

SH7000 Series

Block Transfer (4 Bytes Aligned)

Label: MOVE4

Functions Used: MOV.L Instruction
Post-Increment Register Indirect Addressing
Register Indirect Addressing with Displacement

Contents

1. Function	2
2. Arguments.....	2
3. Internal Register Changes and Flag Changes.....	3
4. Programming Specifications	4
5. Notes.....	4
6. Description	5
7. Flowchart.....	9
8. Program Listing	13

1. Function

Transfers a block of data. Note that the start addresses for the block data source and destination areas must be 4n addresses, and the number of bytes to be transferred must be 4n bytes.

2. Arguments

Description		Storage Location	Data Length (Bytes)
Input	Number of transfer bytes (4n bytes)	R0	4
	Start address of transfer data source area (4n address)	R1	4
	Start address of transfer data destination area (4n address)	R2	4
Output	—	—	—

3. Internal Register Changes and Flag Changes

(Before Execution) → (After Execution)	
R0	Number of transfer bytes → Change
R1	Start address of transfer data destination area → Change
R2	Start address of transfer data source area → Change
R3	Work
R4	
R5	
R6	
R7	
R8	
R9	
R10	
R11	
R12	
R13	
R14	
R15	(SP)

T bit

*

 — : No change
* : Change
0 : Fixed 0
1 : Fixed 1

4. Programming Specifications

Program memory (bytes)
142
Data memory (bytes)
0
Stack (bytes)
4
Number of states
114
Reentrant
Yes
Relocation
Yes
Intermediate interrupt
Yes

5. Notes

The number of states indicated in the programming specifications is the value when the number of transfer bytes is 100.

6. Description

(1) Function

Details of the arguments are as follows.

- R0: As the input argument, set the number of transfer bytes (4n bytes). Note that hardware limitations apply.
- R1: As the input argument, set the start address of transfer data destination area (4n address).
- R2: As the input argument, set the start address of transfer data source area (4n address).

Figure 1 shows a software MOVE execution example.

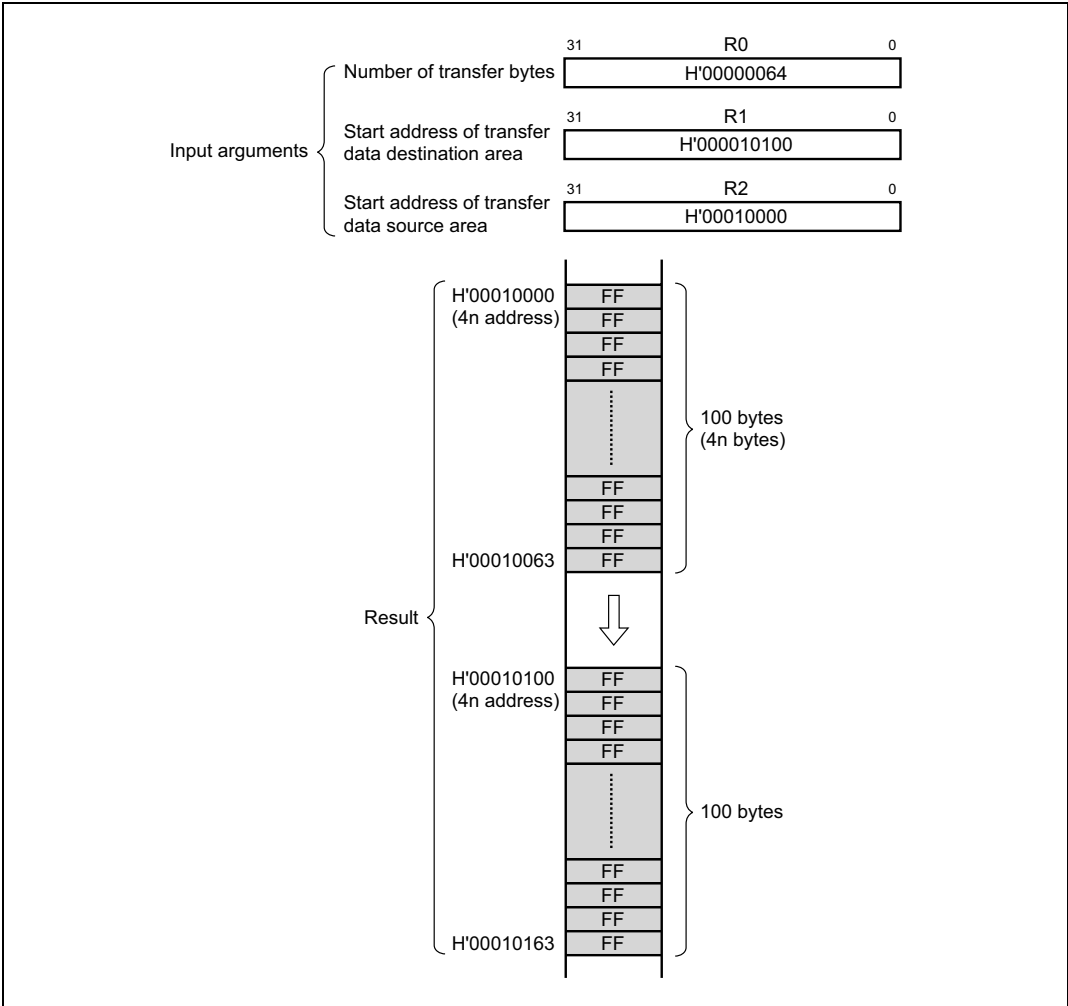


Figure 1 Software MOVE4 Execution Example

(2) Usage Notes

- (a) The input arguments should be set so that the transfer data source area and transfer data destination area do not overlap. If the two areas overlap, as shown in figure 2, the data in the source area will be destroyed.

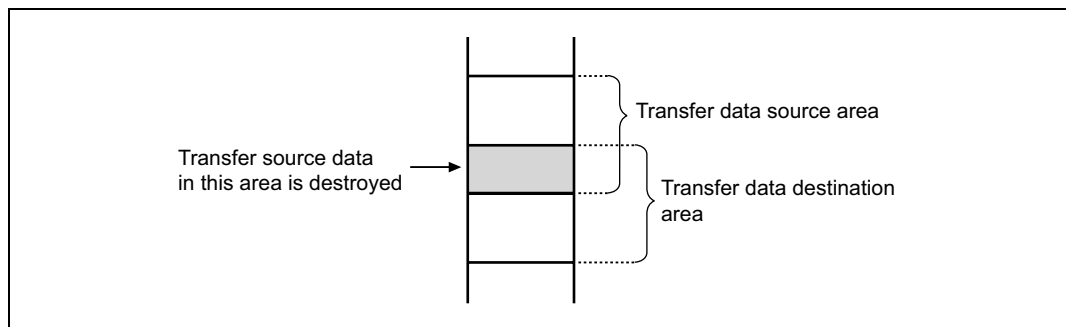


Figure 2 Block Transfer with Overlapping Data

- (b) The contents of R0, R1, and R2, which set the number of transfer bytes, the start address of the transfer data destination area, and the start address of the transfer data source area, are changed using the software MOVE instruction. If the values for the number of transfer bytes, the start address of the transfer data destination area, and the start address of the transfer data source area will be needed after the software MOVE instruction is executed, they should be saved beforehand.

(3) RAM Used

No RAM is used by the software MOVE4 instruction.

(4) Usage Example

After the start address of the transfer data destination area, the start address of the transfer data source area, and the number of transfer bytes have been set in the input arguments, the software MOVE4 instruction is executed by a subroutine call.

```

MOV.L  DATA1,R0    . . . . Sets number of transfer bytes in input argument (R0)
MOV.L  DATA2,R1    . . . . Sets start address of transfer data destination area in input argument (R1)
BSR    MOVE4        . . . . Subroutine call to software MOVE4
MOV.L  DATA3,R2    . . . . Sets start address of transfer data source area in input argument (R2)

    .
    .
    .
.align 4
DATA1 .data.1 H'00000064
DATA2 .data.1 H'00010100
DATA3 .data.1 H'00010000

```

(5) Operating Principle

- (a) Since the transfer source and transfer destination addresses are both 4n addresses (4 bytes aligned), data is transferred from the source to the destination 4 bytes at a time.
- (b) Post-increment register indirect addressing (@R2+) is used to specify the transfer source address, which is then automatically incremented by 4 after each 4 bytes are transferred. Register indirect addressing with displacement is used to specify the transfer destination address. The displacement is 0 to 60, so it is necessary to increment the transfer destination address by 64 after each 60 bytes is transferred. No other increment processing is needed.
- (c) A value equal to the start address of the transfer data source area (R2) plus the number of transfer bytes is set in R3. After the setting is made, R0, which was previously set to the number of data bytes, is used as workspace for the data transfer. After the transfer source data is transferred to R0, it is determined whether or not R2 is less than or equal to R3. If this condition is met ($R2 \leq R3$), the data in R0 is data from the transfer source area and it is transferred to the transfer destination. If the condition is not met ($R2 > R3$), the data in R0 is data from outside the transfer source area and the transfer terminates.

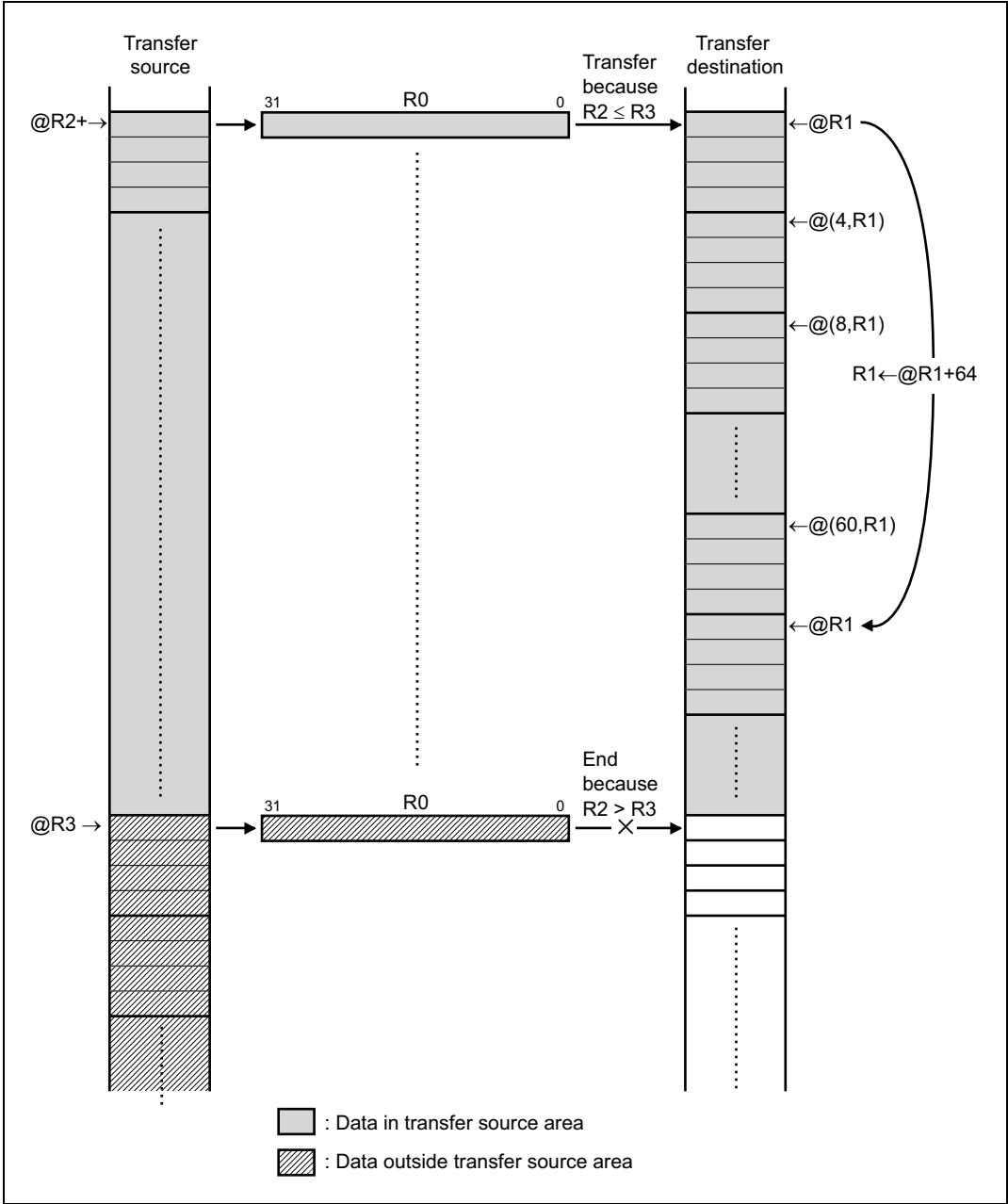
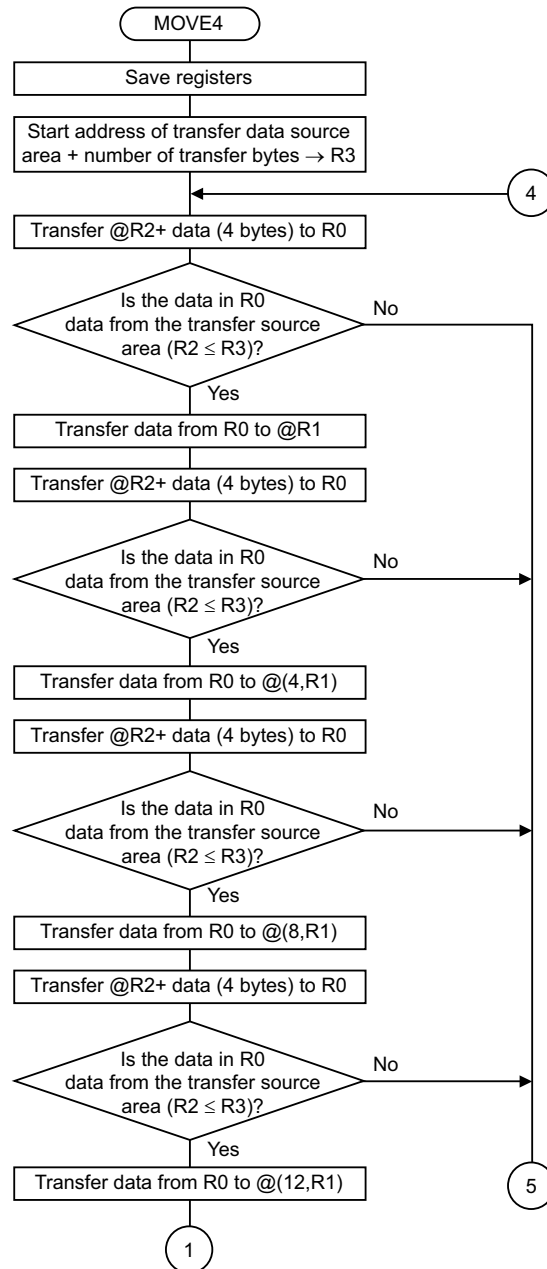
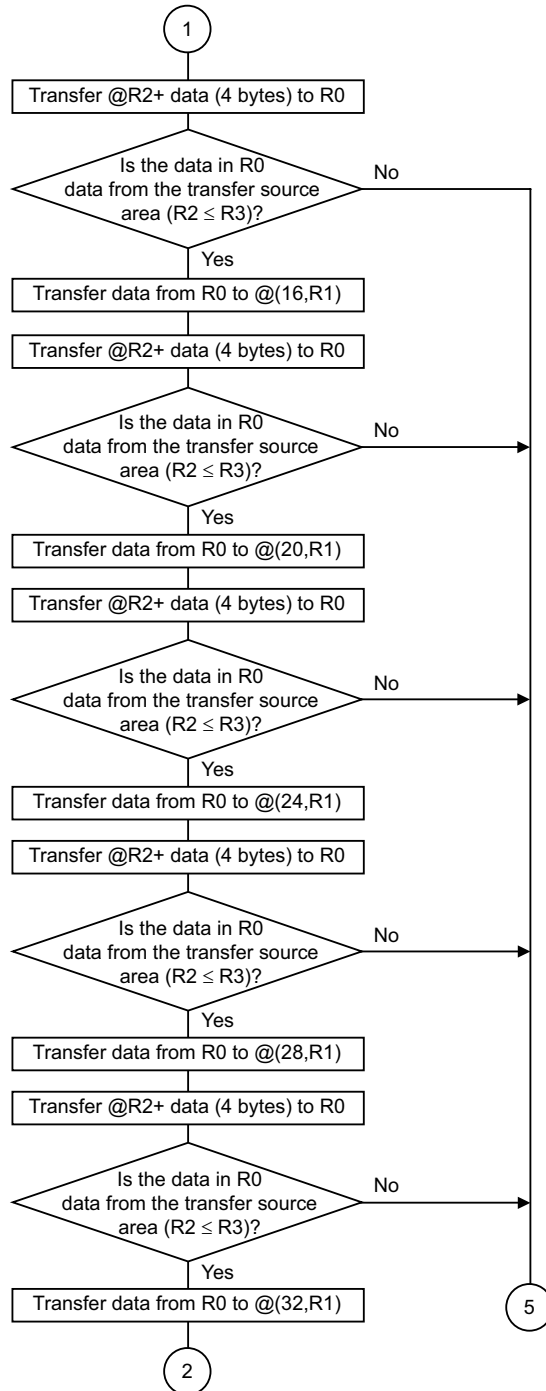
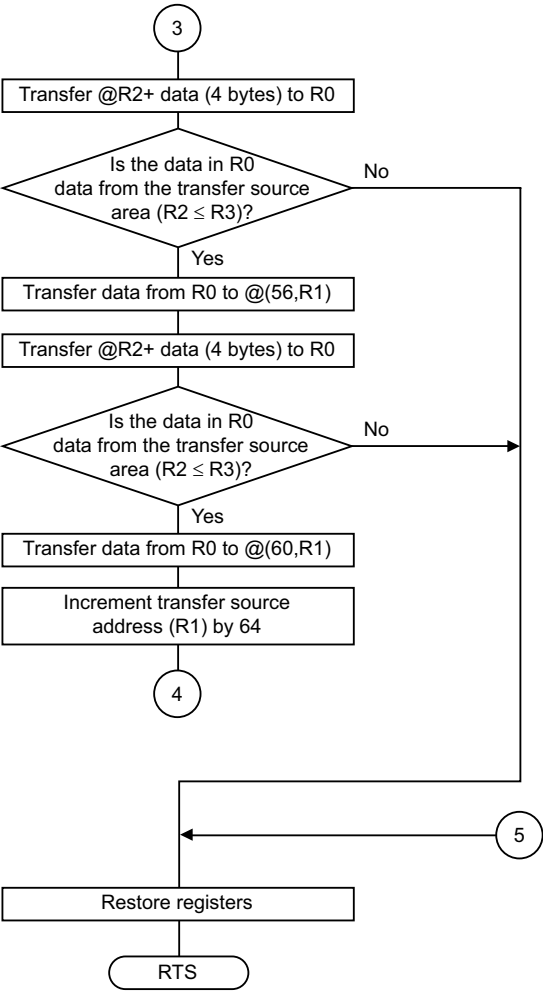


Figure 3 Data Transfer Method

7. Flowchart







8. Program Listing

```

1      1      ;*****
2      2      ;*
3      3      ;*      NAME ; MOVING MEMORY BLOKS (MOVE4)
4      4      ;*
5      5      ;*****
6      6      ;*
7      7      ;*      ENTRY : R0      (NUMBER OF TRANSFER)
8      8      ;*      R1      (DESTINATION ADDRESS)
9      9      ;*      R2      (SOURCE ADDRESS)
10     10     ;*      RETURNS : NOTHING
11     11     ;*
12     12     ;*****
13     13     .SECTION A, CODE, LOCATE=H'1000
14     14     MOVE4 .EQU $      ; Entry point
15     15     MOV.L R3, @-R15    ; Escape register
16     16     MOV R2, R3
17     17     ADD R0, R3
18     18     MOVE41
19     19     MOV.L @R2+, R0      ; Load source data
20     20     CMP/HS R2, R3      ; R2 <= R3 ?
21     21     BF MOVE4_END      ; No
22     22     MOV.L R0, @R1      ; Yes -> Store source data
23     23     MOVE42
24     24     MOV.L @R2+, R0      ; Load source data
25     25     CMP/HS R2, R3      ; R2 <= R3 ?
26     26     BF MOVE4_END      ; No
27     27     MOV.L R0, @(4, R1)  ; Yes -> Store source data
28     28     MOVE43
29     29     MOV.L @R2+, R0      ; Load source data
30     30     CMP/HS R2, R3      ; R2 <= R3 ?
31     31     BF MOVE4_END      ; No
32     32     MOV.L R0, @(8, R1)  ; Yes -> Store source data
33     33     MOVE44
34     34     MOV.L @R2+, R0      ; Load source data
35     35     CMP/HS R2, R3      ; R2 <= R3 ?
36     36     BF MOVE4_END      ; No
37     37     MOV.L R0, @(12, R1) ; Yes -> Store source data
38     38     MOVE45
39     39     MOV.L @R2+, R0      ; Load source data
40     40     CMP/HS R2, R3      ; R2 <= R3 ?
41     41     BF MOVE4_END      ; No
42     42     MOV.L R0, @(16, R1) ; Yes -> Store source data
43     43     MOVE46
44     44     MOV.L @R2+, R0      ; Load source data
45     45     CMP/HS R2, R3      ; R2 <= R3 ?
46     46     BF MOVE4_END      ; No
47     47     MOV.L R0, @(20, R1) ; Yes -> Store source data
48     48     MOVE47
49     49     MOV.L @R2+, R0      ; Load source data

```

50 00001038 3322	50	CMP/HS	R2,R3	; R2 <= R3 ?
51 0000103A 8B26	51	BF	MOVE4_END	; No
52 0000103C 1106	52	MOV.L	R0,@(24,R1)	; Yes -> Store source data
53 0000103E	53	MOVE48		;
54 0000103E 6026	54	MOV.L	@R2+,R0	; Load source data
55 00001040 3322	55	CMP/HS	R2,R3	; R2 <= R3 ?
56 00001042 8B22	56	BF	MOVE4_END	; No
57 00001044 1107	57	MOV.L	R0,@(28,R1)	; Yes -> Store source data
58 00001046	58	MOVE49		;
59 00001046 6026	59	MOV.L	@R2+,R0	; Load source data
60 00001048 3322	60	CMP/HS	R2,R3	; R2 <= R3 ?
61 0000104A 8B1E	61	BF	MOVE4_END	; No
62 0000104C 1108	62	MOV.L	R0,@(32,R1)	; Yes -> Store source data
63 0000104E	63	MOVE410		;
64 0000104E 6026	64	MOV.L	@R2+,R0	; Load source data
65 00001050 3322	65	CMP/HS	R2,R3	; R2 <= R3 ?
66 00001052 8B1A	66	BF	MOVE4_END	; No
67 00001054 1109	67	MOV.L	R0,@(36,R1)	; Yes -> Store source data
68 00001056	68	MOVE411		;
69 00001056 6026	69	MOV.L	@R2+,R0	; Load source data
70 00001058 3322	70	CMP/HS	R2,R3	; R2 <= R3 ?
71 0000105A 8B16	71	BF	MOVE4_END	; No
72 0000105C 110A	72	MOV.L	R0,@(40,R1)	; Yes -> Store source data
73 0000105E	73	MOVE412		;
74 0000105E 6026	74	MOV.L	@R2+,R0	; Load source data
75 00001060 3322	75	CMP/HS	R2,R3	; R2 <= R3 ?
76 00001062 8B12	76	BF	MOVE4_END	; No
77 00001064 110B	77	MOV.L	R0,@(44,R1)	; Yes -> Store source data
78 00001066	78	MOVE413		;
79 00001066 6026	79	MOV.L	@R2+,R0	; Load source data
80 00001068 3322	80	CMP/HS	R2,R3	; R2 <= R3 ?
81 0000106A 8B0E	81	BF	MOVE4_END	; No
82 0000106C 110C	82	MOV.L	R0,@(48,R1)	; Yes -> Store source data
83 0000106E	83	MOVE414		;
84 0000106E 6026	84	MOV.L	@R2+,R0	; Load source data
85 00001070 3322	85	CMP/HS	R2,R3	; R2 <= R3 ?
86 00001072 8B0A	86	BF	MOVE4_END	; No
87 00001074 110D	87	MOV.L	R0,@(52,R1)	; Yes -> Store source data
88 00001076	88	MOVE415		;
89 00001076 6026	89	MOV.L	@R2+,R0	; Load source data
90 00001078 3322	90	CMP/HS	R2,R3	; R2 <= R3 ?
91 0000107A 8B06	91	BF	MOVE4_END	; No
92 0000107C 110E	92	MOV.L	R0,@(56,R1)	; Yes -> Store source data
93 0000107E	93	MOVE416		;
94 0000107E 6026	94	MOV.L	@R2+,R0	; Load source data
95 00001080 3322	95	CMP/HS	R2,R3	; R2 <= R3 ?
96 00001082 8B02	96	BF	MOVE4_END	; No
97 00001084 110F	97	MOV.L	R0,@(60,R1)	; Yes -> Store source data
98	98			;
99 00001086 AFBE	99	BRA	MOVE41	;
100 00001088 7140	100	ADD	#D'64,R1	; R1 <- R1 + 64

```
101 0000108A      101  MOVE_END      ;
102 0000108A 000B 102      RTS      ;
103 0000108C 63F6 103      MOV.L   @R15+,R3 ; Return register
104              104      .END
****TOTAL ERRORS      0
****TOTAL WARNINGS    0
```

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
The information described here may contain technical inaccuracies or typographical errors.
Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.