

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>.

MESC TECHNICAL NEWS

No. M7700-72-0010

Corrections and Supplementary Explanation for “7900 Series Software Manual” (REV.D)

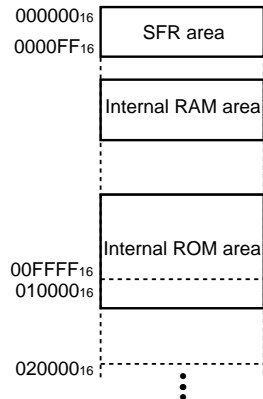
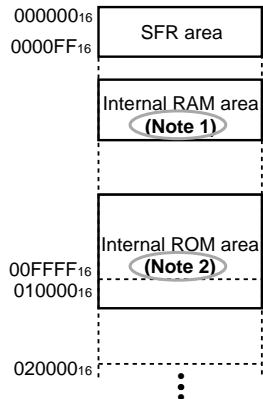
This news includes a few corrections and supplementary explanation for “7900 Series Software Manual” (PDF File, Rev.1.0).

And also, this news includes the information previously announced in MESC TECHNICAL NEWS (No. M7700-57-9911, Corrections and Supplementary Explanation for “7900 Series Software Manual” REV.C).

Note: ☆ represents the new information.

The updated version of the 7900 Series Software Manual (PDF File, Rev. 2.0), which includes the updated information (M7700-72-0010), will be released on the above “Mitsubishi MCU Technical Information” at the end of Dec., 2000.

Corrections and Supplementary Explanation for “7900 Series Software Manual” (REV.D) NO.1

Page	Error	Correction
P2-7 Line 4 below Figure 2.1.4	This register is cleared to “0000 ₁₆ ” at reset.	At reset, DPR0 becomes “0000 ₁₆ ”, and each of DPR1 to DPR3 becomes undefined.
P2-8 (3) Bit 2 Line 4	Use the SEI or SEP instruction to set this flag to “1,” and use the CLI or CLP instruction to clear it to “0.”	Be sure to use the SEI instruction to set this flag to “1”; and be sure to use the CLI or CLP instruction to clear this flag to “0.”
P2-10 Figure 2.2.1	 <p>Note: Memory assignment of internal RAM area varies according to the type of microcomputer. Refer to the latest catalogues or datasheets.</p>	 <p>Notes 1: When the internal RAM area is followed by an unused area or an external area, do not assign a program to the last 8 bytes of the internal RAM area.</p> <p>2: Do not assign a program to the last 8 bytes of the internal ROM area.</p> <p>3: The memory assignment of the internal area varies according to the product type. Refer to the latest datasheets or catalogs.</p>
P2-16 Line 8	<ul style="list-style-type: none"> Use direct page register 0 (DPR0) only. In this case, specify the offset from DPR0 in length of 8 bits. Use direct page registers 0 through 3 (DPR0 through 3). In this case, use the high-order 2 bits of the operand (8 bits) to specify the direct page register and the low-order 6 bits to specify the offset. 	<ul style="list-style-type: none"> <u>When the direct page register select bit = “0”</u> Use direct page register 0 (DPR0) only. In this case, specify the offset from DPR0 in length of 8 bits. <u>When the direct page register select bit = “1”</u> Use direct page registers 0 through 3 (DPR0 through 3). In this case, use the high-order 2 bits of the operand (8 bits) to specify the direct page register and the low-order 6 bits to specify the offset.
P2-16 Figure	<Direct addressing mode>	<When the direct page register select bit = “0”>
P2-18 Figure	<Extension direct addressing mode>	<When the direct page register select bit = “1”>

Corrections and Supplementary Explanation for “7900 Series Software Manual” (REV.D) NO.2

Page	Error	Correction								
<p>P2-19 P2-22 P2-23 P2-25 P2-28 P2-31 P2-33</p> <p>Between the description of Function and Figure in the above pages</p>		<ul style="list-style-type: none"> When the direct page register select bit = “0” Use direct page register 0 (DPR0) only. In this case, specify the offset from DPR0 in length of 8 bits. When the direct page register select bit = “1” Use direct page registers 0 through 3 (DPR0 through 3). In this case, use the high-order 2 bits of the operand (8 bits) to specify the direct page register and the low-order 6 bits to specify the offset. <p><When the direct page register select bit = “0”></p>								
<p>P2-48 Below the Figure</p>		<p>Note: When using the absolute long indexed X addressing mode, make sure that the last code of the instruction does not become the last address of a bank.</p>								
<p>P2-56 Function The last line</p>	<ul style="list-style-type: none"> BBCB and BBSB instructions : in bank 1_{16} is specified. 	<ul style="list-style-type: none"> BBCB and BBSB instructions : in bank 1_{16} is specified. <ul style="list-style-type: none"> When the direct page register select bit = “0” Use direct page register 0 (DPR0) only. In this case, specify the offset from DPR0 in length of 8 bits (the contents of the instruction’s third byte for the BBC and BBS instructions, and the contents of the instruction’s second byte for the BBCB and BBSB instructions). When the direct page register select bit = “1” Use direct page registers 0 through 3 (DPR0 through 3). In this case, use the high-order 2 bits of the operand (8 bits. The contents of the instruction’s third byte for the BBC and BBS instructions, and the contents of the instruction’s second byte for the BBCB and BBSB instructions) to specify the direct page register and the low-order 6 bits to specify the offset. <p><When the direct page register select bit = “0”></p>								
<p>☆ P4-5 Comparison</p>	<table border="1"> <thead> <tr> <th>Instruction</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>*CMPD</td> <td>E – IMM32</td> </tr> </tbody> </table>	Instruction	Description	*CMPD	E – IMM32	<table border="1"> <thead> <tr> <th>Instruction</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>*CMPD</td> <td>E – M32</td> </tr> </tbody> </table>	Instruction	Description	*CMPD	E – M32
Instruction	Description									
*CMPD	E – IMM32									
Instruction	Description									
*CMPD	E – M32									
<p>P4-6 Flag manipulation</p>	<table border="1"> <thead> <tr> <th>Instruction</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>SEP</td> <td>PSL (bit n) ← 1 (n = 0 to 7. Multiple operations can be specified.)</td> </tr> </tbody> </table>	Instruction	Description	SEP	PSL (bit n) ← 1 (n = 0 to 7. Multiple operations can be specified.)	<table border="1"> <thead> <tr> <th>Instruction</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>SEP</td> <td>PSL (bit n) ← 1 (n = 0, 1, 3 to 7. Multiple operations can be specified.)</td> </tr> </tbody> </table>	Instruction	Description	SEP	PSL (bit n) ← 1 (n = 0, 1, 3 to 7. Multiple operations can be specified.)
Instruction	Description									
SEP	PSL (bit n) ← 1 (n = 0 to 7. Multiple operations can be specified.)									
Instruction	Description									
SEP	PSL (bit n) ← 1 (n = 0, 1, 3 to 7. Multiple operations can be specified.)									

Corrections and Supplementary Explanation for “7900 Series Software Manual” (REV.D) NO.3

Page	Error	Correction																														
P4-6 Unconditional branch P4-7 Subroutine	<table border="1"> <thead> <tr> <th>Category</th> <th>Instruction</th> <th>Operation</th> </tr> </thead> <tbody> <tr> <td rowspan="2">Conditional branch</td> <td>JMP</td> <td>PC ← Destination address PC ← mml</td> </tr> <tr> <td>JMPL</td> <td>PG, PC ← Destination address PC ← mml PG ← hh</td> </tr> <tr> <td rowspan="3">Subroutine</td> <td>* BSR</td> <td>Stack ← PC PC ← PC + 2 + REL</td> </tr> <tr> <td>JSR</td> <td>Stack ← PC PC ← Destination address PC ← PC + 3 M(S, S - 1) ← PC S ← S - 2 PC ← mml</td> </tr> <tr> <td>JSRL</td> <td>Stack ← PG, PC PG, PC ← Destination address PC ← PC + 4 M(S, S - 2) ← PG, PC S ← S - 3 PC ← mml PG ← hh</td> </tr> </tbody> </table>	Category	Instruction	Operation	Conditional branch	JMP	PC ← Destination address PC ← mml	JMPL	PG, PC ← Destination address PC ← mml PG ← hh	Subroutine	* BSR	Stack ← PC PC ← PC + 2 + REL	JSR	Stack ← PC PC ← Destination address PC ← PC + 3 M(S, S - 1) ← PC S ← S - 2 PC ← mml	JSRL	Stack ← PG, PC PG, PC ← Destination address PC ← PC + 4 M(S, S - 2) ← PG, PC S ← S - 3 PC ← mml PG ← hh	<table border="1"> <thead> <tr> <th>Category</th> <th>Instruction</th> <th>Operation</th> </tr> </thead> <tbody> <tr> <td rowspan="2">Uncondition- al branch</td> <td>JMP</td> <td>PC ← Destination address</td> </tr> <tr> <td>JMPL</td> <td>PG, PC ← Destination address</td> </tr> <tr> <td rowspan="3">Subroutine</td> <td>* BSR</td> <td>Stack ← PC + 2 PC ← PC + 2 + REL</td> </tr> <tr> <td>JSR</td> <td>Stack ← PC PC ← Destination address</td> </tr> <tr> <td>JSRL</td> <td>Stack ← PG, PC PG, PC ← Destination address</td> </tr> </tbody> </table>	Category	Instruction	Operation	Uncondition- al branch	JMP	PC ← Destination address	JMPL	PG, PC ← Destination address	Subroutine	* BSR	Stack ← PC + 2 PC ← PC + 2 + REL	JSR	Stack ← PC PC ← Destination address	JSRL	Stack ← PG, PC PG, PC ← Destination address
Category	Instruction	Operation																														
Conditional branch	JMP	PC ← Destination address PC ← mml																														
	JMPL	PG, PC ← Destination address PC ← mml PG ← hh																														
Subroutine	* BSR	Stack ← PC PC ← PC + 2 + REL																														
	JSR	Stack ← PC PC ← Destination address PC ← PC + 3 M(S, S - 1) ← PC S ← S - 2 PC ← mml																														
	JSRL	Stack ← PG, PC PG, PC ← Destination address PC ← PC + 4 M(S, S - 2) ← PG, PC S ← S - 3 PC ← mml PG ← hh																														
Category	Instruction	Operation																														
Uncondition- al branch	JMP	PC ← Destination address																														
	JMPL	PG, PC ← Destination address																														
Subroutine	* BSR	Stack ← PC + 2 PC ← PC + 2 + REL																														
	JSR	Stack ← PC PC ← Destination address																														
	JSRL	Stack ← PG, PC PG, PC ← Destination address																														
P4-12 Contents of M (oprd)	<table border="1"> <thead> <tr> <th>Symbol</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>M(oprd)</td> <td>Contents of memory location specified by operand</td> </tr> </tbody> </table>	Symbol	Description	M(oprd)	Contents of memory location specified by operand	<table border="1"> <thead> <tr> <th>Symbol</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>M(oprd)</td> <td>Contents of memory location specified by operand in (), etc.</td> </tr> </tbody> </table>	Symbol	Description	M(oprd)	Contents of memory location specified by operand in (), etc.																						
Symbol	Description																															
M(oprd)	Contents of memory location specified by operand																															
Symbol	Description																															
M(oprd)	Contents of memory location specified by operand in (), etc.																															
P4-60 BSR Operation	Stack ← PC PC ← PC + 2 + REL	Stack ← PC ± 2 PC ← PC + 2 + REL																														
P4-82 CMPD	<p>Operation : E - IMM32</p> <p style="text-align: center;">E - IMM32</p> <p>Description : Subtracts the <u>immediate value</u> from the contents of E in 32-bit length. The result is not stored anywhere.</p>	<p>Operation : E - M32</p> <p style="text-align: center;">E - M32</p> <p>Description : Subtracts the <u>contents of memory</u> from the contents of E in 32-bit length. The result is not stored anywhere.</p>																														
P4-94 DIVS Operation	<p><u>When m = “0”</u> (Figure) * “S” represents MSB of data.</p> <p><u>When m = “1”</u> (Figure) * “S” represents MSB of data.</p>	<p><u>When m = “0”</u> (Figure) * “S” <u>is a sign bit and</u> represents MSB of data.</p> <p><u>When m = “1”</u> (Figure) * “S” <u>is a sign bit and</u> represents MSB of data.</p>																														

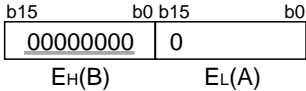
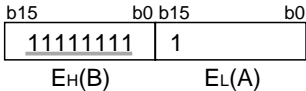
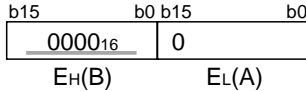

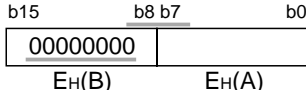

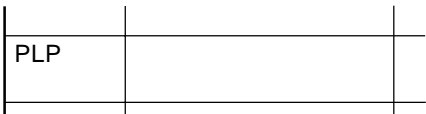
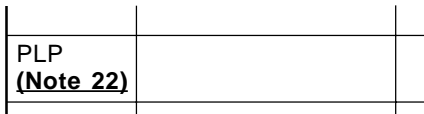
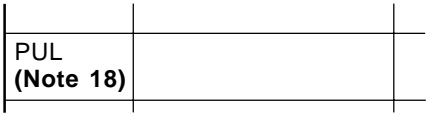
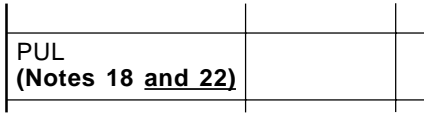
Corrections and Supplementary Explanation for "7900 Series Software Manual" (REV.D) NO.4

Page	Error	Correction
P4-111 JMP/JMPL Operation	<ul style="list-style-type: none"> JMP instruction PC ← Specified address PC ← mml JMPL instruction PG, PC ← Specified address PC ← mml PG ← hh 	<ul style="list-style-type: none"> JMP instruction PC ← Specified address PC ← mml*¹ *¹: In the case of (ABS), M(mml + 1, mml). In the case of (ABS, X), M(mml + X + 1, mml + X). JMPL instruction PG, PC ← Specified address PC ← mml*² PG ← hh*³ *²: In the case of L(ABS), M(mml + 1, mml). *³: In the case of L(ABS), M(mml + 2).
P4-112 JSR/JSRL Operation	<ul style="list-style-type: none"> JSR instruction Stack ← PC Stack ← Specified address PC ← PC + 3 M(S, S - 1) ← PC S ← S - 2 PC ← mml JSRL instruction Stack ← PG, PC PG, PC ← Specified address PC ← PC + 4 M(S to S - 2) ← PG, PC S ← S - 3 PC ← mml PG ← hh 	<ul style="list-style-type: none"> JSR instruction Stack ← PC + 3 Stack ← Specified address PC ← PC + 3 M(S, S - 1) ← PC S ← S - 2 PC ← mml* *: In the case of (ABS, X), M(mml + X + 1, mml + X). JSRL instruction Stack ← PG, PC + 4 PG, PC ← Specified address PC ← PC + 4 M(S to S - 2) ← PG, PC S ← S - 3 PC ← mml PG ← hh
P4-119 LDX Description example	<p><u>CLM</u></p> <p>LDX.W #IMM16 ; X ← IMM16</p> <p>LDX MEM16 ; X ← MEM16</p> <p><u>SEM</u></p> <p>:</p>	<p><u>CLP</u> x</p> <p>LDX.W #IMM16 ; X ← IMM16</p> <p>LDX MEM16 ; X ← MEM16</p> <p><u>SEP</u> x</p> <p>:</p>
P4-121 LDY Description example	<p><u>CLM</u></p> <p>LDY.W #IMM16 ; X ← IMM16</p> <p>LDY MEM16 ; X ← MEM16</p> <p><u>SEM</u></p> <p>:</p>	<p><u>CLP</u> x</p> <p>LDY.W #IMM16 ; X ← IMM16</p> <p>LDY MEM16 ; X ← MEM16</p> <p><u>SEP</u> x</p> <p>:</p>
P4-132 MPYS Operation	<p><u>When m = "0"</u> (Figure) * "S" represents MSB of data.</p> <p><u>When m = "1"</u> (Figure) * "S" represents MSB of data.</p>	<p><u>When m = "0"</u> (Figure) * "S" is a sign bit and represents MSB of data.</p> <p><u>When m = "1"</u> (Figure) * "S" is a sign bit and represents MSB of data.</p>

Corrections and Supplementary Explanation for “7900 Series Software Manual” (REV.D) NO.5

Page	Error	Correction																																				
P4-145 PEI Description	Pushes the contents of the address specified by the sum of the contents of the DPRn and the offset value onto the stack in 16-bit length. ● This instruction is unaffected by flag m.	Pushes the contents of the address specified by the sum of the contents of the DPRn and the offset value onto the stack in 16-bit length. ● When the direct page register select bit = “0” The contents of DPR0 are added to the value of operand dd (8 bits), and the result of addition is pushed onto the stack. ● When the direct page register select bit = “1” In this case, the direct page register DPRn (n = 0 to 3) is specified by the high-order 2 bits of operand dd, and the contents of the specified DPRn are added to the low-order 6 bits of operand dd (8 bits). Then, the result of this addition is pushed onto the stack. ● This instruction is unaffected by flag m.																																				
P4-145 PEI Description example	; (S) ← (DPR0 + <u>dd</u> + 1) ; (S - 1) ← (DPR0 + <u>dd</u>)	; (S) ← (DPR0 + <u>offset</u> + 1) ; (S - 1) ← (DPR0 + <u>offset</u>)																																				
P4-161 PLP, P4-167 PUL, Description Last line		● Be sure to keep flag I = “1” when executing this instruction. Also, be sure to use the SEI instruction when setting flag I to “1.”																																				
P4-167 PUL Status flags	<table border="1"> <thead> <tr> <th>Addressing mode</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>STK</td> <td>3 X <u>i</u> + 13</td> </tr> </tbody> </table> <p>Note i : Number of registers to be restored.</p>	Addressing mode	Cycles	STK	3 X <u>i</u> + 13	<table border="1"> <thead> <tr> <th>Addressing mode</th> <th>Cycles</th> </tr> </thead> <tbody> <tr> <td>STK</td> <td>3 X <u>i₁</u> + 13</td> </tr> </tbody> </table> <p>Note: <u>i₁</u> = number of registers to be restored.</p>	Addressing mode	Cycles	STK	3 X <u>i₁</u> + 13																												
Addressing mode	Cycles																																					
STK	3 X <u>i</u> + 13																																					
Addressing mode	Cycles																																					
STK	3 X <u>i₁</u> + 13																																					
P4-171 ROL, P4-174 ROR	<table border="1"> <thead> <tr> <th colspan="2">Syntax</th> </tr> </thead> <tbody> <tr> <td>ROL/ROR A</td> <td></td> </tr> <tr> <td>ROL/ROR B</td> <td></td> </tr> <tr> <td>ROL/ROR <u>A</u>, dd</td> <td></td> </tr> <tr> <td>ROL/ROR <u>A</u>, dd, X</td> <td></td> </tr> <tr> <td>ROL/ROR <u>A</u>, mml</td> <td></td> </tr> <tr> <td>ROL/ROR <u>A</u>, mml, X</td> <td></td> </tr> </tbody> </table>	Syntax		ROL/ROR A		ROL/ROR B		ROL/ROR <u>A</u> , dd		ROL/ROR <u>A</u> , dd, X		ROL/ROR <u>A</u> , mml		ROL/ROR <u>A</u> , mml, X		<table border="1"> <thead> <tr> <th colspan="2">Syntax</th> </tr> </thead> <tbody> <tr> <td>ROL/ROR A</td> <td></td> </tr> <tr> <td>ROL/ROR B</td> <td></td> </tr> <tr> <td>ROL/ROR dd</td> <td></td> </tr> <tr> <td>ROL/ROR dd, X</td> <td></td> </tr> <tr> <td>ROL/ROR mml</td> <td></td> </tr> <tr> <td>ROL/ROR mml, X</td> <td></td> </tr> </tbody> </table>	Syntax		ROL/ROR A		ROL/ROR B		ROL/ROR dd		ROL/ROR dd, X		ROL/ROR mml		ROL/ROR mml, X									
Syntax																																						
ROL/ROR A																																						
ROL/ROR B																																						
ROL/ROR <u>A</u> , dd																																						
ROL/ROR <u>A</u> , dd, X																																						
ROL/ROR <u>A</u> , mml																																						
ROL/ROR <u>A</u> , mml, X																																						
Syntax																																						
ROL/ROR A																																						
ROL/ROR B																																						
ROL/ROR dd																																						
ROL/ROR dd, X																																						
ROL/ROR mml																																						
ROL/ROR mml, X																																						
P4-189 SEP Operation	PS _L (bit n) ← 1 (n = 0 to 7. Multiple bits can be specified.)	PS _L (bit n) ← 1 (n = 0, 1, 3 to 7. Multiple bits can be specified.)																																				
P4-189 SEP Description	Sets the specified flags be specified. ● This instruction is	Sets the specified flags be specified. <u>Do not use this instruction to specify bit 2 of PS_L (flag I). (When setting flag I to “1,” be sure to use the SEI instruction.)</u> ● This instruction is																																				
P4-189 SEP Status flags	<table border="1"> <tbody> <tr> <td>IPL</td> <td>N</td> <td>V</td> <td>m</td> <td>x</td> <td>D</td> <td>I</td> <td>Z</td> <td>C</td> </tr> <tr> <td>—</td> <td>N</td> <td>V</td> <td>m</td> <td>x</td> <td>D</td> <td>I</td> <td>Z</td> <td>C</td> </tr> </tbody> </table>	IPL	N	V	m	x	D	I	Z	C	—	N	V	m	x	D	I	Z	C	<table border="1"> <tbody> <tr> <td>IPL</td> <td>N</td> <td>V</td> <td>m</td> <td>x</td> <td>D</td> <td>I</td> <td>Z</td> <td>C</td> </tr> <tr> <td>—</td> <td>N</td> <td>V</td> <td>m</td> <td>x</td> <td>D</td> <td>(Note)</td> <td>Z</td> <td>C</td> </tr> </tbody> </table> <p>Note: Do not use this instruction to specify flag I.</p>	IPL	N	V	m	x	D	I	Z	C	—	N	V	m	x	D	(Note)	Z	C
IPL	N	V	m	x	D	I	Z	C																														
—	N	V	m	x	D	I	Z	C																														
IPL	N	V	m	x	D	I	Z	C																														
—	N	V	m	x	D	(Note)	Z	C																														

Corrections and Supplementary Explanation for “7900 Series Software Manual” (REV.D) NO.6

Page	Error	Correction
P4-202 SUBMD Description example	SUBMB MEM32, #IMM32 ; MEM32 ← ...	SUBMD MEM32, #IMM32 ; MEM32 ← ...
P4-229 XAB Description example	CLM x XAB ; A ⇔ B SEM x XAB ; A _L ⇔ B _L	CLM XAB ; A ⇔ B SEM XAB ; A _L ⇔ B _L
P5-10 BSR Function	(S) ← PC PC ← PC + 2 + REL (-1024 to +1023)	(S) ← PC + 2 PC ← PC + 2 + REL (-1024 to +1023)
P5-14 CMPD Function	E – IMM32	E – M32
P5-16 EXTSD Function	E ← E _L (= A) (Extension sign) (Bit 15 of A = 0)  (Bit 15 of A = 1) 	E ← E _L (= A) (Extension sign) (Bit 15 of A = 0)  (Bit 15 of A = 1) 
P5-16 EXTZD Function	E ← E _L (= A) (Extension zero) 	E ← E _L (= A) (Extension zero) 
P5-28 PLP		
P5-30 PUL		

Corrections and Supplementary Explanation for “7900 Series Software Manual” (REV.D) NO.7

Page	Error	Correction																																																																																								
P5-31 PUL STK	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="4">Addressing Modes</th> </tr> <tr> <th colspan="4">STK</th> </tr> <tr> <th>op</th> <th>n</th> <th colspan="2">#</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">67</td> <td style="text-align: center;">13 + 3 i</td> <td colspan="2" style="text-align: center;">2</td> </tr> </tbody> </table>	Addressing Modes				STK				op	n	#		67	13 + 3 i	2		<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="4">Addressing Modes</th> </tr> <tr> <th colspan="4">STK</th> </tr> <tr> <th>op</th> <th>n</th> <th colspan="2">#</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">67</td> <td style="text-align: center;">13 + 3 i</td> <td colspan="2" style="text-align: center;">2</td> </tr> </tbody> </table>	Addressing Modes				STK				op	n	#		67	13 + 3 i	2																																																									
Addressing Modes																																																																																										
STK																																																																																										
op	n	#																																																																																								
67	13 + 3 i	2																																																																																								
Addressing Modes																																																																																										
STK																																																																																										
op	n	#																																																																																								
67	13 + 3 i	2																																																																																								
P5-36 SEP	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Symbol</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">SEP</td> <td>PS_L (bit n) ← 1 (n = 0 to 7. Multiple bits can be specified.)</td> </tr> </tbody> </table>	Symbol	Function	SEP	PS _L (bit n) ← 1 (n = 0 to 7. Multiple bits can be specified.)	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Symbol</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">SEP</td> <td>PS_L (bit n) ← 1 (n = 0, 1, 3 to 7. Multiple bits can be specified.)</td> </tr> </tbody> </table>	Symbol	Function	SEP	PS _L (bit n) ← 1 (n = 0, 1, 3 to 7. Multiple bits can be specified.)																																																																																
Symbol	Function																																																																																									
SEP	PS _L (bit n) ← 1 (n = 0 to 7. Multiple bits can be specified.)																																																																																									
Symbol	Function																																																																																									
SEP	PS _L (bit n) ← 1 (n = 0, 1, 3 to 7. Multiple bits can be specified.)																																																																																									
P5-37 SEP	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="11">Processor Status register</th> </tr> <tr> <th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th> </tr> <tr> <th>IPL</th><th>N</th><th>V</th><th>m</th><th>x</th><th>D</th><th>I</th><th>Z</th><th>C</th><th></th><th></th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">•</td><td style="text-align: center;">•</td><td style="text-align: center;">•</td><td colspan="8">Specified flag becomes “1.”</td> </tr> </tbody> </table>	Processor Status register											10	9	8	7	6	5	4	3	2	1	0	IPL	N	V	m	x	D	I	Z	C			•	•	•	Specified flag becomes “1.”								<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="11">Processor Status register</th> </tr> <tr> <th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th> </tr> <tr> <th>IPL</th><th>N</th><th>V</th><th>m</th><th>x</th><th>D</th><th>I</th><th>Z</th><th>C</th><th></th><th></th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">•</td><td style="text-align: center;">•</td><td style="text-align: center;">•</td><td colspan="8">Specified flag becomes “1” (Note 21).</td> </tr> </tbody> </table>	Processor Status register											10	9	8	7	6	5	4	3	2	1	0	IPL	N	V	m	x	D	I	Z	C			•	•	•	Specified flag becomes “1” (Note 21).							
Processor Status register																																																																																										
10	9	8	7	6	5	4	3	2	1	0																																																																																
IPL	N	V	m	x	D	I	Z	C																																																																																		
•	•	•	Specified flag becomes “1.”																																																																																							
Processor Status register																																																																																										
10	9	8	7	6	5	4	3	2	1	0																																																																																
IPL	N	V	m	x	D	I	Z	C																																																																																		
•	•	•	Specified flag becomes “1” (Note 21).																																																																																							
P5-43 Note 18	Note 18. Letter “ <u>i</u> ” indicates the number of registers to be restored.	Note 18. Letter “ <u>i</u> ” indicates the number of registers to be restored.																																																																																								
P5-43 Below Note 20		<p>Note 21. Do not use the SEP instruction to specify flag I. (When setting flag I to “1”, be sure to use the SEI instruction.)</p> <div style="border: 1px solid black; padding: 5px;"> <p>Note 22. Be sure to keep flag I = “1” when executing the PLP or PUL instruction. Also, be sure to use the SEI instruction when setting flag I to “1.”</p> </div>																																																																																								