To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

μPD1720 INSTRUCTION SET

μPD1720 Instruction Set Table

b_{13} b_{12} b_{11} b_{10} \ b_{15} b_{14}		00	01	10	11
		0	1	2	3
0 0 0 0	0	NOP IFCW w IFC t	KIN M KI M	———	ST M, r
0 0 0 1	1	SPB P, N SS N_1 BANK1 STC	ORI M, I	———	MVRS M, r
0 0 1 0	2	———	MVI M, I	OUT P, r	IN r, P
0 0 1 1	3	RPB P, N RS N_1 BANK0 RSC	ANI M, I	CKSTP HALT h	MVRD r, M
0 1 0 0	4	RT	AI M, I	MVSR M_1, M_2	AD r, M
0 1 0 1	5	RTS	SI M, I	EXL r, M	SU r, M
0 1 1 0	6	JMP ADDR (page 0)	AIC M, I	LD r, M	AC r, M
0 1 1 1	7	CAL ADDR (page 0)	SIB M, I	LCDD M, D	SB r, M
1 0 0 0	8	SBK0 TPF P, N TSF N_2 TCEF	AIN M, I	———	ADN r, M
1 0 0 1	9	SBK1 TPT P, N TST N_2 TCET	SIN M, I	TTM TIP TGC	SUN r, M
1 0 1 0	A	TMF M, N	AICN M, I	TUL	ACN r, M
1 0 1 1	B	TMT M, N	SIBN M, I	PLL M, r	SBN r, M
1 1 0 0	C	SLTI M, I	AIS M, I	SLT r, M	ADS r, M
1 1 0 1	D	SGEI M, I	SIS M, I	SGE r, M	SUS r, M
1 1 1 0	E	SEQI M, I	AICS M, I	SEQ r, M	ACS r, M
1 1 1 1	F	SNEI M, I	SIBS M, I	SNE r, M	SBS r, M

List of μPD75206 Instructions

NOTE : D_H : Data memory address high (row address) (2 bits)
D_L : Data memory address low (column address) (4 bits)
Rn : Register number (4 bits)
I : Immediate data (4 bits)
N : Bit position (4 bits)
ADDR : Program memory address (10 bits)
—— : All "1"
r : General register
One of addresses 00−0FH of BANK0
M : Data memory address
One of 00−3FH of BANK0 and 00−1FH of BANK1

P : Port, $0 \leq P \leq 2$
N_1 : Bit position of status word 1 $0 \leq N_1 \leq 7$
N_2 : Bit position of status word 2 $0 \leq N_2 \leq 7$
( ) : Contents of register or memory
c : Carry
b : Borrow
( )n : Contents on bit n of register or memory
w : Data to IF Control Word $0 \leq w \leq 0FH$
t : Trigger conditions $0 \leq t \leq 3$
h : Halt release conditions $0 \leq h \leq 7$

	Mnemonic	Operand		Function	Operation	Machine code			
		1ST	2ND			Operation code			
Addition	AD	r	M	Add memory to register	r ← (r) + (M)	1 1 0 1 0 0	D_H	D_L	Rn
	ADS	r	M	Add memory to register, then skip if carry	r ← (r) + (M) skip if carry	1 1 1 1 0 0	D_H	D_L	Rn
	ADN	r	M	Add memory to register, then skip if not carry	r ← (r) + (M) skip if not carry	1 1 1 0 0 0	D_H	D_L	Rn
	AC	r	M	Add memory to register with carry	r ← (r) + (M) + c	1 1 0 1 1 0	D_H	D_L	Rn
	ACS	r	M	Add memory to register with carry, then skip if carry	r ← (r) + (M) + c skip if carry	1 1 1 1 1 0	D_H	D_L	Rn
	ACN	r	M	Add memory to register with carry, then skip if not carry	r ← (r) + (M) + c skip if not carry	1 1 1 0 1 0	D_H	D_L	Rn
	AI	M	I	Add immediate data to memory	M ← (M) + I	0 1 0 1 0 0	D_H	D_L	I
	AIS	M	I	Add immediate data to memory, then skip if carry	M ← (M) + I skip if carry	0 1 1 1 0 0	D_H	D_L	I
	AIN	M	I	Add immediate data to memory, then skip if not carry	M ← (M) + I skip if not carry	0 1 1 0 0 0	D_H	D_L	I
	AIC	M	I	Add immediate data to memory with carry	M ← (M) + I + c	0 1 0 1 1 0	D_H	D_L	I
	AICS	M	I	Add immediate data to memory with carry, then skip if carry	M ← (M) + I + c skip if carry	0 1 1 1 1 0	D_H	D_L	I
	AICN	M	I	Add immediate data to memory with carry, then skip if not carry	M ← (M) + I + c skip if not carry	0 1 1 0 1 0	D_H	D_L	I
Subtraction	SU	r	M	Subtract memory from register	r ← (r) − (M)	1 1 0 1 0 1	D_H	D_L	Rn
	SUS	r	M	Subtract memory from register, then skip if borrow	r ← (r) − (M) skip if borrow	1 1 1 1 0 1	D_H	D_L	Rn
	SUN	r	M	Subtract memory from register, then skip if not borrow	r ← (r) − (M) skip if not borrow	1 1 1 0 0 1	D_H	D_L	Rn
	SB	r	M	Subtract memory from register with borrow	r ← (r) − (M) − b	1 1 0 1 1 1	D_H	D_L	Rn
	SBS	r	M	Subtract memory from register with borrow, then skip if borrow	r ← (r) − (M) − b skip if borrow	1 1 1 1 1 1	D_H	D_L	Rn
	SBN	r	M	Subtract memory from register with borrow, skip if not borrow	r ← (r) − (M) − b skip if not borrow	1 1 1 0 1 1	D_H	D_L	Rn
	SI	M	I	Subtract immediate data from memory	M ← (M) − I	0 1 0 1 0 1	D_H	D_L	I
	SIS	M	I	Subtract immediate data from memory, then skip if borrow	M ← (M) − I skip if borrow	0 1 1 1 0 1	D_H	D_L	I
	SIN	M	I	Subtract immediate data from memory, then skip if not borrow	M ← (M) − I skip if not borrow	0 1 1 0 0 1	D_H	D_L	I
	SIB	M	I	Subtract immediate data from memory with borrow	M ← (M) − I − b	0 1 0 1 1 1	D_H	D_L	I
	SIBS	M	I	Subtract immediate data from memory with borrow, then skip if borrow	M ← (M) − I − b skip if borrow	0 1 1 1 1 1	D_H	D_L	I
	SIBN	M	I	Subtract immediate data from memory with borrow, then skip if not borrow	M ← (M) − I − b skip if not borrow	0 1 1 0 1 1	D_H	D_L	I

	Mnemonic	Operand 1ST	Operand 2ND	Function	Operation	Machine code: Operation code			
Comparison	SEQ	r	M	Skip if register equals memory	r − M skip if zero	1 0 1 1 1 0	D_H	D_L	Rn
	SNE	r	M	Skip if register not equals memory	r − M skip if not zero	1 0 1 1 1 1	D_H	D_L	Rn
	SGE	r	M	Skip if register is greater than or equal to memory	r − M skip if not borrow (r) ≧ (M)	1 0 1 1 0 1	D_H	D_L	Rn
	SLT	r	M	Skip if register is less than memory	r − M skip if borrow (r) < (M)	1 0 1 1 0 0	D_H	D_L	Rn
	SEQI	M	I	Skip if memory equals immediate data	M − I skip if zero	0 0 1 1 1 0	D_H	D_L	I
	SNEI	M	I	Skip if memory not equals immediate data	M − I skip if not zero	0 0 1 1 1 1	D_H	D_L	I
	SGEI	M	I	Skip if memory is greater than or equal to immediate data	M − I skip if not borrow (M) ≧ I	0 0 1 1 0 1	D_H	D_L	I
	SLTI	M	I	Skip if memory is less than immediate data	M − I skip if borrow (M) < I	0 0 1 1 0 0	D_H	D_L	I
Logical operation	ANI	M	I	Logic AND of memory and immediate data	M ← (M) ∧ I	0 1 0 0 1 1	D_H	D_L	$\bar{I}$
	ORI	M	I	Logic OR of memory and immediate data	M ← (M) ∨ I	0 1 0 0 0 1	D_H	D_L	I
	EXL	r	M	Exclusive OR Logic of memory and register	r ← (r) ⊕ (M)	1 0 0 1 0 1	D_H	D_L	Rn
Transfer	LD	r	M	Load memory to register	r ← (M)	1 0 0 1 1 0	D_H	D_L	Rn
	ST	M	r	Store register to memory	M ← (r)	1 1 0 0 0 0	D_H	D_L	Rn
	MVRD	r	M	Move memory to destination memory referring to register in the same row	[D_H, Rn] ← (M)	1 1 0 0 1 1	D_H	D_L	Rn
	MVRS	M	r	Move source memory referring to register to memory in the same row	M ← [D_H, Rn]	1 1 0 0 0 1	D_H	D_L	Rn
	MVSR	M_1	M_2	Move memory to memory in the same row	[D_H, DL_1] ← [D_H, DL_2]	1 0 0 1 0 0	D_H	D_{L1}	D_{L2}
	MVI	M	I	Move immediate data to memory	M ← I	0 1 0 0 1 0	D_H	D_L	I
	PLL	M	r	Load N0 ~ N3, N_F & memory to PLL registers	PLLR ← (N0 ~ N3), N_F & (M)	1 0 1 0 1 1	D_H	D_L	Rn
Bit test	TMT	M	N	Test memory bits, then skip if all bits specified are true	if M (N) = all "1", then skip	0 0 1 0 1 1	D_H	D_L	N
	TMF	M	N	Test memory bits, then skip if all bits specified are false	if M (N) = all "0", then skip	0 0 1 0 1 0	D_H	D_L	N
Jump	JMP	ADDR		Jump to the address specified	PC ← ADDR	0 0 0 1 1 0	ADDR(10 bits)		
Subroutine	CAL	ADDR		Call subroutine	Stack ← (PC) + 1, PC ← ADDR	0 0 0 1 1 1	ADDR(10 bits)		
	RT			Return to main routine	PC ← (stack)	0 0 0 1 0 0	–	–	–
	RTS			Return to main routine, then skip unconditionary	PC ← (stack), and skip	0 0 0 1 0 1	–	–	–
F/F test	TTM			Test and reset timer F/F, then skip if it has not been set	if Timer F/F = 1, then Timer F/F ← 0 if Timer F/F = 0, then skip	1 0 1 0 0 1	–	–	–
	TUL			Test and reset unlock F/F, then skip if it has not been set	if UL F/F = 1, then UL F/F ← 0 if UL F/F = 0, then skip	1 0 1 0 1 0	–	–	–
Timer test	TIP			Test interval pulse, then skip if low	if IPG = 0, then skip	1 0 1 0 0 1	–	0 0 0 0	0 0 0 0

	Mnemonic	Operand 1ST	Operand 2ND	Function	Operation	Machine code: Operation code			
Status word and terminal test	SS	N_1		Set status word 1	(STATUS WORD 1)$_N$←1	0 0 0 0 0 1	–	0 N_1	–
	RS	N_1		Reset status word 1	(STATUS WORD 1)$_N$←0	0 0 0 0 1 1	–	0 N_1	–
	TST	N_2		Test status word 2 ture	if (STATUS WORD 2)$_N$=all 1, then skip	0 0 1 0 0 1	–	0 N_2	–
	TSF	N_2		Test status word 2 false	if (STATUS WORD 2)$_N$=all 0, then skip	0 0 1 0 0 0	–	0 N_2	–
	STC			Set carry F/F	carry F/F←1	0 0 0 0 0 1	–	0 0 1 0	–
	RSC			Reset carry F/F	carry F/F←0	0 0 0 0 1 1	–	0 0 1 0	–
	BANK0			Select BANK0	BANK F/F←0	0 0 0 0 1 1	–	1 1 0 0	–
	BANK1			Select BANK1	BANK F/F←1	0 0 0 0 0 1	–	0 1 0 0	–
	TCET			Test CE, skip if true	if CE=1, then skip	0 0 1 0 0 1	–	0 0 1 0	–
	TCEF			Test CE, skip if false	if CE=0, then skip	0 0 1 0 0 0	–	0 0 1 0	–
	SBK0			Skip if BANK0	if BANK F/F0=0, then skip	0 0 1 0 0 0	–	1 1 0 0	–
	SBK1			Skip if BANK1	if BANK F/F0=1, then skip	0 0 1 0 0 1	–	0 1 0 0	–
Input / output	LCDD	M	D	Output segment pattern to LCD DIG "D" based on memory, or output memory to LCD DIG "D" directly	LCD(D)←SEGPLA←(M) or LCD(D)←(M)	1 0 0 1 1 1	D_H	D	D_L
	KI	M		Input key data to memory	M←K_{0-3}	0 1 0 0 0 0	D_H	D_L	0 0 0 0
	KIN	M		Input key data to memory, then skip if data are zero	M←K_{0-3}, skip if (M)=0	0 1 0 0 0 0	D_H	D_L	–
	IN	r	P	Input data on port to register	r←(Port (P))	1 1 0 0 1 0	P	–	Rn
	OUT	P	r	Output contents of register to port	(Port(P))←(r)	1 0 0 0 1 0	P	–	Rn
	SPB	P	N	Set port bits	(Port(P))$_N$←1	0 0 0 0 0 1	P	0 0 0 0	N
	RPB	P	N	Reset port bits	(Port(P))$_N$←0	0 0 0 0 1 1	P	0 0 0 0	N
	TPT	P	N	Test port bits, then skip if all bits specified are true	if (Port(P))$_N$=all 1s, then skip	0 0 1 0 0 1	P	0 0 0 0	N
	TPF	P	N	Test port bits, then skip if all bits specified are false	if (Port(P))$_N$=all 0s, then skip	0 0 1 0 0 0	P	0 0 0 0	N
IF counter	IFCW	w		Set immediate data w to IFCW	IFCW←w	0 0 0 0 0 0	1 0	0 0 0 0	w
	IFC	t		Trigger and/or reset IF counter	trigger←t_1, reset←t_0	0 0 0 0 0 0	0 1	0 0 0 0	0 0 t
	TGC			Test IF counter gate, skip if close	if IF gate close, then skip	1 0 1 0 0 1	0 0	–	–
Others	CKSTP			Clock stop by CE	stop clock if CE=0	1 0 0 0 1 1	–	1 1 1 0	1 1 1 0
	HALT	h		Halt the CPU, Restart by condition h	Halt	1 0 0 0 1 1	0 0	–	h 1
	NOP			No operation		0 0 0 0 0 0	–	–	–

IEM-1121
May 1986P
Printed in Japan