

Programmable Controller CJ1

Replacement Guide From CJ1M/CJ1G to CJ2M

CJ1M-CPU

CJ1G-CPU4□(H)

CJ2M-CPU

Replace Guide



P068-E1-02

About this document

This document provides the reference information for replacing CJ1M/CJ1G PLC systems with CJ2M series PLC.

This document does not include precautions and reminders; please read and understand the important precautions and reminders described on the manuals of PLCs (both of PLC used in the existing system and PLC you will use to replace the existing PLC) before attempting to start operation.

Related Manuals

Man.No.	Manual
W472	CJ2 CPU Unit Hardware USER'S MANUAL
W473	CJ2 CPU Unit Software USER'S MANUAL
W486	CJ2M Pulse I/O Module USER'S MANUAL
W393	CJ Series OPERATION MANUAL
W441	CJ series CJ1M CPU Units with Ethernet Functions OPERATION MANUAL
W395	CJ series Built-in I/O CJ1M CPU Units OPERATION MANUAL
W394	CS/CJ/NSJ PROGRAMMING MANUAL
W474	CS/CJ/NSJ Series INSTRUCTIONS REFERENCE MANUAL
W342	CS/CJ/CP/NSJ Series Communications Commands REFERENCE MANUAL
W345	CS/CJ Series Analog I/O Units AD/DA/MAD42 OPERATION MANUAL
W368	CS/CJ Series Analog I/0 Units OPERATION MANUAL
W466	CJ Series Universal Input Units OPERATION MANUAL
W396	CJ Series Temperature Control Units OPERATION MANUAL
W401	High-speed Counter Units OPERATION MANUAL
W465	EtherNet/IP Units OPERATION MANUAL
W420	CS and CJ Series Ethernet Units OPERATION MANUAL Construction of Networks
W343	CS/CJ Series Ethernet Units OPERATION MANUAL
W421	CS/CJ Series Ethernet Units OPERATION MANUAL Construction of Applications
Z174	CS/CJ Series ID SENSOR UNITS OPERATION MANUAL
W397	CJ Series Position Control Units CJ1W-NC 3 OPERATION MANUAL
W477	CJ Series Position Control Units CJ1W-NC 4 OPERATION MANUAL
W336	CS/CJ Series Serial Communications Boards Serial Communications Units OPERATION MANUAL
W426	CS/CJ Series Position Control Units CS1W-NC 1/CJ1WNC 1-MA OPERATION MANUAL
W435	CS/CJ series Motion Control Unit CS1W/CJ1W-MCH710PERATION MANUAL
W467	Controller Link Support Boards for PCI Bus INSTALLATION GUIDE
W309	Controller Link Units OPERATION MANUAL
V237	SPU-Console Ver.2.1 OPERATION MANUAL
W406	CS/CJ Series Loop Control Boards/Process-control CPU Units /Loop-control CPU Units OPERATION MANUAL
W407	CS/CJ Series Loop Control Boards/Process-control CPU Units /Loop-control CPU Units FUNCTION BLOCK REFERENCE MANUAL
W463	CX-One FA Integrated Tool Package SETUP MANUAL
W446	CX-Programmer OPERATION MANUAL
W447	CX-Programmer OPERATION MANUAL: Function Blocks/Structured Text
W469	CX-Programmer OPERATION MANUAL SFC Programming
W366	CX-Simulator OPERATION MANUAL
W464	CX-Integrator OPERATION MANUAL
W433	CX-Position OPERATION MANUAL
W436	CX-Motion-NCF OPERATION MANUAL
W448	CX-Motion-MCH OPERATION MANUAL

Terms and Conditions Agreement

Read and understand this catalog.

Please read and understand this catalog before purchasing the products. Please consult your OMRON representative if you have any questions or comments.

Warranties.

(a) Exclusive Warranty. Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

(b) Limitations. OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE. Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

(c) Buyer Remedy. Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

See http://www.omron.com/global/ or contact your Omron representative for published information.

Limitation on Liability; Etc.

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Suitability of Use.

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings

and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY OR IN LARGE QUANTITIES WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products.

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Performance Data.

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron's test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron's Warranty and Limitations of Liability.

Change in Specifications.

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron's representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions.

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Microsoft products screen shot(s) reprinted with permission from Microsoft Corporation.

Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.

МЕМО

Table of Contents

WORK FLOW 2	2
1. PERFORMANCE SPECIFICATIONS	6
1.1 CJ1M/CJ2M SPECIFICATIONS COMPARISON 1.2 CJ1G/CJ2M SPECIFICATIONS COMPARISON	
2. SYSTEM CONFIGURATIONS	
2.1 CJ1M/CJ1G/CJ2M SYSTEM COMFIGURATION COMPARISON	
3. MEMORY AREA	
3.1 CJ1M/CJ1G/CJ2M memory area comparison	6
4. EXAMPLE OF CONVERTING LADDER PROGRAM BY CX-PROGRAMMER	3

Appendix

A-1 Instruction operations A-2 Condition flag operations This replacement guide describes the procedure to rebuild the system which uses the CJ1-series PLC by introducing the CJ2M-series PLC instead. The CJ2M-series has functions which can replace the functions and operation of CJ1-series PLC. Take the below work flow to replace your system. Also, refer to the reference pages for details.

Work flow

1) Preliminary Steps: Take the following steps before starting the replacement work.



2) Actual replacement work: Take the steps below to replace the CJ1 to CJ2M.

1. Performance specifications

1.1 CJ1M/CJ2M specifications comparison

The table below lists the major difference in specifications of the CJ1M series and CJ2M series.

	Item	CJ1M-CPU**	CJ2M-CPU**			
Number of I/O		CPU*1: 160 points	2,560 points			
	P - 11 (0	CPU*2: 320 points	_,			
		CPU*3: 640 points				
Program capa	citv	CPU*1: 5k step	CPU*1: 5k step			
og. a oapa	u, j	CPU*2: 10k step	CPU*2: 10k step			
		CPU*3: 20k step	CPU*3: 20k step			
			CPU*4: 30k step			
			CPU*5: 60k step			
Data memory		32k words	32k words			
			EM			
			CPU*1 to *3: 1 bank (32k)			
			CPU*4 to *5: 4 banks (32k x 4)			
Built-in I/O		CJ2*: In:10 points/Out:6 points	Built-in CPU funciton will be available by			
			mounting CJ2M-MD211/CJ2M-MD212. Up			
			to two units can be mounted.			
			In:10 points/Out:6 points (when one unit is			
			used)			
			In:20 points/Out:12 points (when two units			
			are used)			
			Attention: It is possible to use the unit with			
			the CPU Unit of unit version 2.0 or later.			
Length of instr		1-7 steps/one instruction	1-30 steps/one instruction			
Execution	LD instruction	0.10us	0.04us			
time of instruction	MOV instruction	0.30us	0.12us			
Overhead proc	cessing time	CPU*1: 0.7ms	CPU3*: 270us			
	0	CPU*2/*3: 0.5ms	CPU1*: 160us			
Maximum	Number of	CPU*1/CPU*2: 10 units	40 units			
Connectable L	Jnits	CPU*3: 20 units				
Maximum Nur	mber of Expansion	CPU*1/CPU*2: No expansion	3			
Racks		CPU*3: 1				
Clock function		Equipped as a standard function	Equipped as a standard function			
Dimensions (C	CPU Unit)	CPU*1: 90(H)x31(W)x65(D)	CPU*1: 90(H) x 31(W) x 75(D)			
		CPU*2: 90(H)x49(W)x65(D)	CPU*3: 90(H) x 62(W) x 75(D)			
	a officiaria					
Programming		CX-P	CX-P			
Programmin	Programming device for	< Peripheral port connection >	< Peripheral (USB) port > A direct connection can be made between			
g device		Connection with PC requires cables: CS1W-CN*** or CS1W-CN118 +				
connection	personal computer	XW2Z-***S-**	the USB port of the personal computer and the PLC using the			
	computer	<pre>< RS232C port connection ></pre>	commercially-available USB cable			
		Connection with PC requires cables:				
		-	< Serial (RS232C) port connection >			
		XW2Z-***S-CV or XW2Z-***S(-V).	Use the serial cable			
			(XW2Z-200S-CV/500S-CV) to connect the			
			PC and serial port on the CPU Unit. (The			
			CPU3* does not have the RS232C port on			
			it. Mount the RS232C option board			
			(CP1W-CIF01) and connect the cable with			
			the unit)			
	Programming	Available	Not supported			
	Console	C200H-PRO27				
		CQM1-PRO01				
L	1		1			

1.2 CJ1G/CJ2M specifications comparison

The table below lists the major difference in specifications of the CJ1G and CJ2M series.

	Item	CJ1G-CPU4*H/CPU4*	CJ2M-CPU**			
Number of I/O	points	CPU42H/43H: 960 points CPU44/45/44H/45H: 1280 points	2,560 points			
Program capa	city	CPU42H: 10k step CPU43H: 20k step CPU44/44H: 30k step CPU45/45H: 60k step	CPU*1: 5k step CPU*2: 10k step CPU*3: 20k step CPU*4: 30k step CPU*5: 60k step			
Data memory		32k words	32k words			
			CPU*1 to *3: 1 bank (32k) CPU*4 to *5: 4 banks (32k x 4)			
Built-in I/O		-	Built-in CPU funciton will be available by adding the CJ2M-MD211/CJ2M-MD212. Up to two units can be mounted. In:10 points/Out:6 points (when one unit is used) In:20 points/Out:12 points (when two units are used) Attention: It is possible to use the unit with the CPU Unit of unit version 2.0 or later.			
Length of instr	ructions	1-7 steps/one instruction	1-30 steps/one instruction			
Execution time of	LD instruction	CPU4*H: 0.04us CPU4*: 0.08us	0.04us			
instruction	MOV instruction	CPU4*H: 0.20us CPU4*: 0.29us	0.12us			
Overhead pro		CPU4*H : 0.3ms CPU4*: 0.5ms	CPU3*: 270us CPU1*: 160us			
Maximum Connectable L		40 units	40 units			
Racks	mber of Expansion	3	3			
Clock function		Equipped as a standard function	Equipped as a standard function			
Dimensions (C	CPU Unit)	90(H) x 62(W) x 65(D)	CPU1*: 90(H) x 31(W) x 75(D)			
		<u>av 5</u>	CPU3*: 90(H) x 62(W) x 75(D)			
Programming software Programmin g device device for connection personal computer		CX-P < Peripheral port connection > Connection with PC requires cables: CS1W-CN*** or CS1W-CN118 + XW2Z-***S-** < RS232C port connection > Connection with PC requires cables: XW2Z-***S-CV or XW2Z-***S(-V)	CX-P < Peripheral (USB) port > A direct connection can be made between the USB port of the personal computerand the PLC using the commercially-available USB cable < Serial (RS232C) port connection > Use the serial cable (XW2Z-200S-CV/500S-CV) to connect the PC and serial port on the CPU Unit. (The CPU3* does not have the RS232C port on it. Mount the RS232C option board			
	Programming Console	Available C200H-PRO27 CQM1-PRO01	(CP1W-CIF01) and connect the cable with the unit) Not supported.			

2. System Configurations

2.1 CJ1M/CJ1G/CJ2M system comfiguration comparison

Same Power Supply Unit, Special I/O Units, and Basic I/O Unit can be used for CJ1M/CJ1G Series and CJ2M Series.

♦Built-in I/O

CJ1M	CJ1G	CJ2M
Built-in I/O function	Built-in I/O function not supported	Built-in CPU funciton will be available by adding the CJ2M-MD211/CJ2M-MD212 Up to two units can be mounted. *It is possible to use the unit with the CPU Unit of unit version 2.0 or later
In:10 points/Out:6 points Supported by CPU2* only	-	In:10 points/Out:6 points (when one unit is used) In:20 points/Out:12 points (when two units are used)

3. Memory area

3.1 CJ1M/CJ1G/CJ2M memory area comparison

This section explains the difference of the memory area of the CJ1M series, CJ1G series and CJ2M series, using an example of CJ1M-CPU2*, CJ1G-CPU4*H/4* and CJ2M-CPU**.

CI/O area







4. Example of converting ladder program by CX-Programmer

This section explains the method of converting the ladder program using CX-Programmer Ver.9.1. Here, convert the ladder program of CJ1M/CJ1G for CJ2M-CPU^{**} as an example.

♦ Changing model from CJ1M/CJ1G to CJ2M.

As shown on the below figure, select NewPLC1[CJ1M] and right-click or double click it to change the PLC model. Please set the CPU model to the Device Type.

The error report might be displayed if there are instructions which cannot be converted.

Please correct and modify the program using support software function or manually, and execute program check. If errors are detected by the program check, please correct them referring to the error report.



Checking program

Check whether there is problem in the ladder program which was converted from the CJ1M/CJ1G series for CJ2M series.

■ Program check

There are 2 types of program check; automatic check on the CX-Programmer and check conducted by users. CX-Programmer checks the program when "Change model" is executed and the ledder program is converted.

• Automatic program check on the CX-Programmer

Timing of program check	Description
When PLC model is changed.	Program check for each PLC model
	Check for all instructions and all operands.

You can see the check result on the "Compile (Program check)" tab of the Output Window. The left bus-bar on the ladder section window turns red if there is an error in the rung.

• Program check conducted by users

This section describes the procedure of program check, an example of checking result, and explanation of error levels.

<Program check for one program (task)>

- 1. Select the ladder section window or nimonic window to check.
- 2. Select "Program" "Compile (Program check)".

The results of program check will be displayed on the Output Window. Refer to "Results of program check" on the next page for details.

• Checking the entire program

Select "PLC" – "Compile All PLC Programs". You can see the program check results on the Output Window. Refer to "Results of program check" for details.

<Results of program check>

You can see the check result on the "Compile (Program check)" tab of the Output Window. There are three error levels; errors are divided and shown for each level.

When there is no error.

Conversion issues [PLC: 'NewPLC1' (PLC Model 'CQM1H CPU11' to 'CJ2M CPU11') [PLC/Program Name : Programs/NewProgram1] [Ladder Section Name : Section1] [Ladder Section Name : END] NewPLC1 - 0 errors, 0 warnings.
When there are errors.
Compiling [PLC/Program Name : NewPLC1/NewProgram1] [Ladder Section Name : Section1] ERROR: Element at rung 0 (0, 0) is not connected at its output. ERROR: Element at rung 0 (0, 1) is not connected at its output. ERROR: Missing operand at rung 1 (1, 0). ERROR: Missing operand at rung 1 (0, 0). [Ladder Section Name : END]
NewProgram1 - 4 errors, 0 warnings. The programs have been checked with the program check option set to Unit Ver.1.0.

Double-click an error on the Output Window to jump to the correposnding cell. Numeric data in (,) shows the position of a cell with an error.

If you right-click on the Output Window, below menus are shown.

Menu	Functions
[Clear]	Clears the content of Output Window.
	Same as selecting "Edit" – "Clear Compile Window".
[Next Reference]	Jump to the error cell next to the error now selected.
	Same as selecting "Edit" – "Next Reference".
[Allow Docking]	Output Window is shown on the main window of the
	CX-Programmer. If unckeck the check box, Output
	Window will be shown on the separate window.
[Hide]	Close the output window.
	Same as selecting "View" – "Window" – "Output".
[Float In Main Window]	Output window will be changed to other window
	(ex. Ladder section window).

Conversion: **= Support software converts the instruction./*= Support software converts the instruction, but it is necessary to manually modify it. /- = There is no corresponding instruction.

	Blank cells: Support software converts the	instructions	s, though the	are are som	e difference in CQMT	H/CJ1W/CJ1G and CJ	J2M.		0/0/01/01	
	Instructions	CQM1H	CJ1M/CJ1 G	Conversion	Differe Nemonic	FUN No.	and CJ1M/CJ1G/CJ2 Number of operand	M (CQM1H->CJ1M/CJ1 BCD => BIN	G/CJ2M) Settings	Remarks
Sequ	uence input instructions									
	LOAD LOAD NOT	LD LD NOT	LD LD NOT	**						
	AND	AND	AND	**						
	AND NOT OR	AND NOT OR	AND NOT OR	**	+	<u> </u>	+		<u> </u>	
	OR NOT	OR NOT	OR NOT	**						
	AND LOAD	AND LD	AND LD	**						
Seau	OR LOAD uence output instructions	OR LD	OR LD	**						
Joqu	OUTPUT	OUT	OUT	**						
	OUTPUT NOT	OUT NOT		**						
	TR Bits KEEP	TR KEEP	TR KEEP	**						
	DIFFERENTIATE UP	DIFU	DIFU	**						
	DIFFERENTIATE DOWN SET	DIFD	DIFD SET	**						
	RESET	SET RSET	RSET	**						
Sequ	uence control instructions			**						
	END NO OPERATION	END NOP	END NOP	**						
	INTERLOCK	IL	IL	**						
	INTERLOCK CLEAR	ILC	ILC	**						
	JUMP JUMP END	JMP JME	JMP JME	**				Jump No. Jump No.		
Time	er and counter instructions									
		TIM TIMH	TIM TIMH	**						
	HIGH-SPEED TIMER TOTALIZING TIMER	TTIM	TTIM	*		Expansion ->87			Operand3: reset input	
				I					relay No will be deleted.	
	COUNTER	CNT	CNT	**	<u> </u>	·····			Enter the reset input.	
	COUNTER REVERSIBLE COUNTER	CNTR	CNTR	**						
om	parison instructions	CMP	CMP	**	<u> </u>	<u></u>	<u> </u>		<u> </u>	
	COMPARE DOUBLE COMPARE	CMP	CMP	**	<u>+</u>	Expansion ->60	3 (None)->2		t	
	SIGNED BINARY COMPARE	CPS	CPS	**	[Expansion ->114	3 (None)->2			
	DOUBLE SIGNED BINARY COMPARE	CPSL MCMP	CPSL MCMP	**	<u> </u>	Expansion ->115	3 (None)->2		<u> </u>	
	MULTI-WORD COMPARE TABLE COMPARE BLOCK COMPARE	TCMP	TCMP	**	<u> </u>		t		<u> </u>	
	BLOCK COMPARE	BCMP	BCMP	**	<u> </u>					
	AREA RANGE COMPARE DOUBLE AREA RANGE COMPARE	ZCP ZCPL	ZCP ZCPL		<u> </u>	Expansion ->88 Expansion ->116			<u>+</u>	
Data	movement instructions			**						
	MOVE MOVE NOT	MOV MVN	MOV MVN	**	<u> </u>		<u> </u>			
	MOVE NOT	MOVB	MOVB	*				Change bit position		
		-		I				specification from in		
			MOVBC	**		82->568		BCD to in BIN.		
			Ver.3.0 or	I		02->000				
	L		later] MOVD	l	L					
	MOVE DIGIT TRANSFER BITS	MOVD XFRB	MOVD XFRB	**	<u> </u>	Expansion ->62				
	BLOCK TRANSFER	XFER	XFER	*		Expansion ->62		Number of words:		
				l				BCD -> BIN		
			XFERC	**		70->565				
			[Ver.3.0 or	I						
	BLOCK SET	BSET	later] BSET	**		-				
	DATA EXCHANGE SINGLE WORD DISTRIBUTE	XCHG	XCHG	**				Stock longth data act	Use PUSH instruction	
	SINGLE WORD DISTRIBUTE	DIST	DIST	-				Stack length data set in words: BCD -> BIN	instead, for stack	
				l					operation.	
			DISTC [Ver.3.0 or	**		80->566				
			[ver.3.0 or later]	I						
	DATA COLLECT	COLL	COLL	*				Stack length data set	Use FIFO instruction	
				I				in words: BCD -> BIN	instead, for stack	
				1						
		1							operation and read	
	1								operation and read FIFO. Use LIFO instruction	
									FIFO. Use LIFO instruction instead, for stack	
									FIFO. Use LIFO instruction instead, for stack operation and read	
			6011.6	**		81.567			FIFO. Use LIFO instruction instead, for stack	
			COLLC [Ver.3.0 or	**		81->567			FIFO. Use LIFO instruction instead, for stack operation and read	
)ata	shift instructions		[Ver.3.0 or			81->567			FIFO. Use LIFO instruction instead, for stack operation and read	
)ata	SHIFT REGISTER	SFT	[Ver.3.0 or SFT	**		81->567			FIFO. Use LIFO instruction instead, for stack operation and read	
)ata	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER	SFTR ASFT	[Ver.3.0 or SFT SFTR ASFT	**		81->567			FIFO. Use LIFO instruction instead, for stack operation and read	
)ata	SHIFT REGISTER REVERSIBLE SHIFT REGISTER	SFTR	[Ver.3.0 or SFT SFTR	**		81->567	2->3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
)ata	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT	SFTR ASFT WSFT	[Ver.3.0 or SFT SFTR ASFT WSFT	**		81->567	2>3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO.	
lata	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT	SFTR ASFT WSFT ASL ASR	[Ver.3.0 or SFT SFTR ASFT WSFT ASL ASR	**		81->567	2->3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
lata	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT ROTATE LEFT	SFTR ASFT WSFT ASL ASR ROL	Ver.3.0 or SFT SFTR ASFT WSFT ASL ASR ROL	** ** ** ** * * *		81->567	2->3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
lata	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT	SFTR ASFT WSFT ASL ASR ROL ROR	[Ver.3.0 or SFT SFTR ASFT WSFT ASL ASR ROL ROR	**		81->567	2->3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT	SFTR ASFT WSFT ASL ASR ROL	Ver.3.0 or SFT SFTR ASFT WSFT ASL ASR ROL	** ** ** ** ** ** ** **		81->567	2->3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT ROTATE LEFT ROTATE LEFT ROTATE RIGHT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT ment/ decrement instructions	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD	[Ver.3.0 or SFT ASFT ASFT WSFT ASL ASR ROL ROR SLD SRD	** ** ** ** ** ** ** ** ** ** ** **			2>3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT LEFT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT ment/ decrement instructions IINCREMENT	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD INC	Ver.3.0 or SFT SFTR ASFT WSFT ASR ROL ROR SLD SRD ++B	** ** * * * * * * * * * * * * * * *	INC->++B	38->594	2->3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ICTE	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT LEFT ROTATE LEFT ROTATE RIGHT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT ment/ decrement instructions INCREMENT BCD DECREMENT BCD DECREMENT DI math instructions	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD INC DEC	IVer.3.0 or SFT SFTR ASFT WSFT ASL ASL ASL ROL ROL ROR SLD SRD ++B B	** ** ** ** ** ** ** ** ** ** ** ** **	INC->++B DEC->+B	38->594 39-596	2>>3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ICTE	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT ment/ decrement instructions INCREMENT BCD DECREMENT Dol math instructions BINARY ADD	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD INC DEC ADB	Ver.3.0 or SFT SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD ++B B +C	** ** ** ** ** ** ** ** **	INC->++B DEC->B ADB->+C	38->594 39->596 50->402	2->3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ICTE	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT ONE DIGIT SHIFT RIGHT BID DECREMENT BCD DECREMENT BCD DECREMENT BINARY ADD	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD INC DEC ADB ADBL	IVer.3.0 or SFT SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD ++B B +C +CL	** ** ** ** ** ** ** ** ** ** ** ** **	INC->++B DEC->B ADB->+C ADBL->+CL	38->594 39->596 50->402 Expansion ->403	2>3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ICTE	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT RIGHT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT ment/ decrement instructions INCREMENT BCD DECREMENT bol math instructions BINARY ADD DOUBLE BINARY ADD BCD ADD DOUBLE BINARY ADD BCD ADD DOUBLE BCD ADD	SFTR ASFT WSFT ASL ASR ROL ROR SRD INC DEC ADB ADBL ADDL	Ver.3.0 or SFT SFTR ASFT WSFT ASL ASR ROL ROR SRD SRD ++B B +C +C +C +BC +BCL	** ** ** ** ** ** ** ** ** **	INC->++B DEC->-B ADB->+C ADB->+C ADD->+BC ADD->+BC ADD->+BC	38->594 38->596 50->402 Expansion ->403 30->406 54->407	2>3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ICTE	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT LEFT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT ment/ decrement instructions INCREMENT BCD DECREMENT BD Math instructions BINARY ADD BCD ADD DOUBLE BINARY ADD BCD ADD DOUBLE BINARY ADD BINARY SUBTRACT	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD INC DEC ADB ADBL ADBL ADD SBB	Ver.3.0 or SFT SFTR ASFT WSFT ASF ASF ASF ASF ASF ASF ASF ASF ASF ASF		INC->++B DEC->B ADB->+C ADBL->+BC ADDL->+BCL SBB->C	38->594 39->596 50->402 Expansion ->403 30->406 54->407 51->412	2->3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ICTE	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT LEFT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT ment/ decrement instructions INCREMENT BCD DECREMENT BCD DECREMENT BD DECREMENT DOUBLE BINARY ADD DOUBLE BINARY SUBTRACT DOUBLE BINARY SUBTRACT	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD INC DEC ADB ADBL ADBL ADD SBB SBBL	Ver.3.0 or SFT SFTR ASFT WSFT ASR ROR SLD SRD ++B +C +CL +BC +BC -C		INC->++B DEC->B ADB->+C ADB->+C ADD->+BC ADD->+BC ADD->+BC SBB->-C SBB->-CL	38->594 39->596 50->402 Expansion ->403 30->406 54->407 51->412 Expansion ->413	2->3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ncre	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT LEFT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT ment/ decrement instructions INCREMENT BCD DECREMENT BD Math instructions BINARY ADD BCD ADD DOUBLE BINARY ADD BCD ADD DOUBLE BINARY ADD BINARY SUBTRACT	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD INC DEC ADB ADBL ADBL ADD SBB	Ver.3.0 or SFT SFTR ASFT WSFT ASL ASR ROR SLD SRD ++B B +C +CL +BC +BC C		INC->++B DEC->B ADB->+C ADBL->+BC ADDL->+BCL SBB->C	38->594 39->596 50->402 Expansion ->403 30->406 54->407 51->412	2>3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ncre	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT LEFT ROTATE LEFT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT ment/ decrement instructions INCREMENT BCD DECREMENT BOUBLE BINARY ADD BCD DADD DOUBLE BINARY ADD BCD BLOB ADD DOUBLE BINARY SUBTRACT DOUBLE BINARY SUBTRACT DOUBLE BINARY SUBTRACT DOUBLE BINARY SUBTRACT BCD SUBTRACT DOUBLE BCD SUBTRACT SIGNED BINARY MULTIPLY	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD NC DEC ADB ADBL ADD ADDL SBB SBBL SUBL SUBL SUBL	Ver.3.0 or SFT SFTR ASFT WSFT MSFT MSFT ASR ROR SLD SRD SRD ++B B +CL +BCL -C. -CL -BC		INC>++B DEC>B ADB->+C ADB->+C ADD->+BCL SBB->-C SBB->-C SBBL->-CL SUB->-BC SUB->-BC SUB->-BC	38->594 39->596 50->402 Expansion ->403 30->406 54->407 51->412 Expansion ->413 31->416 55->417 Expansion ->420	2>>3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ncre	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT LEFT ROTATE LEFT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT DONE DIGIT SHIFT LEFT BCD DECREMENT BCD DECREMENT BCD DECREMENT BINARY ADD BCD ADD DOUBLE BINARY ADD BCD ADD DOUBLE BINARY ADD BINARY SUBTRACT DOUBLE BINARY SUBTRACT DOUBLE BCD SUBTRACT DOUBLE BCD SUBTRACT DOUBLE BCD SUBTRACT SIGNED BINARY MULTIPLY	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD INC DEC ADB ADBL ADD ADBL ADD SBB SBBL SBBL SUB SUBL MBS	Ver.3.0 or SFT SFTR ASFT WSFT ASR ROL ROR SLD SRD SH ++B +-B ++CL +BC -C -C -C -C -BC -BCL **L		INC->++B DEC->-B ADB->+C ADD->+BC ADD->+BC ADD->+BC SBB->-C SBB->-C SBB->-C SBB->-C SBB->-C SBB->-C SBB->-C SBB->-C SBB->-C SBB->-C	38->594 39->594 39->596 50->402 Expansion ->403 30->406 54->407 51->412 Expansion ->413 31->416 55->417 Expansion ->420 Expansion ->421	2->3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ncre	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT ROTATE LEFT ROTATE RIGHT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT ment/ decrement instructions liNCREMENT BCD DECREMENT bol math instructions BINARY ADD DOUBLE BINARY ADD BCD ADD DOUBLE BINARY ADD BCD ADD DOUBLE BINARY SUBTRACT DOUBLE BO SUBTRACT DOUBLE BCD SUBTRACT SIGNED BINARY MULTIPLY BINARY MULTIPLY	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD NC DEC ADB ADBL ADD ADDL SBB SBBL SUBL SUB SUBL	Ver.3.0 or SFT SFTR ASFT WSFT MSFT MSFT ASR ROR SLD SRD SRD ++B B +CL +BCL -C. -CL -BC		INC>++B DEC>B ADB->+C ADB->+C ADD->+BCL SBB->-C SBB->-C SBBL->-CL SUB->-BC SUB->-BC SUB->-BC	38->594 39->596 50->402 Expansion ->403 30->406 54->407 51->412 Expansion ->413 31->416 55->417 Expansion ->420	2>>3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ncre	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT LEFT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT ment/ decrement instructions INCREMENT BCD DECREMENT BCD DECREMENT BINARY ADD DOUBLE BINARY ADD DOUBLE BINARY ADD DOUBLE BINARY ADD DOUBLE BINARY SUBTRACT BCD SUBTRACT DOUBLE BINARY SUBTRACT BCD SUBTRACT DOUBLE DIS SUBTRACT SIGNED BINARY MULTIPLY BINARY MULTIPLY BINARY MULTIPLY BINARY MULTIPLY BINARY MULTIPLY	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD INC DEC ADB ADB ADB ADB ADB ADDL SBB SBBL SUBL SUBL SUBL MBS MBSL MBS MLB MULL	Ver.3.0 or SFT SFTR ASFT WSFT ASI ASR ROR SLD SRD ++B +CL +BC +BC -BCL -BCL *U		INC->++B DEC->=B ADB->+C ADB->+C ADD->+BC ADD->+BC ADD->+BC SBB->C SBB->C SBB->CL SUB->=BC SUB->=BC MBSL->'L MBSL->'L MBS-'' MBSL->'L MUL->*B UUL->*B	38->594 39->596 50->402 Expansion ->403 30->406 54->407 51->412 Expansion ->413 31->412 Expansion ->420 Expansion ->421 52->422 32->422 32->424 56->425	2->3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ncre	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT ment/ decrement instructions INCREMENT BCD DECREMENT BCD DECREMENT DOUBLE BINARY ADD BCD ADD DOUBLE BINARY ADD BCD ADD DOUBLE BINARY ADD BCD ADD DOUBLE BINARY SUBTRACT DOUBLE BINARY SUBTRACT DOUBLE BCD ADD DOUBLE BCD ADD DOUBLE BINARY SUBTRACT DOUBLE BINARY SUBTRACT DOUBLE BINARY MULTIPLY DOUBLE SCO BINARY MULTIPLY DOUBLE SCO BINARY MULTIPLY BINARY MULTIPLY BINARY MULTIPLY BCD MULTIPLY BCD MULTIPLY	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD NC DEC ADB ADBL ADD ADDL ADD SBBL SUBL SUB SUBL MBS SUBL MBS MUL MULL DBS	Ver.3.0 or SFT SFTR ASFT WSFT ASR ROL SRD +HB B +CL +BC +BCL -C -CL -BC -BC "B"		INC>>++B DEC>B DEC>B ADB->+C ADB->+C ADD->+BCL SBB->-C SBB	38->594 39->596 50->402 Expansion ->403 30->406 54->407 51->412 Expansion ->413 31->416 55->417 Expansion ->421 52->422 32->424 35->425 Expansion ->430	2>3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ncre	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT LEFT ROTATE LEFT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT BCD DECREMENT BCD DECREMENT BCD DECREMENT BINARY ADD BCD ADD DOUBLE BINARY ADD BCD ADD DOUBLE BINARY ADD BCD ADD DOUBLE BINARY SUBTRACT DOUBLE BINARY SUBTRACT DOUBLE BCD SUBTRACT DOUBLE BCD SUBTRACT DOUBLE BCH SUBTRACT SIGNED BINARY MULTIPLY BINARY MULTIPLY DOUBLE BCD MULTIPLY DOUBLE BCD MULTIPLY DOUBLE BCD MULTIPLY DOUBLE BCH SUBTRACT SIGNED BINARY DIVIDE	SFTR ASFT WSFT ASL ASR ROL ROR SLD SRD INC DEC ADB ADB ADB ADB ADB ADDL SBB SBBL SUBL SUBL SUBL MBS MBSL MBS MLB MULL	Ver.3.0 or SFT SFTR ASFT WSFT ASR ROL SRD +HB B +CL +BC +BCL -C -CL -BC -BC "B"		INC->++B DEC->=B ADB->+C ADB->+C ADD->+BC ADD->+BC ADD->+BC SBB->C SBB->C SBB->CL SUB->=BC SUB->=BC MBSL->'L MBSL->'L MBS-'' MBSL->'L MUL->*B UUL->*B	38->594 39->596 50->402 Expansion ->403 30->406 54->407 51->412 Expansion ->413 31->416 55->417 Expansion ->421 52->422 32->424 36->425 Expansion ->430 Expansion ->430 Expansion ->431	2>3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	
ncre	SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT LEFT ROTATE LEFT ROTATE RIGHT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT LEFT DONE DIGIT SHIFT LEFT BCD DECREMENT BCD DECREMENT BCD DECREMENT BCD DECREMENT DOUBLE BINARY ADD BCD ADD DOUBLE BINARY ADD BCD ADD DOUBLE BINARY ADD BCD ADD DOUBLE BINARY SUBTRACT DOUBLE BINARY SUBTRACT DOUBLE BINARY SUBTRACT SIGNED BINARY MULTIPLY BINARY MULTIPLY DOUBLE SIGNED BINARY MULTIPLY BINARY MULTIPLY DOUBLE BINARY DIVIDE DOUBLE SIGNED BINARY MULTIPLY BINARY DIVIDE DOUBLE SIGNED BINARY DIVIDE DOUBLE SIGNED BINARY DIVIDE BINARY DIVIDE	SFTR ASFT WSFT ASL ASR ROL ROR SLD SC SRD INC DEC ADB ADB ADB ADB ADB ADB ADB ADDL SBB SBBL SBB SBBL SUBL MBS UB BS UB BS UB DES	Ver.3.0 or SFT SFTR ASFT WSFT ASFT ASFL A		INC->++B DEC->-B ADB->+C ADD->+BC ADD->+BC ADD->+BC SBB->-B SBB->-C SB	38->594 39->596 50->402 Expansion ->403 30->406 54->407 51->412 Expansion ->413 31->416 55->417 Expansion ->413 31->416 52->422 Expansion ->420 Expansion ->421 52->422 52->425 Expansion ->430	2>3		FIFO. Use LIFO instruction instead, for stack operation and read LIFO. Set the shift sata in	

Conversion: **= Support software converts the instruction./*= Support software converts the instruction, but it is necessary to manually modify it. /- = There is no corresponding instruction.

	Blank cells: Support software converts the Instructions	CQM1H	CJ1M/CJ1 G	Conversion		FUN No.	and CJ1M/CJ1G/CJ2 Number of operand	M (CQM1H->CJ1M/CJ1 BCD => BIN	G/CJ2IM) Settings	Remarks
	version instructions			**						
	BCD-TO-BINARY DOUBLE BCD-TO-DOUBLE BINARY	BIN BINL	BIN BINL	**						
	BINARY TO BCD DOUBLE BINARY-TO-DOUBLE BCD	BCD BCDL	BCD BCDL	**			+			
	2'S COMPLEMENT	NEG	NEG	**		Expansion ->160	3 (None)->2			
	DOUBLE 2'S COMPLEMENT 4-TO-16 DECODER	NEGL MLPX	NEGL MLPX	**		Expansion ->161	3 (None)->2			
	16-TO-4 ENCODER ASCII CONVERT	DMPX ASC	DMPX ASC	**						
	ASCII-TO-HEXADECIMAL	HEX	HEX	**		Expansion ->162				
	LINE	LINE	LINE	*		Expansion ->63		Bit number set in words: BCD -> BIN		
	LINE TO COLUMN	COLM	COLM	*		Expansion ->64		Bit number set in		
ogic	: instructions							words: BCD -> BIN		
Ugic	LOGICAL AND	ANDW	ANDW	**						
	LOGICAL OR EXCLUSIVE OR	ORW XORW	ORW XORW	**						
	EXCLUSIVE NOR	XNRW	XNRW	**						
	COMPLEMENT ial math instructions	COM	COM							
	BSQUARE ROOT	ROOT	ROOT	**		E				
	ARITHMETIC PROCESS BIT COUNTER	APR BCNT	APR BCNT	*		Expansion ->69		Number of words set		
			BCNTC	**		67->621		in words: BCD -> BIN		
			[Ver.3.0 or			07->021				
laat	ing point math instructions	-	later]							
d(ing point math instructions FLOATING TO 16-BIT	FIX	FIX	**	<u> </u>	Expansion ->450	3 (None)->2			·
	FLOATING TO 32-BIT 16-BIT TO FLOATING	FIXL FLT	FIXL FLT	**	 	Expansion ->451 Expansion ->452	3 (None)->2 3 (None)->2			
	32-BIT TO FLOATING	FLTL	FLTL	**		Expansion ->453	3 (None)->2		_	
	FLOATING-POINT ADD FLOATING-POINT SUBTRACT	+F -F	+F -F	**	<u> </u>	Expansion ->454 Expansion ->455			<u> </u>	
	FLOATING-POINT MULTIPLY	*F	*F	**		Expansion ->456				
	FLOATING-POINT DIVIDE DEGREES TO RADIANS	/F RAD	/F RAD	**	+	Expansion ->457 Expansion ->458	3 (None)->2		<u> </u>	+
	RADIANS TO DEGREES	DEG	DEG	**		Expansion ->459	3 (None)->2			
	SINE COSINE	SIN COS	SIN COS	**	<u> </u>	Expansion ->460 Expansion ->461	3 (None)->2 3 (None)->2		<u> </u>	<u> </u>
	TANGENT ARC SINE	TAN ASIN	TAN ASIN	**		Expansion ->462	3 (None)->2 3 (None)->2			
	ARC COSINE	ACOS	ACOS	**	<u> </u>	Expansion ->463 Expansion ->464	3 (None)->2		<u> </u>	<u> </u>
	ARC TANGENT	ATAN SQRT	ATAN SQRT	**		Expansion ->465 Expansion ->466	3 (None)->2 3 (None)->2			·
	EXPONENT	EXP	EXP	**		Expansion ->467	3 (None)->2			
able	LOGARITHM e data processing instructions	LOG	LOG	**		Expansion ->468	3 (None)->2			
10,0	DATA SEARCH	SRCH	SRCH	*	<u> </u>	Expansion ->181		Number of words set	Output selection to	Operand1: 1 word ->
								in words: BCD -> BIN	enable or disable the Outputs number of	words Comparison data,
					L				matches.	result word: C+1 -> Control data: 1word -
	FIND MAXIMUM	MAX	MAX	*		Expansion ->182		Number of words in range: BCD -> BIN,	Select signed or unsigned/Outputs	Control data: 1word - 2 word
								Settings 12 bits -> 15	address to IR or not.	Output address: D+1
	FIND MINIMUM	MIN	MIN	*	<u> </u>	Expansion ->183		bits Number of words in	Select signed or	> IR00 Control data: 1word -
								range: BCD -> BIN,	unsigned/Outputs	2 word
								Settings 12 bits -> 15 bits	address to IR or not.	Output address: D+1 > IR00
	SUM	SUM	SUM	*	t	Expansion ->184		table length: BCD ->	Set the Starting	Control data: 1word -
								BIN, Settings 12 bits - > 15 bits	byte/Units/Data type/signed or not in	2 word
	F00 041 000 475	F00	500		l	F			C+1. Set the Starting	
	FCS CALCULATE	FCS	FCS	*		Expansion ->180		table length: BCD -> BIN, Settings 12 bits -	Set the Starting byte/Units in C+1.	Control data: 1word - 2 word
								> 15 bits	.,	
	control instructions PID CONTROL	PID	PID	*	+	Expansion ->190		Set value: BCD -> BIN	Check setting items	PID parameter area:
					l				and set value.	33ch -> 39ch
	SCALING	SCL	SCL	*		66->194				Acaled value: variabl accepted -> variable
		001.0	0.01 6	**	l					not accepted
	SIGNED BINARY TO BCD SCALING BCD TO SIGNED BINARY SCALING	SCL2 SCL3	SCL2 SCL3	**	<u> </u>	Expansion ->486 Expansion ->487			<u> </u>	+
	AVERAGE VALUE	AVG	AVG	*		Expansion ->195		Number of cycles set	[Average Valid Flag:
								in words: BCD -> BIN		None -> Processing information D15 bit
		000	CDC	**						
	SUBROUTINE ENTRY MACRO	SBS MCRO	SBS MCRO	**	+		+		<u> </u>	Macro area input
		-								words: 96 to 99 ->
										A600 to A603, 196 to 199 -> A604 to A607
										(No influence on the
										ladder program).
		SBN RET	SBN RET	**			+			
	SUBROUTINE RETURN rupt control instructions				 				 	<u> </u>
	INTERRUPT CONTROL	INT	MSKS	*	INT000->MSKS	89->690			Interrupt unit/CJ1M	Interrupt program:
			MSKR CLI		INT001->CLI INT002->MSKR	89->691 89->692			built-in interrupt input: newly configure the	interrupt subroutine - interrupt task (Also
			DI		INT003->MSKS/INI	89->690/880			settings.	change the number
			EI		(CJ1M built-in input only) INT100->DI	89->693				again).
					INT200->EI	89->694				
	INTERVAL TIMER	STIM	MSKS	* (Partly "-")	STIM003 to 005-	69->690	+	Set the operands in	Newly configure the	One-shot interrupt
			MSKR	Instruction	>MSKS	69->692		BCD ->BIN.	settings again.	start: None
				converted if	STIM006 to 008- >MSKR					Stopping timer function: None
				timer start/stop						Set the unit of 0.1ms
ļ				time is						in PLC settings. Interrupt program:
							1	1	i -	michapt program.
				specified.						interrupt subroutine -
				specified.						interrupt subroutine - interrupt task (Newly
100	instructions			specified.						interrupt subroutine

Conversion: **= Support software converts the instruction./*= Support software converts the instruction, but it is necessary to manually modify it. /- = There is no corresponding instruction.

Outcome Outcome <t< th=""><th>Blank cells: Support software converts the Instructions Basic I/O Unit instructions</th><th>CQM1H</th><th></th><th>Conversion</th><th>Differe Nemonic</th><th>nce between CQM1H FUN No.</th><th>and CJ1M/CJ1G/CJ2 Number of operand</th><th>M (CQM1H->CJ1M/CJ1 BCD => BIN</th><th>G/CJ2M) Settings</th><th>Remarks</th></t<>	Blank cells: Support software converts the Instructions Basic I/O Unit instructions	CQM1H		Conversion	Differe Nemonic	nce between CQM1H FUN No.	and CJ1M/CJ1G/CJ2 Number of operand	M (CQM1H->CJ1M/CJ1 BCD => BIN	G/CJ2M) Settings	Remarks
Sector Processor Data Data Particle Processor P	I/O REFRESH	IORF	IORF				<u> </u>		L	
Description Description <thdescription< th=""> <thdescription< th=""></thdescription<></thdescription<>			SDEC	**			3->4		Set the address of First	
DIATE DIATE <th< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>destination word.</td><td></td></th<>									destination word.	
THALM TV V <td>DIGITAL SWITCH</td> <td>DSW</td> <td>DSW</td> <td>*</td> <td></td> <td></td> <td>3->5</td> <td></td> <td>Digits and System</td> <td></td>	DIGITAL SWITCH	DSW	DSW	*			3->5		Digits and System	
BEAGECLINE, Key APUT Name	TEN KEY INPUT	ТКҮ	TKY	**					Word.	
Image: 10 model Image: 10			later]	*			2 . 4		Set the first serieter	
D.O. COMMAND TRANSDOCH DOC - <td></td> <td>нкт</td> <td>[Ver.2.0 or</td> <td>_</td> <td></td> <td></td> <td>3->4</td> <td></td> <td></td> <td></td>		нкт	[Ver.2.0 or	_			3->4			
PRODUCID_LWCKG PALE PALE PALE Constraint Sections no.200 >-4 Sections no.200 No.4		IOTC	lateri —	×						
Image: Source of the second	PROTOCOL MACRO	PMCR	PMCR	*		Expansion ->260	3->4			Change related relay
Image: Section of the sectio										settings.
FROMENT PDD FAD P PAD PAD </td <td></td>										
PRAABANT PAD PAD P PAD								BCD -> BIN	sequence No in the	
HECHVIDE ROD RO	TRANSMIT	TXD	TXD	*		48->236				Peripheral port/serial
RECEIVE ROD ROD <throd< th=""> <throd< <="" td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>communication can not be selected for</td></throd<></throd<>										communication can not be selected for
Introduct NUD Process Automatic Physics Matchine Ph										port spedifier. Change related relay
Lower Stratut Study			5.45							settings.
Delawing SERIAL PORT SETUP STUP · Control of the second set of t	RECEIVE	RXD	RXD	×		47->235		store specified in		Peripheral port/serial communication can
CHANCE BERIAL PORT SETUP STUP Stup <tup<< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>words: BCD -> BIN</td><td></td><td>not be selected for port spedifier.</td></tup<<>								words: BCD -> BIN		not be selected for port spedifier.
Diverse Server STUP STUP Y Pressure Pres										Change related relay settings.
Image: Instructions	CHANGE SERIAL PORT SETUP	STUP	STUP	*		Expansion ->237	3->2			Settings after turning
Image: statutions									method is changed.	
NETWORK SEND SEND SEND - Set the control data spin. Control data spin. Control data spin. Control data spin. DELVER COMAND CAND -										change the related relay settings.
NETWORK RECEIVE RECV RECV -		SEND	SEND	*					Set the control data	Control data: 4 words
NETWORK RECEIVE RECV FECV * DELLYER COMMAND CMMD CMMD * Expension ~460 Set the control data table Control									again.	
DELIVIE COMMAND CMND CMND CMND CMND Control data (structure) Control data	NETWORK RECEIVE	RECV	RECV	*						Control data: 4 words
Image: indications: Image: indid: Image: indid:	DELIVER COMMAND	CMND	CMND	*		Expansion ->490			Set the control data	Change related relays Control data: 5 words
Name MSG MSG <td></td>										
Image: Instructions		MSG	MSG	*			1->2		Set the message	
Index Intervalues Exc Image: constraints Image:	MEGGAGE	MOO	WIGO				1-22		number in the	
ISECONS TO HOURS HMS HMS " Example on HAUMER HMS " Description ITRACE LIENDORY DAMAGE TRSM TRSM " In Operand, enter FALURE ALARM AND RESET FAL FAL * In Operand, enter FALURE ALARM AND RESET FAL * In Operand, enter FALURE ALARM AND RESET FAL * In Operand, enter FALURE ALARM In Operand, enter FALURE CALARM FAL * In Operand, enter FALURE CALARM FAL * In Operand, enter FALURE CALARM FALS * In Operand, enter FALURE POINT DETECT FPD FPD * In Operand, enter FALURE POINT DETECT FPD * In Operand, enter specified in words: EST CARRY Enter CARRY		SEC	SEC	**		Expansion ->65	3 (None)->2			
TTRACE MEMORY SAMPLE TRSM "" Otherage rates Transfer deprese microtions FAL FAL <td>SECONDS TO HOURS</td> <td></td> <td></td> <td>**</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	SECONDS TO HOURS			**						
FALURE ALARM AND RESET FAL FAL FAL FAL Image: Construction of the consequence	TRACE MEMORY SAMPLE	TRSM	TRSM	**						Change related relays.
SEVERE FAILURE ALARM FALS · Inclusion Inclusion Severe FAILURE ALARM FALS · Inclusion Severe FAILURE ALARM FAILS · Inclusion Severe FAILURE ALARM FAILS · Inclusion Severe FAILURE ALARM FAILS · Inclusion Severe CAURURE ALARM · <th< td=""><td></td><td>FAL</td><td>FAL</td><td>*</td><td></td><td></td><td>1->2</td><td></td><td></td><td></td></th<>		FAL	FAL	*			1->2			
SEVERE FAILURE ALARM FALS FALS * Investment Investment Note of ALOO. Work to send message of Error ocide to generate or word containing the error ocide and error details Investment Investment <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>non-fatal error with the</td><td></td></t<>									non-fatal error with the	
SEVERE FAILURE ALARM FALS FALS * 1->2 issue of the segare of error of code to generate or word containing the error details FAILURE POINT DETECT FPD FPD * 1->2 In Operand2, set Frist message or error code and error details Output area segare if a segar									number.	
SEVERE FAILURE ALARM FALS FALS<									send message or Error	
SEVERE FAILURE ALARM FALS FALS * 1->2 In Operand2, set First message wid or error code and error details FAILURE POINT DETECT FPD FPD * Monitoring time specified in words: BCD ->BIN Dupt area and then output mode is set in bit address and message word or error details Output area and then output mode is set in bit address and message word or error details Output area and then output mode is set in bit address and then output mode is set in bit address and then output mode is set in bit address and then output mode is set in bit address and then output mode is set in bit address and then output mode is set in bit address and then output mode is set in bit address and then output message output. Output area and then output hard then output message output. Output area and then output hard										
FAILURE POINT DETECT FPD FPD FPD FPD FPD FPD Configure the operands again if dagnostic again if dagnostin again if dagnostic again if dagnostin again if dag									error details	
FAILURE POINT DETECT FPD FPD ··· Monitoring time spedified in words: BCD ->BIN Configure the operands apain if diagnostic output mote is set in Bit address and message output. Output area apain if diagnostic output mote is set in Bit address and message output. Output area apain if diagnostic output mote is set in Bit address and message output. Output area apain if diagnostic output mote is set in Bit address and message output. Output area apain if diagnostic output mote is set in Bit address and message output. Output area apain if diagnostic output mote is set in Bit address and message output. Output area apain if diagnostic output mote is set in Bit address and message output. Output area apain if diagnostic output mote is set in Bit address and message output. Output area apain if diagnostic output mote is set in Bit address and message output. Output area apain if diagnostic output mote is set in Bit address and message output. Output area apain if diagnostic output mote is set in Bit address and message output. Output area apain if diagnostic output apaint mote is set in Bit address and message output. Output area apaint diagnostic output apaint mote is set in Bit address and message output. Output area apaint diagnostic output apaint mote is set in Bit address and message output apaint diagnostic output apaint apaint diagnostic output apaint dinterest output apaint diagnostic output apai	SEVERE FAILURE ALARM	FALS	FALS	-			1->2		message word or error	
specified in words: BCD ->BIN specified in words: BCD ->BIN specified in words: BIT address and message output. when output address and message output. when output address and message output. when output address and message output address address addr										
ther instructions STC STG STG STG STG STG STG STG	FAILURE POINT DETECT	FPD	FPD	*					again if diagnositic	Output area: When output in codes
Image: set of the set of								BCD ->BIN	Bit address and	= 2 words -> 4 words When output in
Where instructions STC									message output.	character =9 words ->
CLEAR CARRY CLC		STC	STC	**						
MODE CONTROL INI INI INI * 61->880 First word with new PV: BCD ->BIN Refer to 5.1 High- speed counter/pulse output instruction. HIGH-SPEED COUNTER PV READ PRV * 62->881 PV output in BCD -> BIN. Refer to 5.1 High- speed counter/pulse output instruction. Configure th speed counter/pulse output instruction. Set to 5.1 High- speed counter/pulse output instruction. Configure th speed counter/pulse output instruction. In Ring mode speed counter/pulse output instruction. In Ring speed counter/pulse output instruction. In Ring mode speed counter/pulse output instruction. In Ring mode speed counter/pulse output instruction. SET PULSES PULS * 65->886 Number of pulses: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. Refer to 5.1 High- speed counter/pulse output instruction. SPEED OUTPUT SPED * 64->885 Target frequency speed counter/pulse output instruction. Speed counter/pulse output instruction. ACCELERATION CONTROL ACC ACC * <td< td=""><td>CLEAR CARRY</td><td>CLC</td><td></td><td>**</td><td></td><td></td><td></td><td></td><td></td><td></td></td<>	CLEAR CARRY	CLC		**						
HIGH-SPEED COUNTER PV READ PRV PRV * 62->881 PV output in BCD -> BIN. Refer to 5.1 High- speed counter/pulse output instruction. Configure th position of st output instruction. COMPARISON TABLE LOAD CTBL CTBL * 63->883 Number of target values/target values/target Refer to 5.1 High- speed counter/pulse output instruction. In Ring mod the ring value values/target SET PULSES PULS * 66->886 Number of pulses: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. SPEED OUTPUT SPED * 64->885 Target frequency BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. ACCELERATION CONTROL ACC ACC * Expansion ->888 Acceleration/decelerati reguency.SED -> bin Refer to 5.1 High- speed counter/pulse output instruction. PULSE OUTPUT PLS2 PLS2 * Expansion ->887 3->4 Acceleration/decelerati requency.Mumber of output pulses: BCD -> bin Refer to 5.1 High- speed counter/pulse output instruction.		INI	INI	*		61->880				
BIN. speed counter/pulse output instruction. position of st output instruction. COMPARISON TABLE LOAD CTBL CTBL * 63>883 Number of target values/target values/target Number of target values/target speed counter/pulse output instruction. In Ring mod the ring value interrupt ras interrupt ras change the' SET PULSES PULS * 65>886 Number of pulses: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. Refer to 5.1 High- speed counter/pulse output instruction. SPEED OUTPUT SPED * 64->885 Target frequency specified in words: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. ACCELERATION CONTROL ACC ACC * Expansion ->888 Acceleration/decelerati requency/unsection/ BCD -> BIN speed counter/pulse output instruction. PULSE OUTPUT PLS2 PLS2 * Expansion ->887 3->4 Acceleration/decelerati requency/number of output pulses: BCD -> BIN speed counter/pulse output instruction.				*		62->881		-	output instruction.	Configure the referenc
COMPARISON TABLE LOAD CTBL CTBL CTBL In Ring mode 63->883 Number of target value/Interrupt task number: BCD -> BIN Refer to 5.1 High-speed counter/pulse output instruction. In Ring mode SET PULSES PULS PULS 65->886 Number of pulses: Refer to 5.1 High-speed counter/pulse output instruction. Refer to 5.1 High-speed counter/pulse output instruction. SPEED OUTPUT SPED PULS 65->886 Number of pulses: Refer to 5.1 High-speed counter/pulse output instruction. ACCELERATION CONTROL ACC ACC Expansion ->887 Acceleration/decelera						02 2001			speed counter/pulse	position of status data.
SET PULSES PULS * 65->886 Number of pulses: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. PLC setting interrupt tax change the: SET PULSES PULS * 65->886 Number of pulses: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. Refer to 5.1 High- speed counter/pulse output instruction. SPEED OUTPUT SPED * 64->885 Target frequency BCD -> BIN output instruction. Refer to 5.1 High- speed counter/pulse output instruction. ACCELERATION CONTROL ACC ACC * Expansion ->888 Acceleration/decelerati frequency/score.BCD -> output instruction. PULSE OUTPUT PLS2 PLS2 * Expansion ->887 3->4 Acceleration/decelerati frequency/number of output pulses: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction.	COMPARISON TABLE LOAD	CTBL	CTBL	*		63->883			Refer to 5.1 High-	In Ring mode, enter
SET PULSES PULS * 66->886 Number of pulses: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. SPEED OUTPUT SPED SPED * 64->885 Target frequency specified in words: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. ACCELERATION CONTROL ACC ACC * Expansion ->888 Acceleration/deceleration/deceleration/deceleration/ on rate/target frequency/number of pulses: speed counter/pulse Refer to 5.1 High- speed counter/pulse output instruction. PULSE OUTPUT PLS2 PLS2 * Expansion ->887 3->4 Acceleration/deceleration on rate/target on rate/target on rate/target on rate/target on rate/target on rate/target on rate/target on rate/target on rate/target output pulses: BCD -> BIN Refer to 5.1 High- speed counter/pulse output pulses: BCD -> BIN								value/Interrupt task		PLC settings.
SET PULSES PULS PULS 65->886 Number of pulses: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. SPEED OUTPUT SPED SPED 64->885 Target frequency specified in words: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. ACCELERATION CONTROL ACC ACC * Expansion ->888 Acceleration/deceleration/ on rate/target frequency: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. PULSE OUTPUT PLS2 PLS2 * Expansion ->887 3->4 Acceleration/deceleration on rate/target frequency: BCD -> BIN Refer to 5.1 High- speed counter/pulse output pulses: BCD -> BIN								number: BCD -> BIN		Interrupt program: interrupt subroutine ->
SET PULSES PULS PULS * 66->886 Number of pulses: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. SPEED OUTPUT SPED SPED 64->885 Target frequency specified in words: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. ACCELERATION CONTROL ACC ACC * Expansion ->888 Acceleration/decelerati frequency: specified in words: BCD -> BIN Refer to 5.1 High- speed counter/pulse output instruction. PULSE OUTPUT PLS2 PLS2 * Expansion ->887 3->4 Acceleration/deceleration/deceleration/ on rate/target frequency/number of output pulses: BCD -> BIN Speed counter/pulse output pulses: BCD -> BIN										interrupt task (Also change the task No.).
SPEED OUTPUT SPED SPED 64->885 Target frequency specified in words: BCD > BIN Refer to 5.1 High- output instruction. ACCELERATION CONTROL ACC ACC * Expansion ->888 Acceleration/decelerati/deceleration/deceleration/deceleration/deceleration/decele	SET PULSES	PULS	PULS	*		65->886				
ACCELERATION CONTROL ACC ACC Expansion ->888 speed counter/pulse output instruction. PULSE OUTPUT PLS2 PLS2 Expansion ->887 3->4 Acceleration/de		005-	005-						output instruction.	
ACCELERATION CONTROL ACC ACC Expansion ->888 Acceleration/deceleration	SPEED OUTPUT	SPED	SPED	*		64->885		specified in words:	speed counter/pulse	
PULSE OUTPUT PLS2 PLS2 * Expansion ->887 3->4 on rate/target frequency: ECD -> on rate/target frequency/number of output pulses: BCD -> BIN speed counter/pulse output instruction. output pulses: BCD -> BIN	ACCELERATION CONTROL	ACC	ACC	*		Expansion ->888		Acceleration/decelerati	Refer to 5.1 High-	
on rate/target speed counter/pulse frequency/number of output pulses: BCD -> BIN								on rate/target	speed counter/pulse	
frequency/number of output instruction. output pulses: BCD -> BIN	PULSE OUTPUT	PLS2	PLS2	*		Expansion ->887	3->4	Acceleration/decelerati	Refer to 5.1 High-	
BIN								frequency/number of		
		DW/M		*				BIN	Refer to 5.1 High	
POLSE WITH VARIABLE DUTT FACTOR PWW PWW EXpansion->og1 Duty factor specified - Relet to 5. Figh- in words: BCD->BIN speed counter/pulse output instruction.	FULSE WITH VAKIABLE DUTY FACTOR	r'vvM	r' vviVI	-		Expansion ->891			speed counter/pulse	

Appendix

A-2 Condition flag operations

Conversion: *** = same condition flag operation, ** = a part of condition flag operation differs, - = Different condition flag operation, None = no corresponding instruction Condition flags: Left of "/" = Operation of CQM1H. Right of "/" = Operation of CJ1M/CJ1G/CJ2M No "/" = Same operation in CQM1H and CJ *= ON/OFF depending on the instruction statuus

*= ON/OFF depe	ang on t	CJ1M/CJ			C	Condition	n flags ((CJ) = (CQM1H	does n	ot have this s	ettings	.)	
Instructions	CQM1H		Conversion	ER	GT(>)					LE(CJ)		UF		N (CJ)
						(CJ)								
Sequence input instructions								 						
	LD	LD	***					 						 -
LOAD NOT AND	LD NOT AND	LD NOT AND	***					<u> </u>		 -				<u> </u>
AND NOT		AND NOT	***					<u> </u>						<u> </u>
OR	OR	OR	***					<u> </u>						<u> </u>
OR NOT	OR NOT	OR NOT	***				1	<u> </u>		1				<u> </u>
AND LOAD	AND LD	AND LD	***					<u> </u>		†				1
OR LOAD	OR LD	OR LD	***											
Sequence output instructions														
OUTPUT		OUT	***					 						ļ
OUTPUT NOT		OUT NOT	***					 		 				
TR Bits	TR	TR	***					 						
		KEEP	***					<u> </u>						<u> </u>
DIFFERENTIATE UP DIFFERENTIATE DOWN	DIFU DIFD	DIFU DIFD	***					<u> </u>						<u> </u>
SET	SET	SET	***					<u> </u>		<u> </u>				<u> </u>
RESET	RSET	RSET	***				1	<u> </u>		1				<u> </u>
Sequence control instructions														
END	END	END	-	OFF/	OFF/		OFF/	t	OFF/	1	OFF/	OFF/	OFF/	1
NO OPERATION	NOP	NOP	***					[[]	I
INTERLOCK	IL	IL	***					ļ		ļ				ļ
INTERLOCK CLEAR	ILC	ILC	***		ļ		ļ	ļ	ļ	ļ		ļ	ļ	ļ
JUMP	JMP	JMP	- ***	/*			 -	 	 -	 -			 	
JUMP END	JME	JME	***				 			 			<u> </u>	
Timer and counter instructions	TINA	TINA	***	*	 		 	┟		 		+		{
HIGH-SPEED TIMER	TIM TIMH	TIM TIMH	***	*	 			 	<u> </u>	<u> </u>	<u> </u>		+	<u> </u>
TOTALIZING TIMER	TTIM	TTIM	***	*				<u> </u>		<u> </u>				<u> </u>
COUNTER	CNT	CNT	***	*				<u> </u>						<u> </u>
REVERSIBLE COUNTER	CNTR	CNTR	***	*				<u> </u>		<u> </u>				<u> </u>
Comparison instructions														
COMPARE	CMP	CMP	**	*	*	/*	*	/*	*	/*				†
DOUBLE COMPARE	CMPL	CMPL	**	*	*	/*	*	/*	*	/*				1
SIGNED BINARY COMPARE	CPS	CPS	**	*	*	/*	*	/*	*	/*]	I
DOUBLE SIGNED BINARY COMPARE	CPSL	CPSL	**	*	*	/*	*	/*	*	/*				
MULTI-WORD COMPARE		MCMP	***	*			*	 						ļ
TABLE COMPARE	TCMP	TCMP	**	*/OFF				 						l
BLOCK COMPARE	BCMP ZCP	BCMP ZCP	***		<u>*</u>		/*	 	*					
AREA RANGE COMPARE DOUBLE AREA RANGE COMPARE	ZCP ZCPL	ZCP	***	*	*		*	<u> </u>	*	 -				<u> </u>
Double AREA RANGE COMPARE	ZUPL	ZUPL												
MOVE	MOV	MOV	**	*			*	<u> </u>		+				/*
MOVE NOT	MVN	MVN	**	*			*	<u> </u>		<u> </u>				/* /*
MOVE BIT		MOVB	***	*				<u> </u>		1				<i>-</i>
		MOVBC	***	*			1			1				1
		[Ver.3.0												
		or later]												
MOVE DIGIT	MOVD	MOVD	***	*				 						
TRANSFER BITS		XFRB		*/OFF				 						ļ
BLOCK TRANSFER	XFER	XFER	-	*/OFF	 		 	 		 	 		l	
		XFERC	***	*										
		[Ver.3.0												
BLOCK SET	BSET	or later] BSET	***	*	<u> </u>		f	<u> </u>	<u> </u>	<u> </u>	+	+	<u> </u>	<u> </u>
DATA EXCHANGE	XCHG	XCHG	+ -	*/	 		†	<u> </u>	<u> </u>	t	<u> </u>	+	t	t
SINGLE WORD DISTRIBUTE		DIST	**	*/OFF			*	t	<u> </u>	t	<u> </u>	1	<u> </u>	/*
		DISTC	**	*	1		*	t	t	1		1	1	/*
		[Ver.3.0												
i [or later1		L	L		L	L	l	L	l	<u> </u>		L
DATA COLLECT	COLL	COLL	**	*/OFF		<u></u>	*	[[[[]]	/*
		COLLC	**	*			*	[1		/*
		[Ver.3.0												1
		or later]											I	
Liggs chitt instructions	057	057	 	/+	 			}	<u> </u>	 	·			
Data shift instructions	SFT	SFT	- ***	/*	 		ł	}	<u> </u>	+	*			{
SHIFT REGISTER	0000	SFTR	***	*			<u> </u>	 		} -		+	<u> </u>	{
SHIFT REGISTER REVERSIBLE SHIFT REGISTER			***		1	L	l	+	<u> </u>	ł	 			<u> </u>
SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER	ASFT	ASFT	***	*										1
SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT	ASFT	ASFT					*				*			/*
SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT	ASFT WSFT ASL	ASFT WSFT ASL	***	*/OFF			*				*		 	/* */OFF
SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT	ASFT WSFT ASL	ASFT	*** ** ** **				* * *	 			*		 	/* */OFF /*
SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT	ASFT WSFT ASL ASR	ASFT WSFT ASL ASR	*** ** **	*/OFF */OFF			* * * *				* * * *		 	/*
SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT ROTATE LEFT	ASFT WSFT ASL ASR ROL	ASFT WSFT ASL ASR ROL	*** ** ** ** ** **	*/OFF */OFF */OFF			* * * *				* * * *			
SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT ROTATE LEFT ROTATE RIGHT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT	ASFT WSFT ASL ASR ROL ROR	ASFT WSFT ASL ASR ROL ROR	*** ** ** ** **	*/OFF */OFF */OFF */OFF			* * *				* * * *			/*
SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT Increment/ decrement instructions	ASFT WSFT ASL ASR ROL ROR SLD SRD	ASFT WSFT ASL ASR ROL ROR SLD SRD	*** ** ** ** ** ** **	*/OFF */OFF */OFF */OFF *			*				*			/*
SHIFT REGISTER REVERSIBLE SHIFT REGISTER ASYNCHRONOUS SHIFT REGISTER WORD SHIFT ARITHMETIC SHIFT LEFT ARITHMETIC SHIFT RIGHT ROTATE LEFT ROTATE LEFT ONE DIGIT SHIFT LEFT ONE DIGIT SHIFT RIGHT Increment/ decrement instructions INCREMENT	ASFT WSFT ASL ASR ROL ROR SLD	ASFT WSFT ASL ASR ROL ROR SLD	*** ** ** ** ** **	*/OFF */OFF */OFF */OFF			*				*			/*

Appendix

A-2 Condition flag operations

Conversion: *** = same condition flag operation, ** = a part of condition flag operation differs, - = Different condition flag operation, None = no corresponding instruction Condition flags: Left of "/" = Operation of CQM1H. Right of "/" = Operation of CJ1M/CJ1G/CJ2M No "/" = Same operation in CQM1H and CJ *= ON/OFF depending on the instruction statuus

Instructions	CQM1H	CJ1M/CJ ²	IG Conversion	ER				(CJ) = C NE (CJ)			ot have this second	ettings.		N (CJ)
Instructions	CQIVITH	/CJ2IVI	Conversion	ER	GT(>)	GE (CJ)		NE (CJ)	L1(<)	LE(CJ)	CY	UF	OF	N (CJ)
Symbol math instructions						(00)								
BINARY ADD	ADB	+C	**	*/OFF			*				*	*	*	/*
DOUBLE BINARY ADD BCD ADD	ADBL ADD	+CL	**	*/OFF *			*				*			/*
DOUBLE BCD ADD	ADD	+BC +BCL	***	*			*				*			+
BINARY SUBTRACT	SBB	-C	**	*/OFF			*				*	*	*	/*
DOUBLE BINARY SUBTRACT	SBBL	-CL	**	*/OFF			*				*	*	*	/*
BCD SUBTRACT	SUB	-BC	***	*			*				*			
DOUBLE BCD SUBTRACT	SUBL	-BCL	***	* */0FF			*				*			/*
SIGNED BINARY MULTIPLY DOUBLE SIGNED BINARY MULTIPLY	MBS MBSL	*L	**	*/OFF			*							/*
BINARY MULTIPLY	MLB	*U	**	*/OFF			*							/*
BCD MULTIPLY	MUL	*В	***	*			*							1
DOUBLE BCD MULTIPLY	MULL	*BL	***	*			*							
SIGNED BINARY DIVIDE	DBS	/	**	*			*							/*
DOUBLE SIGNED BINARY DIVIDE BINARY DIVIDE	DBSL DVB	/L /U	**	*			*							/* /*
BCD DIVIDE		/0 /B	***	*			*							· /
DOUBLE BCD DIVIDE	DIV DIVL	/BL	***	*			*							+
Conversion instructions														
BCD-TO-BINARY	BIN	BIN	**	*			*							*/OFF
	BINL	BINL	**	*			*					 		*/OFF
BINARY TO BCD DOUBLE BINARY-TO-DOUBLE BCD	BCD BCDL	BCD BCDL	***	*			*							ł
2'S COMPLEMENT	NEG	NEG	**	*/OFF			*					*/		/*
	NEGL	NEGL	**	*/OFF			*				<u> </u>	*/		/* /*
4-TO-16 DECODER	MLPX	MLPX	***	*				t		t		1		1
16-TO-4 ENCODER	DMPX	DMPX	***	*										I
ASCII CONVERT	ASC	ASC	***	*										ļ
ASCII-TO-HEXADECIMAL	HEX	HEX	***	*										
LINE LINE TO COLUMN	LINE COLM	LINE COLM	***	*			*							
Logic instructions	COLINI	COLIVI												
LOGICAL AND	ANDW	ANDW	**	*/OFF			*							/*
LOGICAL OR	ORW	ORW	**	*/OFF			*							/* /*
EXCLUSIVE OR	XORW	XORW	**	*/OFF			*							
EXCLUSIVE NOR		XNRW	**	*/OFF			*							/*
COMPLEMENT	COM	COM	**	*/OFF			*							/*
Special math instructions BSQUARE ROOT	ROOT	ROOT	***	*			*							+
ARITHMETIC PROCESS	APR	APR	**	*			*							/*
BIT COUNTER	BCNT	BCNT	***	*			*							<i>-</i>
		BCNTC	***	*			*							1
		[Ver.3.0												
Flanting a pint moth in struction of		or later]												
Floating point math instructions FLOATING TO 16-BIT	FIX	FIX	**	*			*							/*
FLOATING TO 32-BIT	FIXL	FIXL	**	*			*							/*
16-BIT TO FLOATING	FLT	FLT	**	*/			*							/*
32-BIT TO FLOATING		FLTL	**	*/			*							/*
FLOATING-POINT ADD	+F	+F	**	*			*					*	*	/*
FLOATING-POINT SUBTRACT	-F	-F	**	*			*					*	*	/*
FLOATING-POINT MULTIPLY	*F	*F /F	**	*			*					*	*	/* /*
FLOATING-POINT DIVIDE DEGREES TO RADIANS	/F RAD	/F RAD	**	*			*	<u> </u>		<u> </u>	<u> </u>	*	*	/* /*
RADIANS TO DEGREES	DEG	DEG	**	*			*	<u> </u>		<u> </u>	+	*	*	/*
SINE	SIN	SIN	**	*			*	[<u></u>		[OFF/	OFF/	/*
COSINE	COS	COS	**	*			*					OFF/	OFF/	/*
TANGENT	TAN	TAN	** **	*			*					OFF/	*	/*
ARC SINE		ASIN	**	*			*						OFF/ OFF/	/*
ARC COSINE ARC TANGENT	ACOS ATAN	ACOS ATAN	**	*			*			<u> </u>		OFF/	OFF/	/*
SQUARE ROOT	SQRT	SQRT	**	*			*				<u> </u>	OFF/	*	+ <i>'</i>
EXPONENT	EXP	EXP	***	*			*	t		t		*	*	1
LOGARITHM	LOG	LOG	**	*			*					OFF/	*	/*
Table data processing instructions							ļ							ļ
	SRCH	SRCH	***	*			*					 		14
FIND MAXIMUM	MAX MIN	MAX MIN	**	*			*					+		/* /*
		SUM	**	*			*					<u>+</u>		/* /*
FIND MINIMUM			***	*			f	<u> </u>		<u> </u>	+	<u> </u>		<u> </u>
FIND MINIMUM SUM	SUM FCS	FCS	***				1	1		1	1	1	1	1
FIND MINIMUM SUM FCS CALCULATE Data control instructions	SUM FCS	FCS					l	L		L				т
FIND MINIMUM SUM FCS CALCULATE Data control instructions PID CONTROL	SUM FCS PID	FCS PID	**	*	/*				/*		*			
FIND MINIMUM SUM FCS CALCULATE Data control instructions PID CONTROL SCALING	SUM FCS PID SCL	FCS PID SCL	**	*	/*		*		/*		*			
FIND MINIMUM SUM FCS CALCULATE Data control instructions PID CONTROL SCALING SIGNED BINARY TO BCD SCALING	SUM FCS PID SCL SCL2	FCS PID SCL SCL2	** *** ***		/*		*		/*		*			
FIND MINIMUM SUM FCS CALCULATE Data control instructions PID CONTROL SCALING SIGNED BINARY TO BCD SCALING BCD TO SIGNED BINARY SCALING	SUM FCS PID SCL SCL2 SCL3	FCS PID SCL SCL2 SCL3	** *** *** ***	*	/*				/*		*			/*
FIND MINIMUM SUM FCS CALCULATE Data control instructions PID CONTROL SCALING SIGNED BINARY TO BCD SCALING BCD TO SIGNED BINARY SCALING AVERAGE VALUE	SUM FCS PID SCL SCL2	FCS PID SCL SCL2	** *** ***	*	/*				/*		*			/*
FIND MINIMUM SUM FCS CALCULATE Data control instructions PID CONTROL SCALING SIGNED BINARY TO BCD SCALING BCD TO SIGNED BINARY SCALING AVERAGE VALUE Subroutines instructions	SUM FCS PID SCL SCL2 SCL3 AVG	FCS PID SCL SCL2 SCL3 AVG	** *** *** ***	*					/*		*			/*
FIND MINIMUM SUM FCS CALCULATE Data control instructions PID CONTROL SCALING SIGNED BINARY TO BCD SCALING BCD TO SIGNED BINARY SCALING AVERAGE VALUE	SUM FCS PID SCL SCL2 SCL3 AVG SBS MCRO	FCS PID SCL SCL2 SCL3	** *** *** *** *** ***	* * * *					/*		*			/*
FIND MINIMUM SUM FCS CALCULATE Data control instructions PID CONTROL SCALING SIGNED BINARY TO BCD SCALING BCD TO SIGNED BINARY SCALING AVERAGE VALUE Subroutines instructions SUBROUTINE ENTRY MACRO SUBROUTINE DEFINE	SUM FCS PID SCL SCL2 SCL3 AVG SBS	FCS PID SCL SCL2 SCL3 AVG SBS	** *** *** *** ***	* * * *	/*				/*		*			/*

Appendix

A-2 Condition flag operations

Conversion: *** = same condition flag operation, ** = a part of condition flag operation differs, - = Different condition flag operation, None = no corresponding instruction Condition flags: Left of "/" = Operation of CQM1H. Right of "/" = Operation of CJ1M/CJ1G/CJ2M No "/" = Same operation in CQM1H and CJ *= ON/OFF depending on the instruction statuus

*= ON/OFF depe	ending on th				_									
		CJ1M/CJ ²									ot have this se			
Instructions	CQM1H	/CJ2M	Conversion	ER	GT(>)	GE	EQ(=)	NE (CJ)	LT(<)	LE(CJ)	CY	UF	OF	N (CJ)
						(CJ)								
Interrupt control instructions			<u> </u>					_	_					
INTERRUPT CONTROL	INT	MSKS	None	*										
		MSKR												
		CLI												
		DI												
		EL												
INTERVAL TIMER	STIM	MSKS	None	*	1			[[1
		MSKR												
Step instructions														
STEP DEFINE	STEP	STEP	-	/*			1	[[1
STEP START	SNXT	SNXT	-	/*	1			[[1
Basic I/O Unit instructions														
I/O REFRESH	IORF	IORF	-	/*			1	[[1
7-SEGMENT DECODER	SDEC	SDEC	***	*			1	[[1
7-SEGMENT DISPLAY OUTPUT	7SEG	7SEG	-	*/			1	[[1
		[Ver.2.0												
		or later1												
DIGITAL SWITCH	DSW	DSW		*/	1		1	t	t			1		t
		[Ver.2.0												
		or later1												
TEN KEY INPUT	ткү	TKY	-	*/	†		<u> </u>	t	t	t				t
		[Ver.2.0	1	,										
		or later]												
HEXADECIMAL KEY INPUT	НКҮ	HKY	-	*/				<u> </u>	<u> </u>					
	1 11 1	[Ver.2.0	_	'										
		-												
IO COMMAND TRANSMISSION	IOTC	or later]	None	*										ł
Serial communications instructions	1010		NULLE											
PROTOCOL MACRO	PMCR	PMCR	***	*				<u> </u>	<u> </u>					
TRANSMIT	TXD	TXD	***	*				<u> </u>	<u> </u>					
RECEIVE	RXD	RXD	***	*			<u> </u>	<u> </u>	<u> </u>					t
CHANGE SERIAL PORT SETUP	STUP	STUP	***	*			<u> </u>	<u> </u>	<u> </u>					t
Network instructions	0101	0101	1											
NETWORK SEND	SEND	SEND	***	*			<u> </u>	<u> </u>	<u> </u>					t
NETWORK RECEIVE	RECV	RECV	***	*			<u> </u>	<u> </u>	<u> </u>					t
DELIVER COMMAND	CMND	CMND	***	*			<u> </u>	<u> </u>	<u> </u>					t
Display instructions	ONITE	Olline	1											
MESSAGE	MSG	MSG	***	*				<u> </u>	<u> </u>					†
Clock instructions			1											
HOURS TO SECONDS	SEC	SEC	***	*			*							+
SECONDS TO HOURS	HMS	HMS	***	*			*	<u> </u>	<u> </u>					<u> </u>
Debugging instructions	1 11010	1 11/10	1											
TRACE MEMORY SAMPLE	TRSM	TRSM	***											+
Failure diagnosis instructions	TROM	TROM	1											
FAILURE ALARM AND RESET	FAL	FAL	+ <u>-</u>	/*			<u> </u>	<u> </u>	<u> </u>					t
SEVERE FAILURE ALARM	FALS	FALS	+ <u>-</u>				<u> </u>	<u> </u>	<u> </u>					t
FAILURE POINT DETECT	FPD	FPD	***	/* *	t		<u> </u>	t	t	t	*	 		t
Other instructions			1		1		1	1	1		1	1		
SET CARRY	STC	STC	***		<u> </u>		<u> </u>	<u> </u>	<u> </u>	<u> </u>	ON	 		t/
CLEAR CARRY	CLC	CLC	***		<u> </u>		<u> </u>	<u> </u>	<u> </u>	<u> </u>	OFF	 		t/
High-speed counter/pulse output instructions		510			<u> </u>							<u> </u>		
MODE CONTROL	INI	INI	***	*	t		<u> </u>	<u>+</u>	<u>+</u>	<u> </u>		 		t
HIGH-SPEED COUNTER PV READ	PRV	PRV	***	*	1		<u> </u>	<u> </u>	<u> </u>	t	ON/OFF	†		t
	1 1. V	1 1. V									depending on			
	1		1								instruction			
	1		1								operation			
<u> </u>	<u> </u>	L	L	L	L		L	L	L	l	(CJ2M only)	L		
COMPARISON TABLE LOAD	CTBL	CTBL	***	*	1		1	r	r	[[1		[]
SET PULSES	PULS	PULS	***	*	1		1	r	r	[[1		[]
SPEED OUTPUT	SPED	SPED	***	*	1		1	t	t			1		t
ACCELERATION CONTROL	ACC	ACC	***	*	1		1	t	t			1		t
PULSE OUTPUT	PLS2	PLS2	***	*	1		1	l	l			1		1
PULSE WITH VARIABLE DUTY FACTO		PWM	***	*	1		1	r	r	[[1		T
								4						

МЕМО

МЕМО

Note: Do not use this document to operate the Unit.

OMRON Corporation Industrial Automation Company Tokyo, JAPAN

Contact: www.ia.omron.com

Regional Headquarters OMRON EUROPE B.V. Wegalaan 67-69, 2132 JD Hoofddorp The Netherlands Tel: (31)2356-81-300/Fax: (31)2356-81-388

OMRON ASIA PACIFIC PTE. LTD. No. 438A Alexandra Road # 05-05/08 (Lobby 2), Alexandra Technopark, Singapore 119967 Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON ELECTRONICS LLC 2895 Greenspoint Parkway, Suite 200 Hoffman Estates, IL 60169 U.S.A Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

OMRON (CHINA) CO., LTD. Room 2211, Bank of China Tower, 200 Yin Cheng Zhong Road, PuDong New Area, Shanghai, 200120, China Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:

© OMRON Corporation 2015 All Rights Reserved. In the interest of product improvement, specifications are subject to change without notice.

Cat. No. P068-E1-02

0415(1210)