises 128 words. In the event of underflow, bit S28.4 in the system area is set; overflow sets system area bit S28.5. The I/O state resets/deletes the entire application stack.**

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example			Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)				
PUSH	W	R					•							4	4.0	PUSH	W	A	Saves the register contents to application stack, and lowers the stack address.
POP	W	R					•							4	4.0	POP	W	B	Raises the application stack address and reads the saved contents from the stack.

9.15 No-operation Instructions & CARRY Manipulations

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example	Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)		
NOP0														2	0.15	NOP0	No-operation with zeroes in the buffer location.
NOP1														2	0.15	NOP1	No-operation with ones in the buffer location.
SCY										•				2	0.15	SCY	Set CARRY bit absolutely to 1.
RCY										•				2	0.15	RCY	Set CARRY bit absolutely to 0.

9.16 Shift Instructions

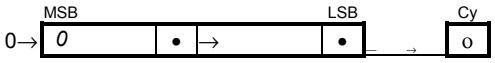
Shift the contents of the source operand (SRC_OPD)

- **by the number n**
- **when n=0, and when [C], by the number stored in C**
- **when OPA = W n: 1-15**
- **when OPA = BY n: 1-7**

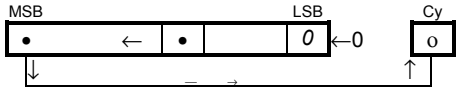
Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example	Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)		
SLR	W,BY	R	, n , 0 , [C]			•	•	•		• • •		0 0 0	• • •	6 10 10	0.45[1)]	SLR W A,7 SLR BY B,[C]	SHIFT logical RIGHT [1)] applies to W, BY + 0.3 µs
SLL	W,BY	R	, n , 0 , [C]			•	•	•		• • •		0 0 0	• • •	6 12 12	0.45[1)]	SLL W A,7 SLL BY B,[C]	SHIFT logical LEFT [1)] applies to W, BY + 0.45 µs
SAR	W,BY	R	, n , 0 , [C]			•	•	•		0 0 0	• • •	• • •	• • •	6 10 10	0.45[1)]	SAR W A,7 SAR BY B,[C]	SHIFT arithmetical RIGHT [1)] applies to W, BY + 0.3 µs

Logical SHIFT

SLR B, n

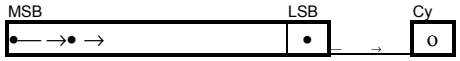


SLL B, n



Arithmetical SHIFT (All bits being vacated are filled up with the contents of the MSB)

SAR B, n



In the case of shift operations exceeding one space, the overflow bit is set after a "1" was shifted through Cy.

9.17 Rotate Instructions

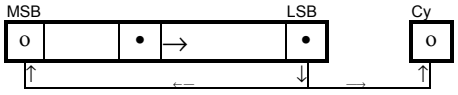
Shift the contents of the source operand:

- **by the number n**
- **when n=0, and when [C], by the number stored in C**
  when OPA = W n: 1-15
- **when OPA = BY n: 1-7**

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example			Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)				
ROR	W,BY	R	, n , 0 , [C]			•				•	•	•	•	6 12 12	0.45[1)]	ROR ROR ROR	BY W W	A,7 A,0 B,[C]	Rotate RIGHT [1)] applies to W, BY + 0.45 µs
							•			•	•	•	•						
								•		•	•	•	•						
ROL	W,BY	R	, n , 0 , [C]			•				•	•	•	•	6 12 12	0.45[1)]	ROL ROL ROL	BY W W	A,7 A,0 B,[C]	Rotate LEFT [1)] applies to W, BY + 0.45 µs
							•			•	•	•	•						
								•		•	•	•	•						
RCR	W,BY	R	, n , 0 [C]			•				•	•	•	•	8	4.35[1)]	RCR RCR RCR	BY W W	A,7 A,0 B,[C]	Rotate RIGHT through CARRY [1)] applies to W, BY + 0.3 µs
							•			•	•	•	•						
								•		•	•	•	•						
RCL	W,BY	R	, n , 0 , [C]			•				•	•	•	•	8	4.8[1)]	RCL RCL RCL	BY W W	A,7 A,0 B,[C]	Rotate LEFT through CARRY [1)] applies to W, BY + 0.6 µs
							•			•	•	•	•						
								•		•	•	•	•						

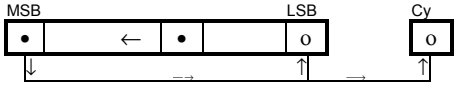
Rotate RIGHT

ROR B, n



Rotate LEFT

ROL B, n

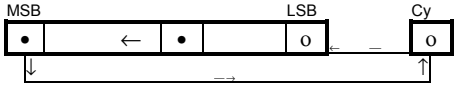


Rotate RIGHT through CARRY

RCR B, n

Rotate LEFT through CARRY

RCL B, n



With Rotate instructions by more than one digit, the following occurs

- The overflow bit is set when a "1" has passed through Cy.
- The negative bit is set, when the MSB contains a "1".
  MSB: Bit 7 when OPA = BY
  Bit 15 when OPA = W

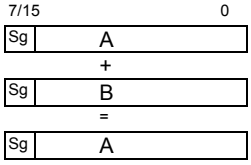
9.18 Arithmetic

9.18.1 Add Instructions

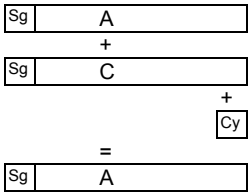
Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example	Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)		
ADD	W,BY	K R	, R			•	•			• •	• •	• •	• •	8[1] 6[2]	0.6 0.45	ADD W 255,B ADD BY B,C	Fixed-point addition of signed integers Source + destination = DEST [1] applies to constant [2] applies to register
ADC	W,BY	K R	, R			•	•			• •	• •	• •	• •	12[1] 10[2]	0.9 0.75	ADC W 255,B ADC BY B,C	Fixed-point addition of signed integers with consideration of CARRY. Source + destination + Cy = DEST [1] applies to constant [2] applies to register

Byte or Word addition

ADD B/W B , A



ADC B/W C , A

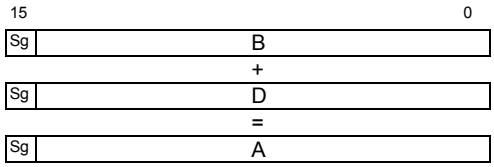


Double-word addition: value 1 + value 2

Value 1: LOW word in B, HIGH word in A
Value 2: LOW word in D, HIGH word in C

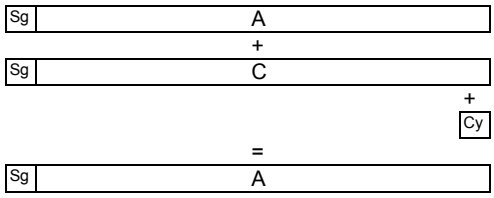
LOW word

ADD W D , B



HIGH word

ADC W C , A



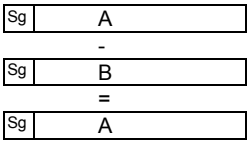
9.18.2 Subtract Instructions

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example	Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)		
SUB	W,BY	K R	, R			•	•			• •	• •	• •	• •	8 6	0.6[1] 0.45[2]	SUB W 255,B SUB BY B,C	Fixed-point subtraction of signed integers. Destination - source = DEST [1] applies to constant [2] applies to register
SBB	W,BY	K R	, R			•	•			• •	• •	• •	• •	12 10	0.9[1] 0.75[2]	SBB W 255,B SBB BY B,C	Fixed-point subtraction of signed integers with consideration of negative CARRY. (- Carry = Borrow) Destination - source - Cy = DEST [1] applies to constant [2] applies to register

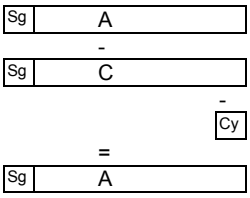
Byte or Word subtraction

Bit 7 when OPA = BY
MSB Bit 15 when OPA = W

SUB *OPA* B , A



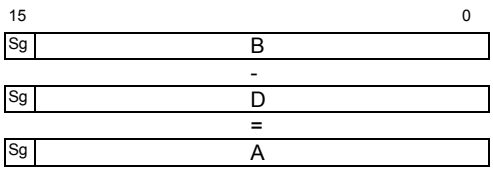
SBB *OPA* C , A



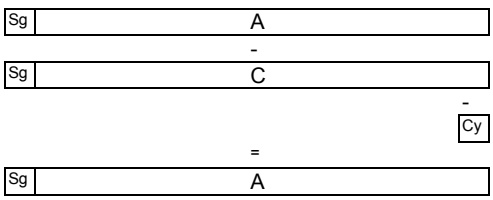
Double-word subtraction: value 1 – value 2

Value 1: LOW word in B, HIGH word in A
Value 2: LOW word in D, HIGH word in C

LOW word
SUB D D , B



HIGH word
SBB W C , A



9.18.3 Multiply Instructions

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example	Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)		
MUL	W,BY	K R	, R			•	 •			0 0	0 0	• •	• •	10 10	5.3[1)] 5.0[1)]	MUL BY 100,A MUL W B,A	Fixed-point multiplication of signed integers. Byte instruction: SRC BY x DEST BY = DEST W Word instruction: SRC W x DEST W = DEST W and DEST W +1. [1)] applies to BY; if W = + 1.0µs

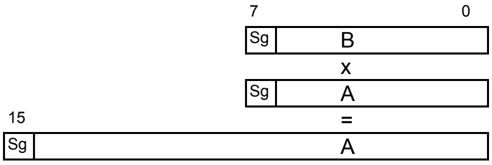
The product of all Multiply operations occupies twice the width of the starting operands.

Byte multiplication

MUL B B , A ; SRC BY

; DEST BY

; DEST W



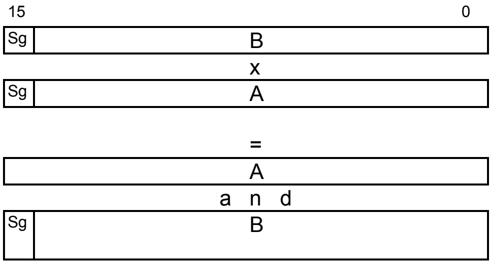
Word multiplication

MUL W B , A ; SRC W

; SRC BY

; DEST W

; DEST W+1



9.18.4 Divide Instructions

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example	Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)		
DIV	W,B	K R	, R			•	•			0 0	• •	• •	• •	10 10	7.1[1)]	DIV B 100,A DIV W B,A	Fixed-point division of signed integers. Byte instruction: DEST W : SRC BY = DEST W (RES remainder.) Word instruction: DEST DW : SRC W = DEST W and DEST W+1 RES remainder 1) applies to W; if BY = + 0.4µs

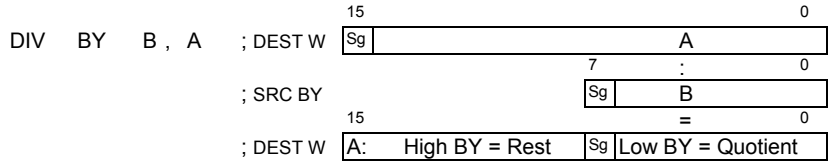
The dividend of all Divide operations occupies twice the width of the divisor.

Note: To simplify command entry, the instruction

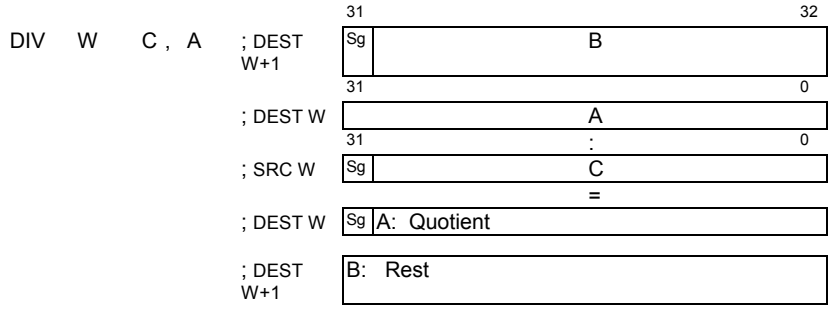
```
L    DW    KhhhhllllH,A
```

may be used.

Byte division



Word division



⇒ **In the case of a Divide operation by 0, the Divide instruction will not be carried out, and the overflow bit will be set. The overflow bit will also be set if the result is > 64k.**

```
Example:    L     W     K2,C
            L     DW    K20000H,A
            DIV   W     C,A        ; O bit set
```

When overflow = 1, the status of the N bit is not defined.

9.19 Parameter Assignments

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example			Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)				
Pn	W BY B	I/O/M/T/C/K II/IO/EI/EO S/SYM D/DX /DF				• • • •								8	-	P0 P1 P2 P3	 W W	E0.0 IE0 S0 D0	Parameter definition for parameterized module calls. n: 0-31

9.20 Local Symbol Names & Auxiliary Flags for Program Tracking

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example			Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)				
DEF		I/O/M/T/C/K II/IO/EI/EO S/SYM D/DX /DF FC/DM	, SYM											-	-	DEF DEF		I0.0,-Symbol I0,-Name	Definition of symbolic names that are valid only within the module ("local") in which they have been entered (essential for the creation of library modules).
*		n n = 0-63												6	6.6	*		1	Definition of auxiliary flags for program tracking. Processing of these aux. flags is entered only in the marker buffer, and is interpretable only in the case of an error. *n has no influence on the program.

9.21 System Variable

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example			Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)				
DEFW	W	K												4	-	DEFW	W	0	Definition of function for system variable in OM2. (Refer to related chapter.)

9.22 Jump Instructions

Jump operations may be executed unconditionally, and also in dependence of a binary link and/or mathematical operation. With one exception, Jump operations are programmed symbolically (exception: JP [R]). The entry point may not be located within a program branch because this would cause the RES at the branching point to be included in the link. This dispenses with the necessity to observe the enter point because that is calculated by the PG programming unit in any case.

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example		Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)			
JP		SYM				•								12	1	JP	-LABEL1	unconditional to -LABEL destination
		[R]						•						12	13.8	JP	[A]	by jump distance (words), in register A
JPC		SYM			•	•			1					12	1	JPC	-LABEL2	conditional, see status bit
JPCI		SYM			•	•			0					12	1	JPCI	-LABEL3	conditional, see status bit
JPCY		SYM				•				1				12	1	JPCY	-LABEL4	conditional, see status bit
JPCN		SYM				•				0				12	1	JPCN	-LABEL5	conditional, see status bit
JPO		SYM				•					1			12	1	JPO	-LABEL6	conditional, see status bit
JPON		SYM				•					0			12	1	JPON	-LABEL7	conditional, see status bit
JPM		SYM				•						1		12	1	JPM	-LABEL8	conditional, see status bit
JPP		SYM				•						0		12	1	JPP	-LABEL9	conditional, see status bit
JPZ		SYM				•							1	12	1	JPZ	-LABEL10	conditional, see status bit
JPN		SYM				•							0	12	1	JPN	-LABEL11	conditional, see status bit
JPAG		SYM				•			For binary flag query, refer to Section 9.2, "Status Bits".					12	1	JPAG	-LABEL12	conditional, see status bits
JPMZ		SYM				•								12	1	JPMZ	-LABEL13	conditional, see status bits
JPLG		SYM				•								12	1	JPLG	-LABEL14	conditional, see status bits
JPCZ		SYM				•								12	1	JPCZ	-LABEL15	conditional, see status bits

The JP [R] instruction causes an unconditional jump whose destination must always be a jump instruction. This instruction variant was created with the objective to facilitate a simple implementation of jump distributors. The controller monitors the instruction mnemonics of the enter point and will enter STOP mode if the former fails to correspond to a valid jump instruction. Information regarding the origin of the error can subsequently be obtained via the error status of the PG programming unit.

Example:

PLC program interlude

```
        Fixed program sequence

        Jump distance calculation in register A for the following jump sequence:
        A may contain odd values (1, 3, 5, ...) only.
        The parameter n may not be smaller than the number of jumps to follow.

JP      [A]             ; 1-word instruction
JP      Dest1           ; 2-word instruction
JP      Dest2           ; 2-word instruction
  :
  :
JP      Destn           ; 2-word instruction
```

```
Dest1:                  ; Partial program 1

PLC program

JP      End
```

```
Dest2:                  ; Partial program 2

PLC program

JP      End
```

:
:
:
:
:

```
Destn:                  ;Partial program n

PLC program

JP      End
```

:
:
End
PLC successor program
:

9.23 Module Calls

Module calls may be executed unconditionally and also in dependence of a binary link and/or a mathematical operation.

The CL200 facilitates a nesting depth of 32 program modules.

Two data modules may be kept enabled at the same time. For this purpose the following module calls are available:

CM, CMC DMx: enables DMx as 1st DM

CX, CXC DMy: enables DMy as 2nd DM

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example		Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)			
CM		DM			•	•								6		CM	DM0	Unconditional, direct
		FC				•								10		CM	FC0	
		FC	, n			•								10		CM	FC1,2	Parameterized, list to follow
		OPD[R]						•						10	↓	CM	FC[A]	Indirect
CMC		DM			•	•								6		CMC	DM0	Conditional, RES-dependent
		FC				•								10		CMC	FC0	Direct
		FC	, n			•								10		CMC	FC1,2	Parameterized, list to follow
		OPD[R]						•						10	↓	CMC	FC[A]	Indirect
CX		DM			•	•								6		CX	DM0	Unconditional, direct
		OPD[R]						•						10	↓	CX	DM[A]	
CXC		DM			•	•								6		CXC	DM0	Conditional, RES-dependent
		OPD[R]						•						10	↓	CXC	DM[A]	

Instruction	not executed	executed
DM, FC(,n)	4.8	10.6
OPD[R]	4.8	20.0

9.24 End Of Module Instructions

End Of Module instructions may be executed unconditionally, and also in dependence of a binary link and/or mathematical operation.

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example	Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)		
EM					•									6	↓	EM	Unconditional
EMC					•									6	↓	EMC	Conditional, RES-dependent

Instruction	not executed	executed
	4.5	18

9.25 Interrupt Instructions

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example		Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)			
TIM		R	, TI/PI			•								8	6.8	TIM	A,TI	Transfer interrupt mask. Writes interrupt mask for disabling or enabling interrupts. The masks were first loaded into a register.
LIM		TI/PI	, R			•								8	6.8	LIM	PI,B	Load interrupt mask Define interrupt mask
EAI		TI/PI				•								6	6.8	EAI	PI	Enable interrupt group (activate)
DAI		TI/PI				•								6	6.8	DAI	PI	Disable interrupt group (deactivate)
LI		TI/PI	, R			•								8	6.8	LI	PI,A	Load interrupt register (Read statuses)
RI		R	, TI/PI			•								8	6.8	RI	A,TI	Resetting interrupts in accordance with previously loaded mask.

9.26 Program Stop / End

Control Instruction				RG		Addr Mode			Influences FLAG					Instr. Length	Proc. Time	Example	Comment
OPP	OPA	SRC	DEST	O	I	D	R	[R]	V	CY	O	N	Z	(Byte)	(µs)		
HLT														6	-	HLT	HALT instruction. The controller enters STOP mode, the program address is entered in error stack, and outputs are cleared (deleted).
EP														6	-	EP	Program End. The I/O state is initialized, and the program cycle start again at the beginning. At least one EP must be present.

1070 072 151-108 (99.12) GB • HB SP • AT/PLS • Printed in Germany