APPENDIX B SETTING EXAMPLES

B.1 Ports

Shown below is an example of port 1 initialization executed after a power-on reset.

When reset

P1 (undefined)	PM1 (input)	PMC1 (port mode)
7 0	7 0	7 0
××××××××	11111111	00000000

After setup

P1	PM1	PMC1
7 0	7 0	7 0
11000000	00011111	00100000
	(Output) (Input)	Control mode

```
;**********************************************************************************
;***                     Special function register setting                      ***
;**********************************************************************************
;

                                  :
                MOV       P1,     11000000B
                MOV       PM1,    00011111B
                MOV       PMC1,   00100000B        ; TOUT output
                                  :
```

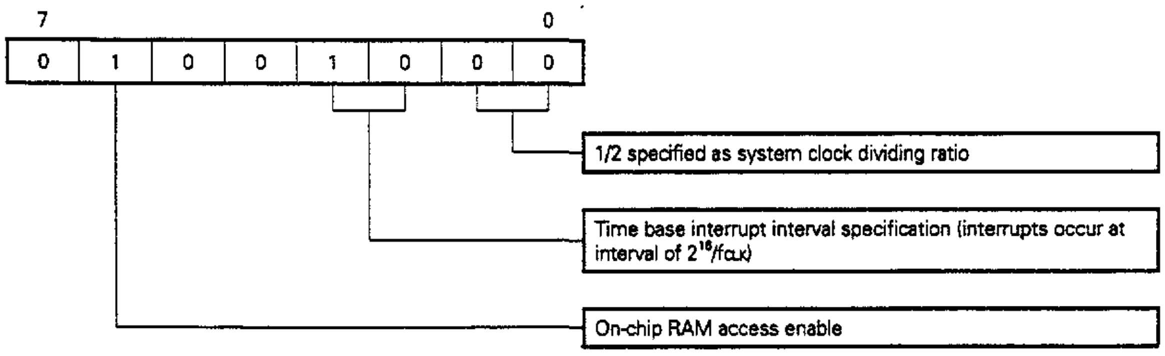
P10 to P14	: Changes when data is input via input port
P15	: TOUT output occurs when in control mode
P16 and P17	: "1" is output from the output port

B.2 Programmable Wait, Processor Control, and Refresh Function

Mode setting method for wait insertion, refresh, etc.

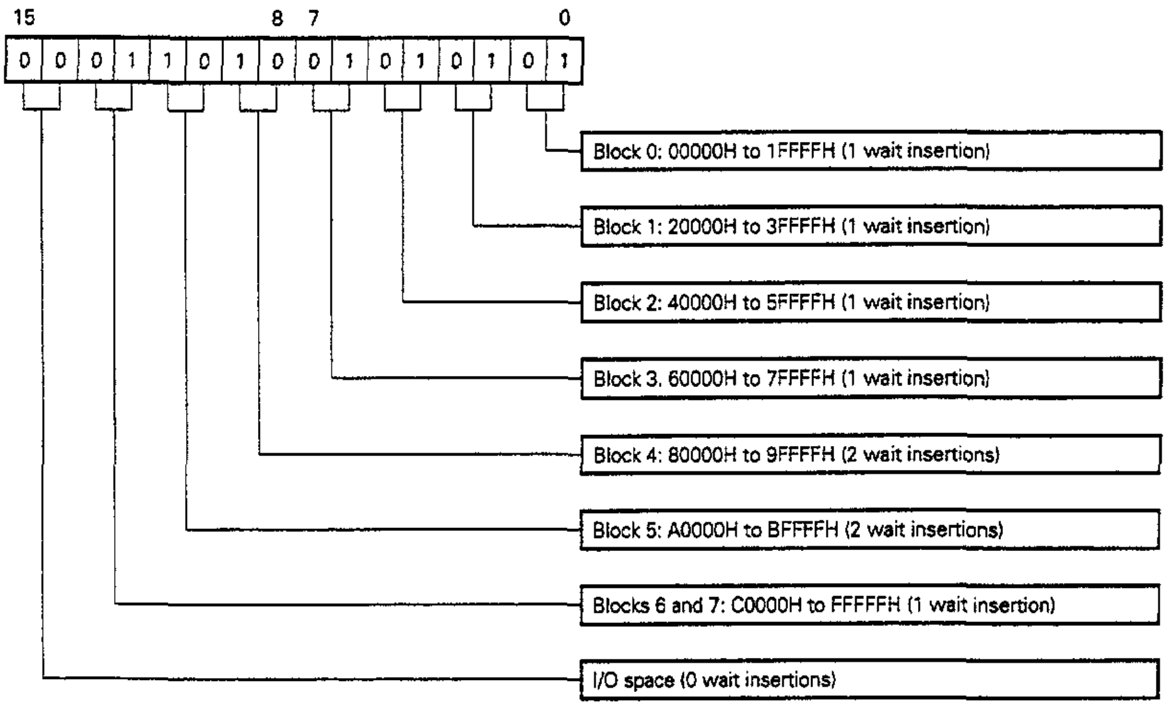
```
;*******************************************************************************************
;***                    Initialization of special function register                     ***
;*******************************************************************************************
;
                              :
          MOV        PRC,     01001000B      ;①
          MOV        WTCH,    00011010B      ;②
          MOV        WTCL,    01010101B      ;②
          MOV        RFM,     11110101B      ;③
                              :
          MOV        STBC,    00000001B      ;④
                              :
          HALT

;*******************************************************************************************
;
                              :
          MOV        RFM,     00000000B      ;⑤
          STOP
                              :
```

① **PRC (Processor control register)**


7
0
0 1 0 0 1 0 0 0
1/2 specified as system clock dividing ratio
Time base interrupt interval specification (interrupts occur at interval of $2^{16}/f_{CLK}$)
On-chip RAM access enable

② WTC (Wait control register)

Separate wait insertions for 8-block memory space and I/O space


15
8 7
0
0 0 0 1 1 0 1 0 0 1 0 1 0 1 0 1
Block 0: 00000H to 1FFFFH (1 wait insertion)
Block 1: 20000H to 3FFFFH (1 wait insertion)
Block 2: 40000H to 5FFFFH (1 wait insertion)
Block 3. 60000H to 7FFFFH (1 wait insertion)
Block 4: 80000H to 9FFFFH (2 wait insertions)
Block 5: A0000H to BFFFFH (2 wait insertions)
Blocks 6 and 7: C0000H to FFFFFH (1 wait insertion)
I/O space (0 wait insertions)

These can be organized as shown below.

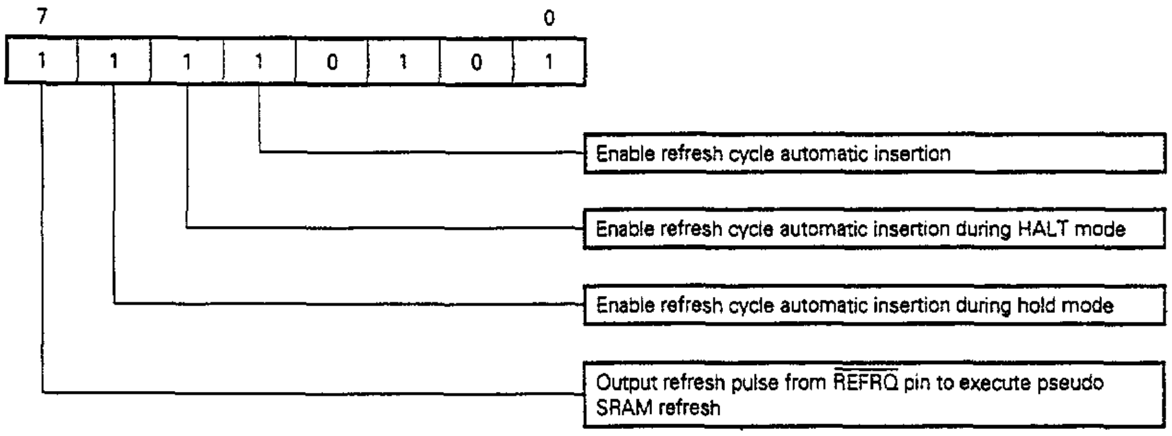
```
MOV    WTC,    1A55H
```

Relation between memory and wait states

Memory location	Capacity	Block(s)	Wait state(s)
00000H-07FFFH	32 Kbytes	Block 0	1 state
40000H-47FFFH	32 Kbytes	Block 2	1 state
80000H-BFFFFH	256 Kbytes	Blocks 4 and 5	2 states
F8000H-FFFFFH	32 Kbytes	Blocks 6 and 7	1 state

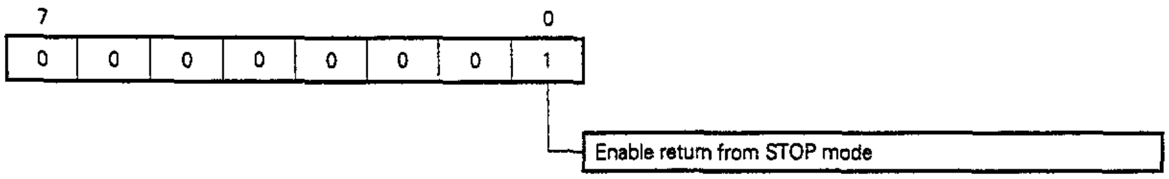
③ **RFM (Refresh mode register)**

Inserts a refresh cycle into a series of bus cycles.


7
0
1 1 1 1 0 1 0 1
Enable refresh cycle automatic insertion
Enable refresh cycle automatic insertion during HALT mode
Enable refresh cycle automatic insertion during hold mode
Output refresh pulse from $\overline{REFRQ}$ pin to execute pseudo SRAM refresh

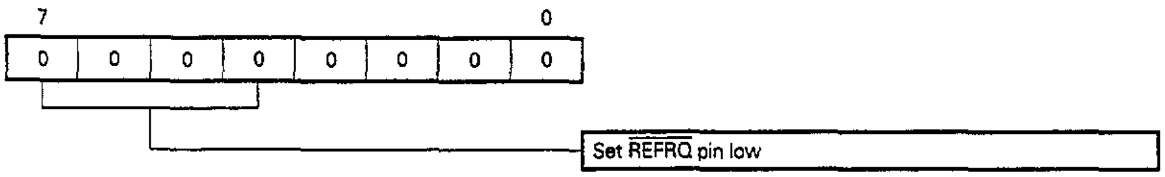
④ **STBC (Standby control register)**

Controls return from STOP mode.



⑤ **RFM (Refresh mode register)**

Sets $\overline{REFRQ}$ pin low to use power-down self-refresh function for pseudo SRAM.



B.3 Register Bank Switching

B.3.1 Register bank switching by interrupt request

In response to INTTU0 (timer unit 0's interrupt), switches from register bank 7 to register bank 6.

```
;*******************************************************************************
;***                Timer interrupt setting (register bank switching)        ***
;*******************************************************************************
;
                              :
        MOV     REGBANK. BK6. BVPC,  OFFSET INTTU0Note ;① Vector PC initialization
        MOV     REGBANK. BK6. BPS,   SEG INTTU0        ;① PS
        MOV     REGBANK. BK6. BDS0, 0                  ;① DS0
        MOV     REGBANK. BK6. BDS1, 0                  ;① DS1
;
        MOV     TMIC0, 00010110B                       ;②
        MOV     MD0,   0FFH                            ;
        MOV     TMC0,  80H                             ;③
;
        EI
                              :
;===============================================================================
;===                    Processing of register bank 6                        ===
;===============================================================================

INTTU0:
        MOVSPA          ;④
                        :
        FINT            ;⑤ Return from register bank
        RETRBI          ;⑤
```

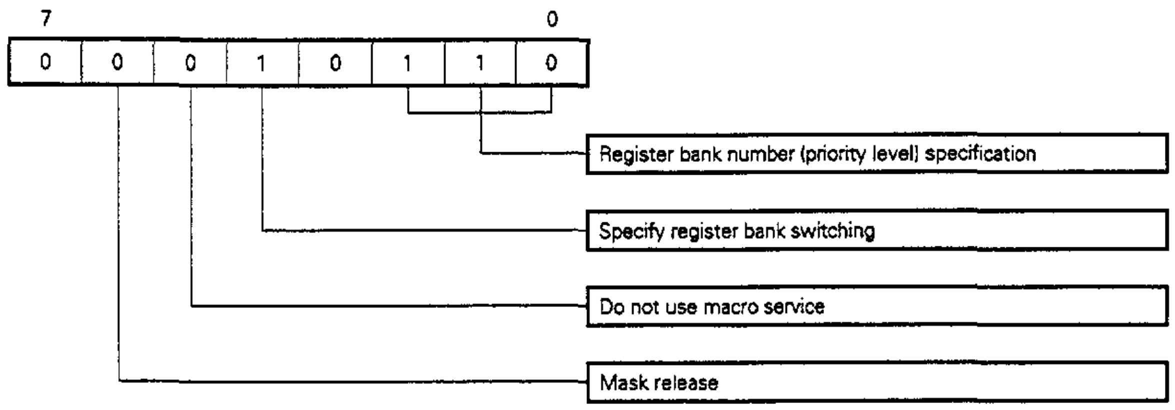
Note The structure field identifier for the assembler (RA70116-I) indicates the location of the vector PC in the on-chip RAM area's register bank 6.

If the above format is not used, add the starting address (xxE00H) of the on-chip RAM area (register bank 0) to register bank 6's offset address (00C0H) and the vector PC's offset address (0002H) to determine the location of the vector PC (xxECH) in register bank 6.

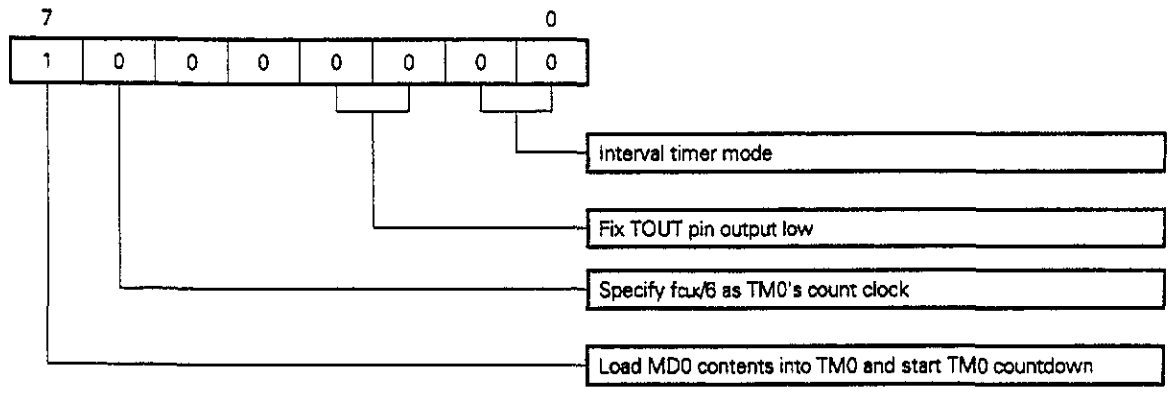
Remark "xx" indicates the high-order eight bits of the internal data area base address.

① The PS and vector PC for register bank 6 to be switched to must be already initialized.
When necessary, initialize other registers such as DS0 and DS1.

② TMIC0 (Timer unit 0's interrupt request control register)


7
0
0 0 0 1 0 1 1 0
Register bank number (priority level) specification
Specify register bank switching
Do not use macro service
Mask release

③ TMC0 (Timer control register 0)


7
0
1 0 0 0 0 0 0 0
Interval timer mode
Fix TOUT pin output low
Specify fCLK/6 as TM0's count clock
Load MD0 contents into TM0 and start TM0 countdown

④ Transfers contents of SS and SP in register bank 7 prior to switching to SS and SP in register bank 6 to be switched to.

⑤ When interrupt servicing from the peripheral hardware terminates, the FINT instruction must be executed before the RETI instruction or RETRBI instruction. Use the RETRBI instruction to return from register bank switching.
When the countdown results in a count value of 0, an interrupt request (INTTU0) occurs.

B.3.2 Register bank switching by instruction (BRKCS or MOVSPA instruction)

Switch from register bank 7 to register bank 5.

```
;****************************************************************************************************
;***                              Switch using BRKCS reg16 and MOVSPA                           ***
;****************************************************************************************************
;
                              :
          MOV       REGBANK. BK5. BVPC, OFFSET BANK5      ;①Initialization of vector PC
          MOV       REGBANK. BK5. BPS,  SEG BANK5         ;① PS
          MOV       REGBANK. BK5. BDS0, 0                 ;① DS0
          MOV       REGBANK. BK5. BDS1, 0                 ;① DS1
;
          MOV       AW, 5                                 ;
          BRKCS     AW                                    ;②
;
          EI
                              :
;====================================================================================================
;===                              Processing of register bank 5                                 ===
;====================================================================================================
;
BANK5:
          MOVSPA         ;③
             :
          RETRBI         ;④
```

① Set PS and PC for interrupt routine to PS and vector PC of register bank 5.
Set registers other than DS0 and DS1 as required.

② Execute BRKCS instruction to switch to register bank indicated by value set to reg16.
In this case, set "5" as the register in order to switch to register bank 5.

③ Transfer contents of SS and SP in register bank 7 prior to switching to SS and SP in the register bank to be switched to (register bank 5).

④ Return from new register bank. In this case, it is not necessary to execute a FINT instruction.

B.3.3 Register bank switching by instruction (MOVSPB or TSKSW instruction)

Switch from register bank 7 to register bank 6.

```
;*****************************************************************************************
;***                     Switch using MOVSPB reg16 and TSKSW reg16                    ***
;*****************************************************************************************
;
                                   :
          MOV       REGBANK. BK6. BPC,    OFFSET BANK6      ;① Initialization of PC save
          MOV       REGBANK. BK6. BPSW,  0E002H             ;① PSW save
          MOV       REGBANK. BK6. BPS,    SEG BANK6         ;① PS
          MOV       REGBANK. BK6. BDS0,  0                  ;① DS0
          MOV       REGBANK. BK6. BDS1,  0                  ;① DS1
;
          MOV       AW, 6                                   ;
          MOVSPB    AW                                      ;②
          TSKSW     AW                                      ;③
                                   :
;
;=========================================================================================
;===                         Processing of register bank 6                            ===
;=========================================================================================
;
BANK6:
          :
          MOV       AW, 7                                   ;④
          TSKSW     AW
          :
```

① To execute the TSKSW instruction, the PS, PC save area, SS, SP, and PSW save area in the register bank to be selected must be previously initialized. (In section B.3.3, this is done for the SS and SP by using the MOVSPB instruction.) When using the TSKSW instruction, the selected register bank's PSW save area and PC save area values are loaded.

② Use the MOVSPB instruction to set the SS and SP values before switching to the SS and SP switching destinations. In this case, you are switching to register bank 6, so set these values to register bank 6's AW register.

③ Use the TSKSW instruction to select register bank 6 as the register bank to be switched to and load the previously stored PC save area contents into the PC for a branch.
Execute the MOVSPB instruction and TSKSW instruction to switch to register bank 6.

④ If, as in 1, initialization was required before register bank switching, the TSKSW instruction would be executed to select register bank 7 for a branch. However, in this case, you are not using the MOVSPB instruction and so the stack cannot be used continuously.

Cautions 1. If the TSKSW instruction is used during register bank switching caused by a BRKCS instruction or an interrupt request occurrence, the contents of the PSW save area are destroyed and a return to the previous bank cannot be made. However, there is no problem if it is used during register bank switching caused by the TSKSW instruction.

2. The values for RB0, RB1, and RB2 in the PSW save area to be switched to must match the number of the register bank to be switched to.

B.4 Access to Internal Data Area

Indicates access when the internal data area (consisting of on-chip RAM and special function registers) is from 0FE00H to 0FFFFH. However, this is when setup of assembler (RA70116-I) pseudo instructions such as ASSUME and ASGNSFR has been completed.

```
;*********************************************************************************************
;***                               Initialization of register                             ***
;*********************************************************************************************
;
START:
          SETIDB    0FH Note          ;Set physical address base address (0F00H) in IDB register

          MOV       AW, DATA          ;Set DATA (0000H) in segment register
          MOV       DS0, AW           ;
;
;==============================================================================================
;===                      Initialization of special function register                     ===
;==============================================================================================
;
          MOV       P0,    00001111B  ;Set to output 1 from port 0 (P00 to P03) and
          MOV       PM0,   00000000B  ;to output 0 from P04 to P07
          MOV       PMC0,  00000000B  ;
;
          MOV       PRC,   01000100B  ;On-chip RAM access enable (set bit 6 to 1)
                      :
          MOV       AL, [FE00]        ;Fetch one byte of on-chip RAM contents
                      :               ;AL←[0FE00]
          MOV       AL, P0            ;Read P0 contents to AL
                      :
```

Note This pseudo instruction indicates the internal data area address in the assembler (RA70116-I). The *assembler generates the following instructions corresponding to this pseudo instruction description.*

```
PUSH    DS0
PUSH    0FFFFH
POP     DS0
MOV     DS0 : BYTE PTR [0FH], xx
POP     DS0
```

"xx" is set as the expression value (0FH) description for the operand.

B.5 Timer Unit

The interval timer (timer 1) and one-shot timer (timer 0) are used when timer unit interrupts occur.

(1) Interval timer mode

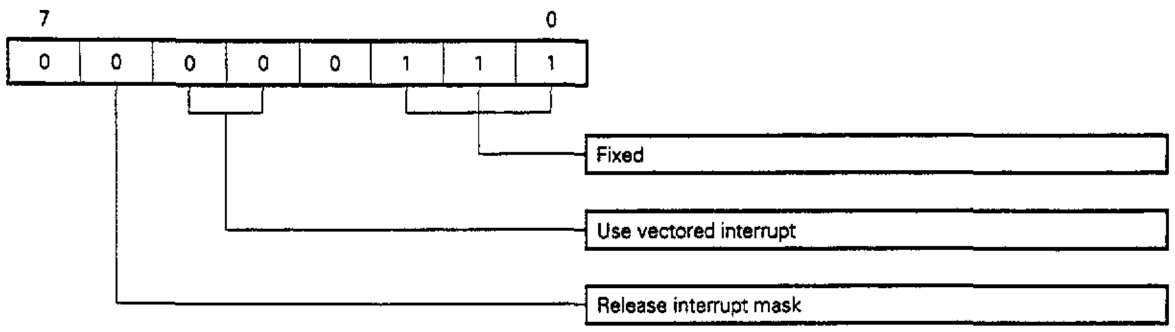
```
                                   :
;**********************************************************************************************
;***                   Interval timer mode setting (about 4.9 ms)                           ***
;**********************************************************************************************
;
                                   :
        MOV       TMIC2,  00000111B       ;①
        MOV       TMC1,   00000000B       ;②
        MOV       MD1,    0FF0H           ;③
                                   :
```

(2) One-shot timer mode

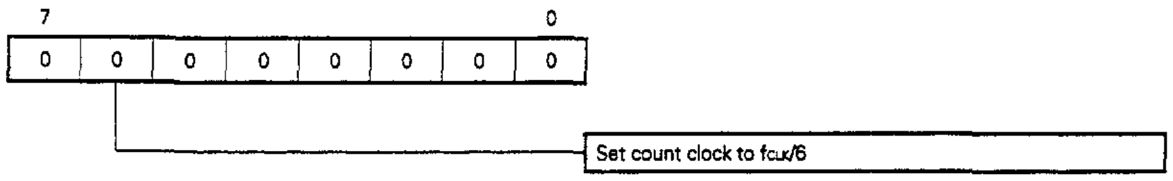
```
                        :
;*********************************************************************************
;***                One-shot timer mode setting (about 9.8 ms)                 ***
;*********************************************************************************
;
                        :
        MOV     TMIC0, 00000001B        ;④
        MOV     TMC0,  00001001B        ;⑤
        MOV     MD0,   0FF0H            ;⑥
                        :
                        :
;=================================================================================
;===                Vector address No. 28 (INTTU0) setting                     ===
;=================================================================================
;
        MOV     IY, 28*4
        MOV     WORD PTR[IY],      OFFSET INTTU0
        MOV     WORD PTR[IY+2],    SEG INTTU0
                        :
;
;=================================================================================
;===                Vector address No. 30 (INTTU1) setting                     ===
;=================================================================================
;
        MOV     IY, 30*4
        MOV     WORD PTR[IY],      OFFSET INTTU1
        MOV     WORD PTR[IY+2],    SEG INTTU1
;
        EI
        SET1    TMC1, 7             ;⑦
                    :
;=================================================================================
;===                          Timer start                                      ===
;=================================================================================
;
        SET1    TMC0, 5             ;⑧
                    :
;=================================================================================
;===                  Timer unit interrupt servicing                           ===
;=================================================================================
;
INTTU0:
                    :
        FINT
        RETI                        ;Return from interrupt
;
INTTU1:
                    :
        FINT
        RETI                        ;Return from interrupt
```

(1) Interval timer mode

① TMIC2 (timer unit interrupt request control register 2)



② TMC1 (timer control register 1)

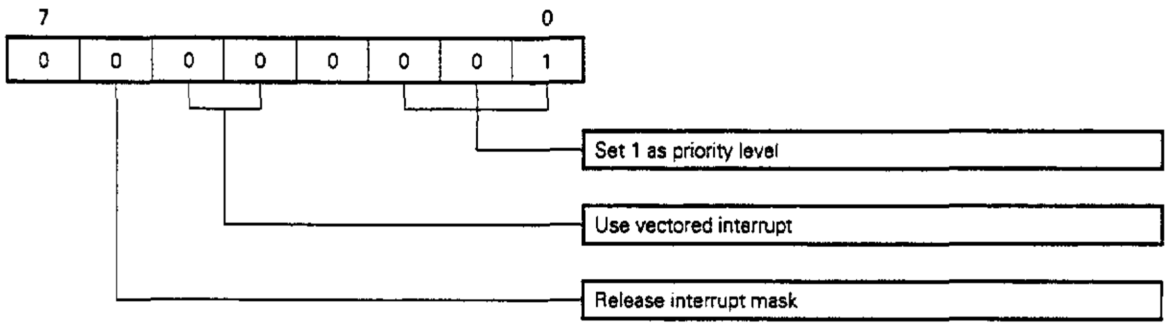


③ MD1

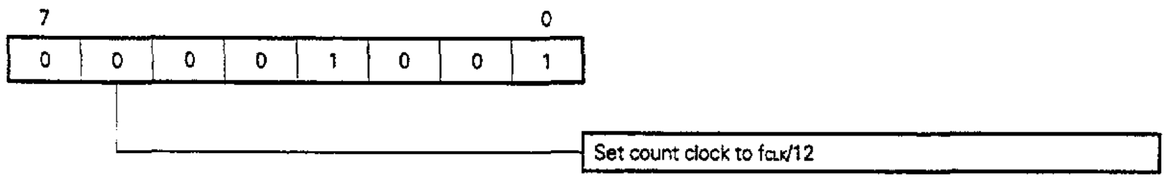
Sets the count value (FF0H). Interrupts occur at an interval of about 4.9 ms.

(2) One-shot timer mode

④ TMIC0 (timer unit interrupt request control register 0)



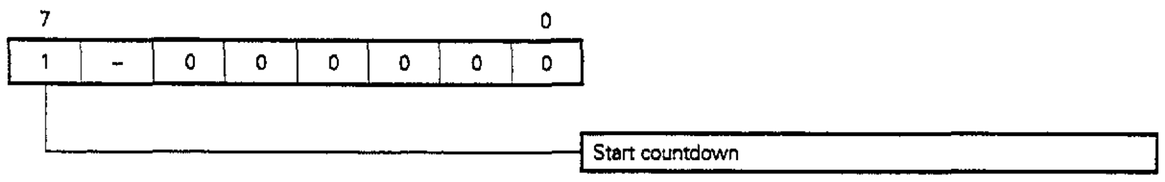
⑤ TMC0 (timer control register 0)



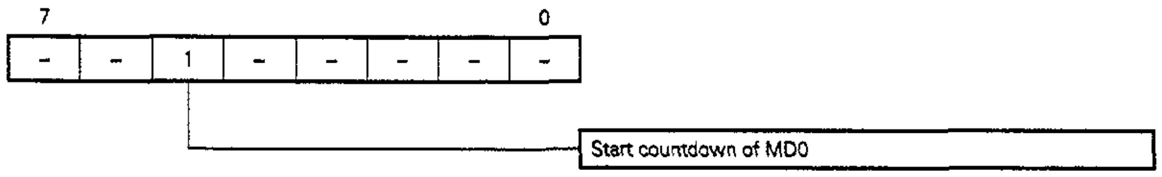
⑥ MD0

Sets the count value (FF0H). One interrupt occurs after an interval of about 9.8 ms.

⑦ TMC1 (timer control register 1)



⑧ TMC0 (timer control register 0)

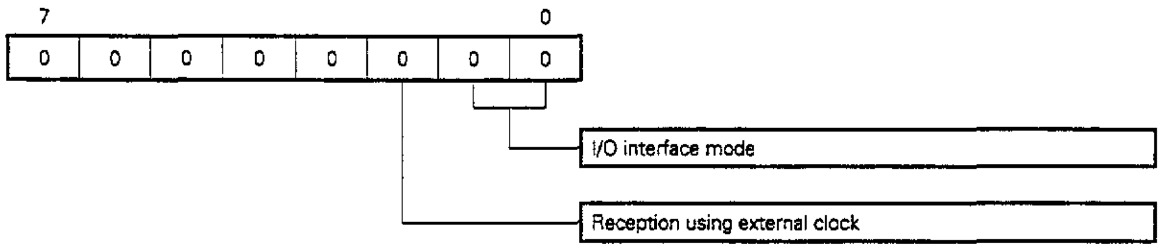


B.6 I/O Interface Mode

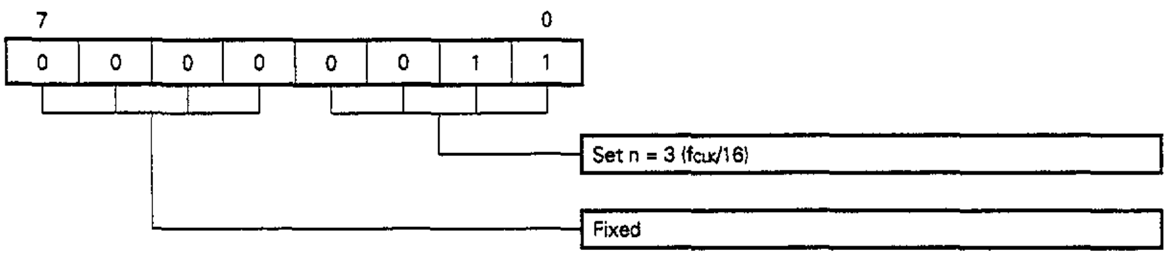
The I/O interface mode (reception) is used with vectored interrupts.

```
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;+++                     Special function register setting                              +++
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
        MOV     SCM0,  00000000B    ;①
        MOV     SCC0,  00000011B    ;②
        MOV     BRG0,  130          ;②
;
        MOV     SEIC0, 01000011B    ;③
        MOV     SRIC0, 10000111B    ;③
;
        MOV     P1,    11111111B    ;Set port 1 (P16) to control mode and execute SCK0 output
        MOV     PM1,   01000000B    ;
        MOV     PMC1,  00000000B    ;
;
        SET1    SCM0, 6             ;④
                  :
        EI        .                 ;SRIC0 (setting bit 7 to 1) enables interrupts and
                  :                 ;reception completion interrupt occurs.


;********************************************************************************************
;***      Serial data reception processing (reception completion interrupt being serviced)     ***
;********************************************************************************************
;
INTSR0:
                  :

        MOV       AL, RXB0          ;Fetch receive data from receive buffer (RxB0) and load to AL.

                  :

        FINT
        RETI
```

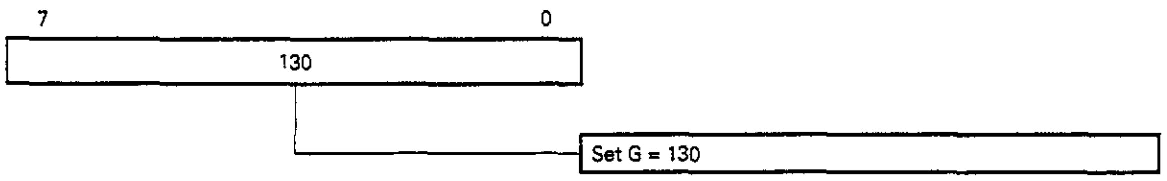
① SCM0 (serial mode register channel 0)



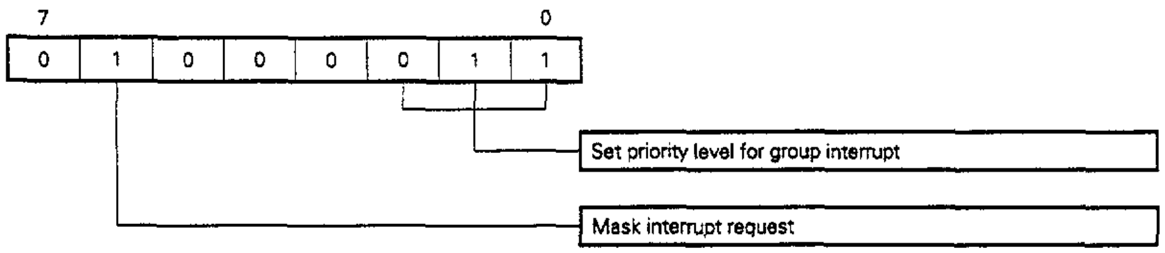
② SCC0 (serial control register 0)



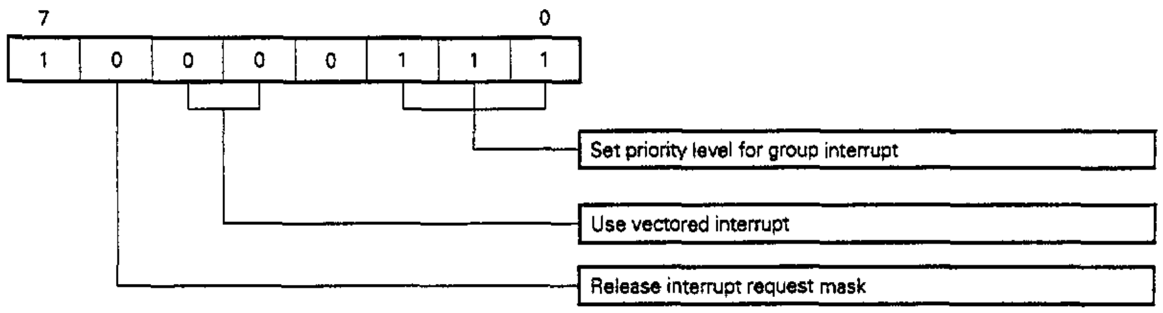
BRG0 (baud rate generator register 0)



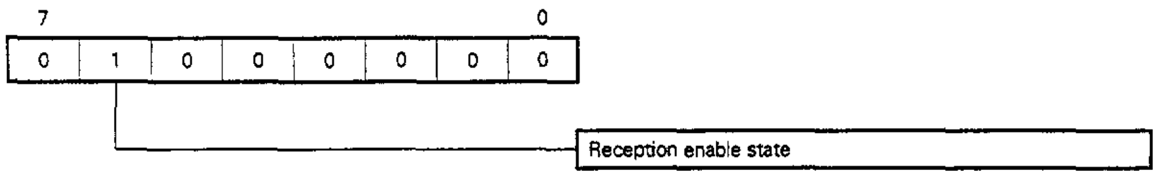
③ SEIC0 (serial error interrupt request control register 0)



SRIC0 (serial reception interrupt request control register 0)


7
0
1 0 0 0 0 1 1 1
Set priority level for group interrupt
Use vectored interrupt
Release interrupt request mask

④ SCM0 (serial mode register channel 0)



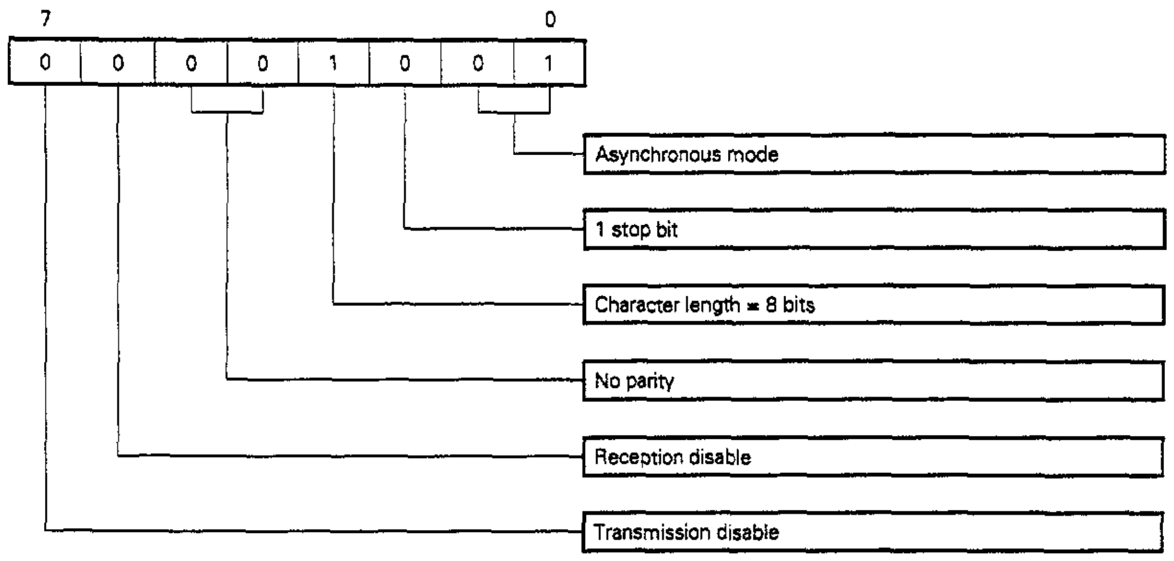
B.7 Macro Service

Shown below is an example in which (BAR_CODE) is the starting address of the memory space where the send data are stored and (REGBANK) is the starting address of macro service channel 0.

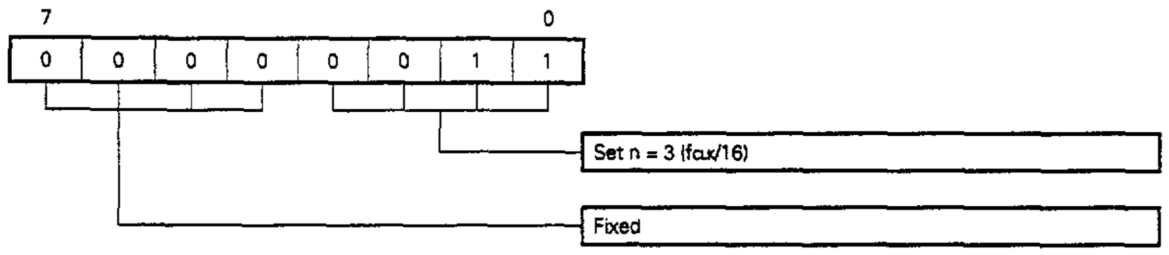
B.7.1 Normal mode (serial interface UART transmission)

```
                      :
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;+++                        Special function register setting                        +++
;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
        MOV     SCM0,  00001001B    ;①
        MOV     SCC0,  00000011B    ;②
        MOV     BRG0,  130          ;②
;
        MOV     SEIC0, 01000011B    ;③
        MOV     STIC0, 00110111B    ;④
;

;***************************************************************************************
;***               Initialization of macro service (channel 2) normal mode           ***
;***************************************************************************************
;
        MOV     STMS0, 00000010B                    ;⑤
;
        MOV     IY, OFFSET REGBANK+8*2              ;⑥
        MOV     BYTE PTR  [IY], 15                  ;⑦
        MOV     BYTE PTR  [IY+1], LOW TXB0          ;⑧
        MOV     WORD PTR [IY+4], OFFSET BAR_CODE    ;⑨
        MOV     WORD PTR [IY+6], SEG BAR_CODE       ;⑩
;

;***************************************************************************************
;***                           Register bank 3 setting                               ***
;***************************************************************************************
;                                                   ;See section B.3

;***************************************************************************************
;
        SET1    SCM0, 7                             ;⑪
                      :
```

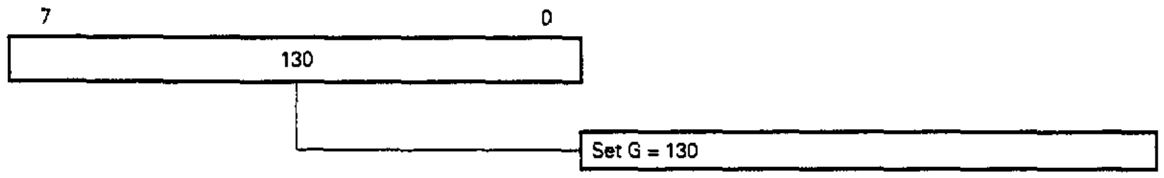
① SCM0 *(serial mode register channel 0)*



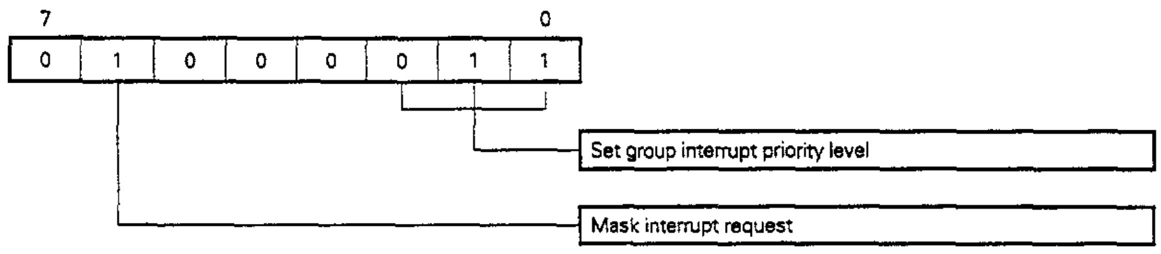
② SCC0 *(serial control register 0)*



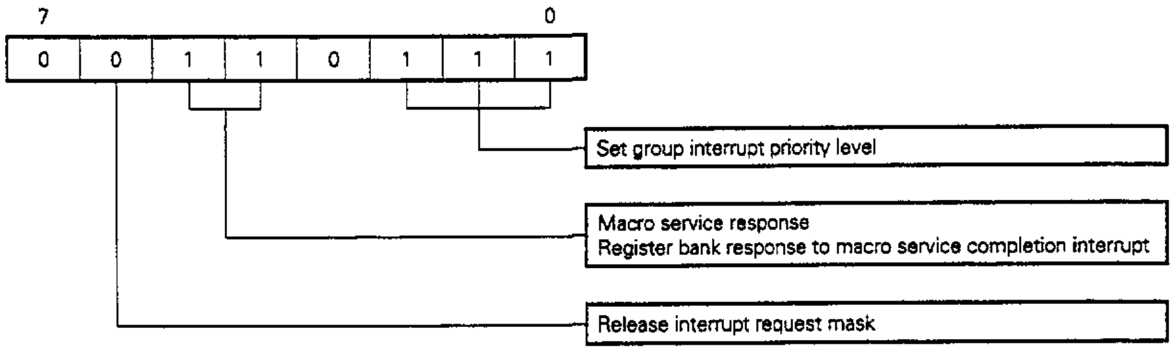
BRG0 (Baud rate generator register 0)



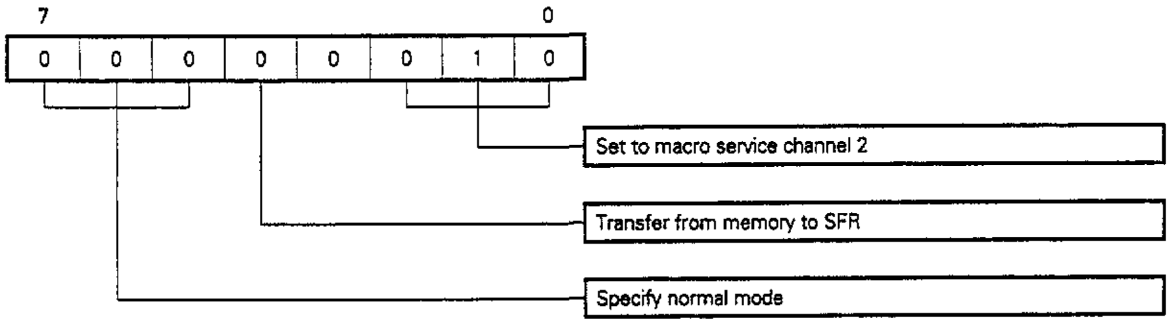
③ SEIC0 (serial error interrupt request control register 0)



④ STIC0 (serial transmission interrupt request control register 0)


7
0
0 0 1 1 0 1 1 1
Set group interrupt priority level
Macro service response
Register bank response to macro service completion interrupt
Release interrupt request mask

⑤ SRMS0 (macro service control register 0)

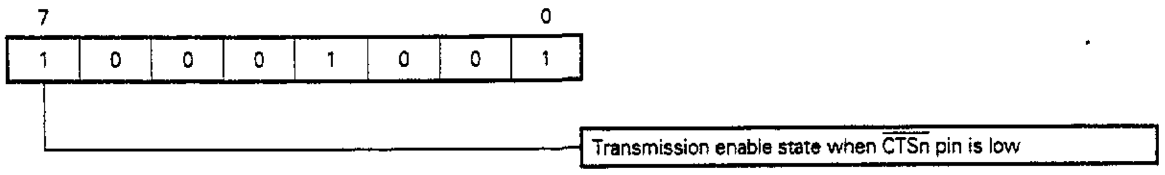


⑥ Set the starting offset address of macro service channel 2 to the index register (IY).
⑦ MSC [IY + 0]: Set number of transfers (15) using macro service.
⑧ SFRP [IY + 1]: Set low-order byte of special function register (TxB0) address.
LOW: (RA70116-I) assembler's byte separator operand; return value of low-order byte in expression.
⑨ MSP [IY + 4]: Set offset value of memory address to which data is sent by macro service.
⑩ MSS [IY + 6]: Set segment value of memory address to which data is sent by macro service.

The memory address to which data is sent is MSS × 16 + MSP.

The n value in brackets ([+ n]) indicates the offset from each macro service channel's start address.

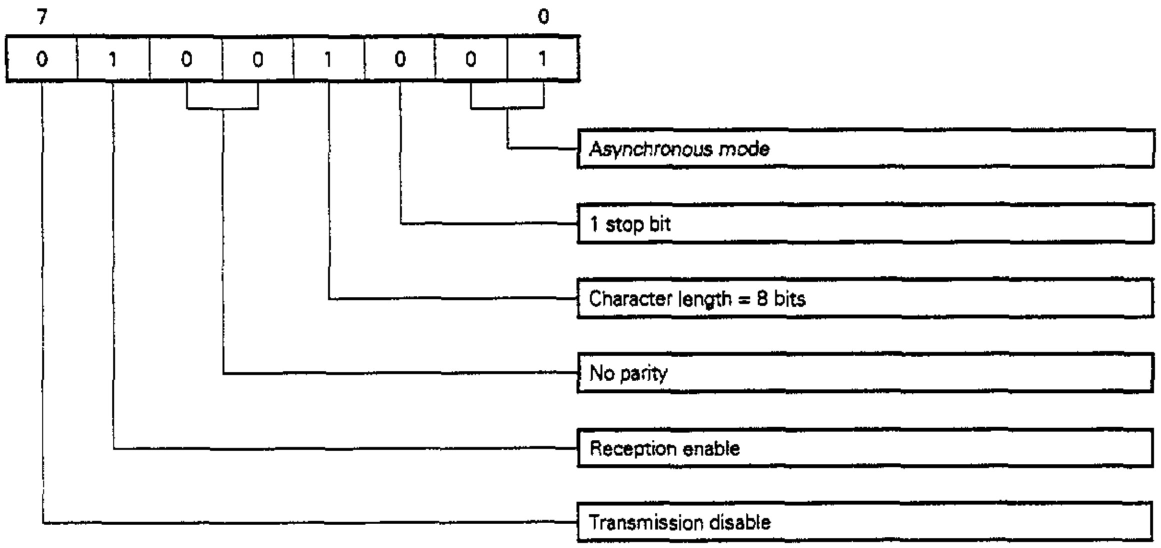
⑪ SCM0 (serial mode register channel 0)


7
0
1 0 0 0 1 0 0 1
Transmission enable state when $\overline{\text{CTSn}}$ pin is low

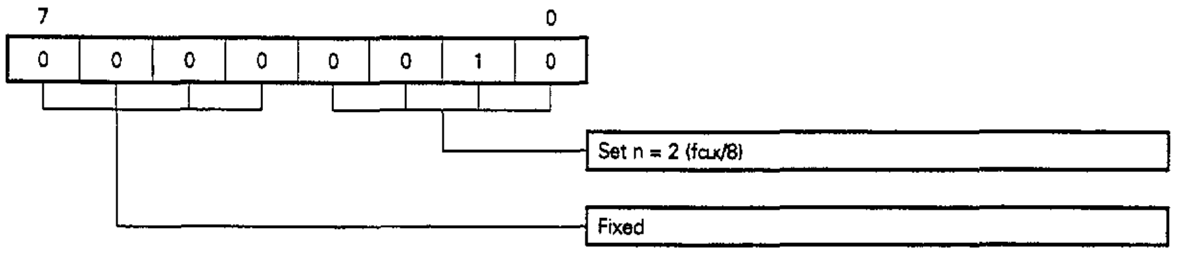
B.7.2 Character search mode (serial interface UART reception)

```
                    :
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;+++                 Special function register setting                       +++
;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
        MOV     SCM0,   01001001B       ;①
        MOV     SCC0,   00000010B       ;②
        MOV     BRG0,   130             ;
;
        MOV     SEIC0,  01000011B       ;③
        MOV     SRIC0,  00110111B       ;③
        MOV     STIC0,  01000111B       ;③
;

;********************************************************************************
;***     Initialization of macro service (channel 2) character search mode    ***
;********************************************************************************
;
        MOV     SRMS0, 10000010B                        ;④
;
        MOV     IY, OFFSET REGBANK+8*2                  ;⑤
        MOV     BYTE   PTR [IY], 15                     ;⑥
        MOV     BYTE   PTR [IY+1], LOW RXB0             ;⑦
        MOV     BYTE   PTR [IY+2], 0AH                  ;⑧
        MOV     WORD PTR [IY+4], OFFSET BAR_CODE        ;⑨
        MOV     WORD PTR [IY+6], SEG BAR_CODE           ;⑩
```

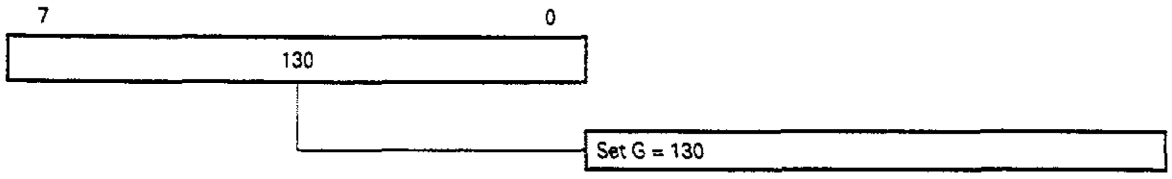
① SCM0 (serial mode register channel 0)



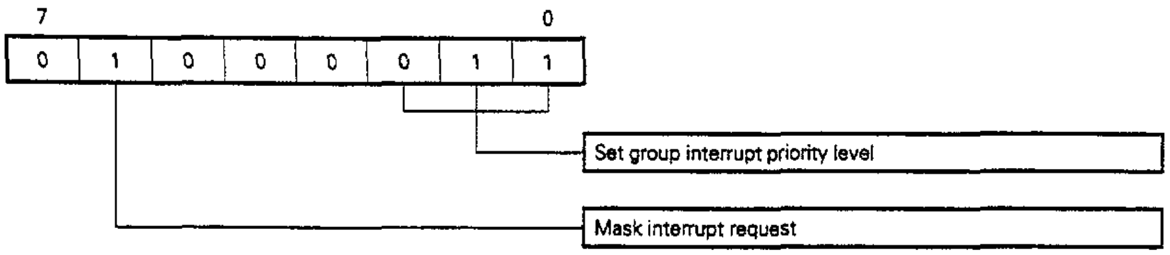
② SCC0 (serial control register 0)



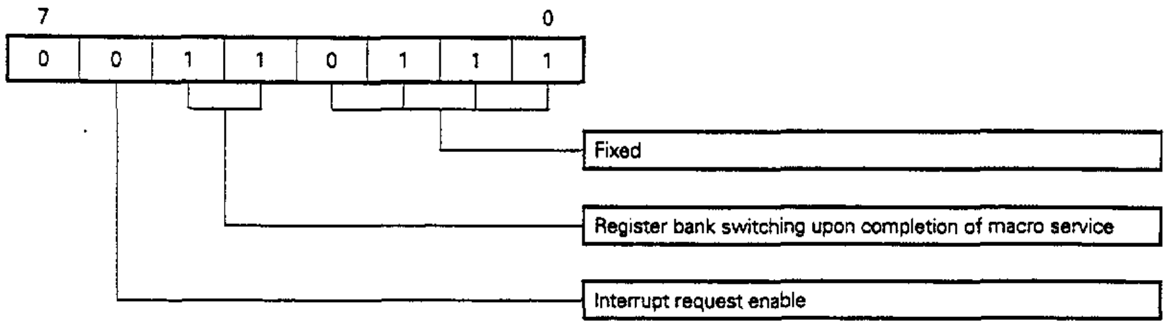
BRG0 (Baud rate generator register 0)



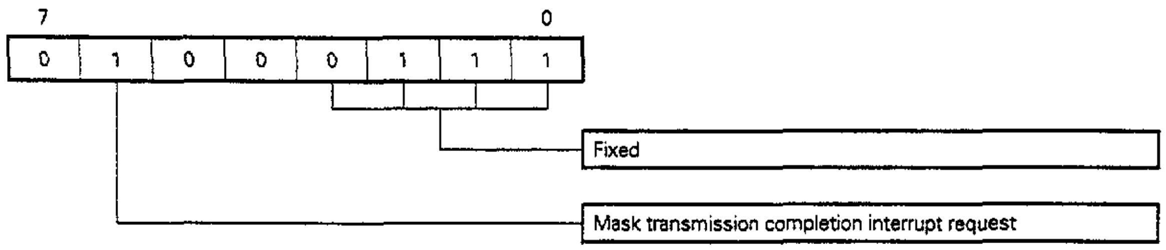
③ SEIC0 (serial error interrupt request control register 0)



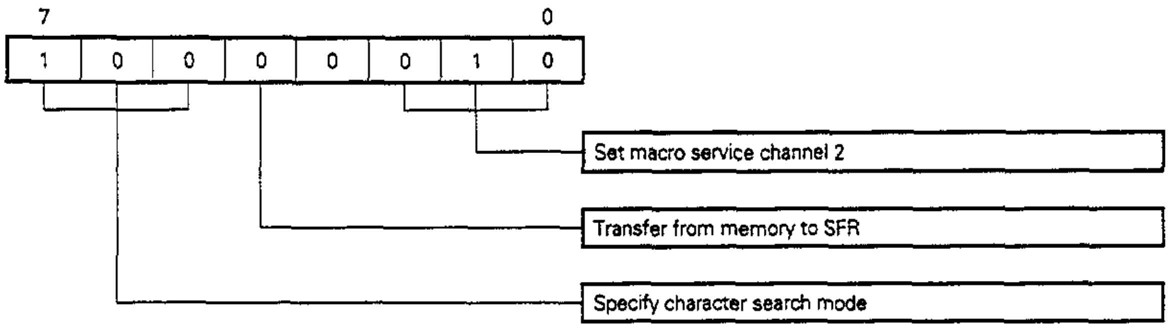
SRIC0 (serial reception interrupt request control register 0)



STIC0 (serial transmission interrupt request control register 0)



④ SRMS0 (macro service control register 0)


7
0
1 0 0 0 0 0 1 0
Set macro service channel 2
Transfer from memory to SFR
Specify character search mode

⑤ Set the starting of macro service channel 2 to the index register (IY).
⑥ MSC [IY + 0]: Set number of transfers (15) using macro service.
⑦ SFRP [IY + 1]: Set low-order byte of special function register (RxB0) address.
LOW: (RA70116-I) assembler's byte separator operand; return value of low-order byte in expression.
⑧ SCHR [IY + 2]: Set 8-bit data (0AH) to be compared during character search mode.
⑨ MSP [IY + 4]: Set offset value of memory address to which data is sent by macro service.
⑩ MSS [IY + 6]: Set segment value of memory address to which data is sent by macro service.

The memory address to which data is sent is MSS x 16 + MSP.

The n value in brackets ([+ n]) indicates the offset from each macro service channel's start address.

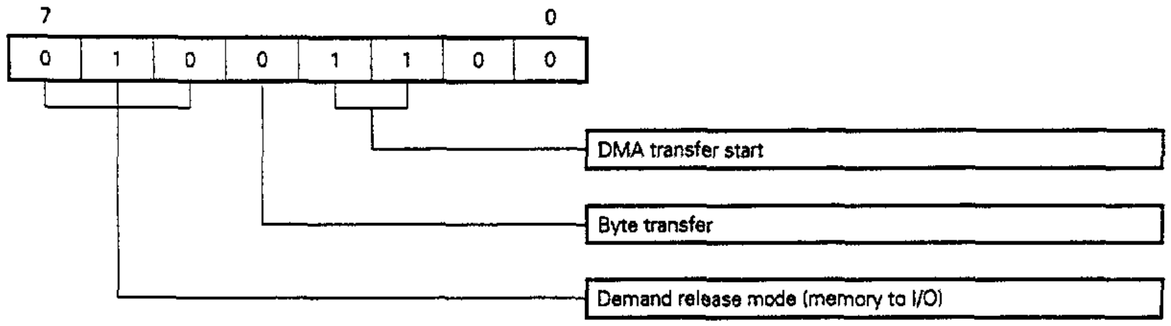
B.8 DMA Controller

Shown below is an example in which (MSG_TBL_SAR) is the starting address of the memory space where the source data to be transferred is stored and (MSG_TBL_DAR) is the starting address of the transfer destination (data storage memory).

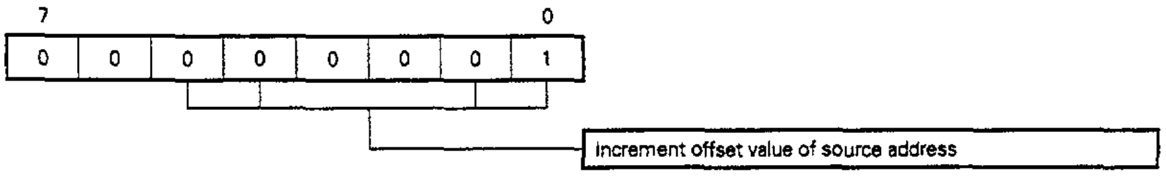
B.8.1 Demand release mode

```
;
;*************************************************************************************************
;***                      Initialization of DMA controller (channel 0)                        ***
;*************************************************************************************************
;
        MOV     DMAM0,  01001100B                       ;①
        MOV     DMAC0,  00000001B                       ;②
;
        MOV     SAR0L,     ⎫
        MOV     SAR0M,     ⎬ MSG_TBL_SAR                ;③
        MOV     SAR0H,     ⎭
        MOV     TC0, 3599                               ;④
;
                    :
```

① DMAM0 (DMA mode register 0)



② DMAC0 (DMA control register 0)



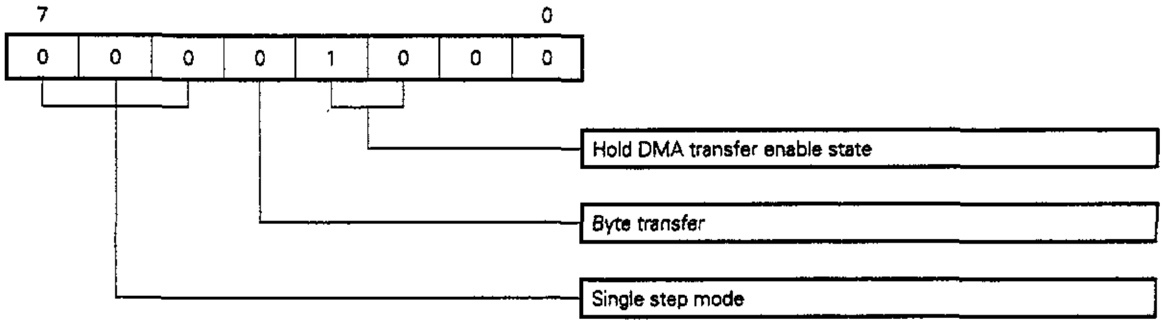
③ Set source address for DMA transfer.

④ Set DMA transfer count (3,600 times)
Because this is byte transfer, 3,600 bytes of data are required.

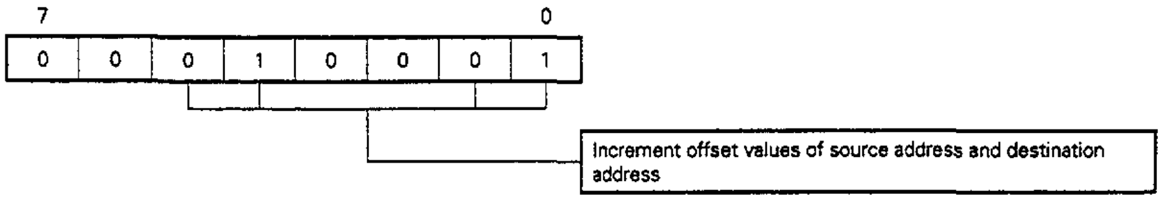
B.8.2 Single step mode

```
;
;*******************************************************************************************
;***                    Initialization of DMA controller (channel 0)                    ***
;*******************************************************************************************
;
        MOV     DMAM0,  00001000B                           ;①
        MOV     DMAC0,  00010001B                           ;②
;
        MOV     SAR0L,  ┐
        MOV     SAR0M,  ├ MSG_TBL_SAR                       ;③
        MOV     SAR0H,  ┘
        MOV     DAR0L,  ┐
        MOV     DAR0M,  ├ MSG_TBL_DAR                       ;④
        MOV     DAR0H,  ┘
        MOV     TC0, 3599                                   ;⑤
;
                  :
        SET1    DMAM0, 2                                    ;DMA transfer start
                  :
```

① DMAM0 (DMA mode register 0)



② DMAC0 (DMA control register 0)

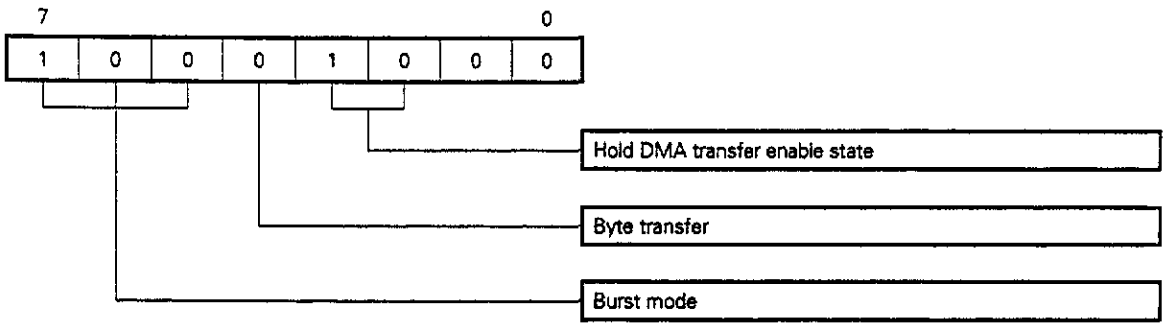

7
0
0 0 0 1 0 0 0 1
Increment offset values of source address and destination address

③ Set source address for DMA transfer.
④ Set destination for DMA transfer.
⑤ Set DMA transfer count (3,600 times)
Because this is byte transfer, 3,600 bytes of data are required.

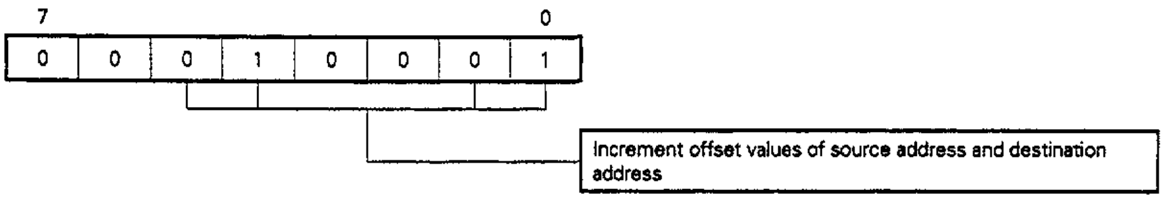
B.8.3 Burst mode

```
;
;*********************************************************************************************
;***                      Initialization of DMA controller (channel 0)                   ***
;*********************************************************************************************
;
        MOV     DMAM0,  10001000B                    ;①
        MOV     DMAC0,  00010001B                    ;②
;
        MOV     SAR0L,  ⎫
        MOV     SAR0M,  ⎬ MSG_TBL_SAR                ;③
        MOV     SAR0H,  ⎭
        MOV     DAR0L,  ⎫
        MOV     DAR0M   ⎬ MSG_TBL_DAR                ;④
        MOV     DAR0H   ⎭
        MOV     TC0, 3599                            ;⑤
;
                   :
        SET1    DMAM0, 2                             ;DMA transfer start
                   :
```

① DMAM0 (DMA mode register 0)



② DMAC0 (DMA control register 0)


7
0
0
0
0
1
0
0
0
1
Increment offset values of source address and destination address

③ Set source address for DMA transfer.
④ Set destination address for DMA transfer.
⑤ Set DMA transfer count (3,600 times)
Because this is byte transfer, 3,600 bytes of data are required.