

致尊敬的顾客

关于产品目录等资料中的旧公司名称

NEC电子公司与株式会社瑞萨科技于2010年4月1日进行业务整合（合并），整合后的新公司暨“瑞萨电子公司”继承两家公司的所有业务。因此，本资料中虽还保留有旧公司名称等标识，但是并不妨碍本资料的有效性，敬请谅解。

瑞萨电子公司网址：<http://www.renesas.com>

2010年4月1日
瑞萨电子公司

【发行】瑞萨电子公司（<http://www.renesas.com>）

【业务咨询】<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.

740 族

740 族参考程序集

目录

1. 740 族的基本程序.....	2
2. 有效利用 740 族固有指令的处理	3
2.1 存储器间的运算	3
2.2 数据的位判定转移.....	5
2.3 数据的位处理（置位/复位）	7
2.4 数据的循环移位	8
3. 基本处理的程序例子	11
3.1 RAM 清除	11
3.2 数据传送（RAM）	13
3.3 数据传送（固定 ROM 地址）	15
3.4 数据传送（可变 ROM 地址）	17
3.5 数据的重新排序	20
3.6 16 位数据加法运算（二进制）	25
3.7 16 位数据减法运算（二进制）	27
3.8 16 位数据加法运算（BCD）	29
3.9 16 位数据减法运算（BCD）	31
3.10 16 位数据乘法运算（二进制）	33
3.11 16 位数据除法运算（二进制）	37
4. 应用程序例子	42
4.1 文件处理（传送）	42
4.2 文件处理（交换）	44
4.3 码转换（压缩 BCD→非压缩 BCD）	47
4.4 码转换（非压缩 BCD→非压缩 BCD）	51
4.5 码转换（BIN→BCD）	55
4.6 码转换（BCD→BIN）	60
4.7 SGN 函数.....	65
4.8 DCB 12 位浮点四则运算.....	66
5. 替代指令	76
5.1 交换累加器	76
5.2 颠倒累加器的位顺序.....	77
5.3 存储器的置位.....	77
5.4 存储器的位清除	77
5.5 存储器的位反转	77
6. 程序利用的注意事项	77
7. 参考文献	77

1. 740 族的基本程序

740 族指令的特点概括如下：

- (1) 能有效使用 ROM 区的高效率的指令群和丰富的寻址方式。
- (2) 能对累加器、存储器、I/O 进行处理的位操作指令和位测试/转移指令。
- (3) 丰富的中断源和处理功能。
- (4) 字节单位处理和表参照功能卓绝的变址寻址功能。
- (5) 不需要软件补正的 10 进制运算功能。
- (6) 不经累加器就能进行的存储器和存储器之间、I/O 和 I/O 之间、存储器和 I/O 之间的运算功能。

2. 有效利用 740 族固有指令的处理

2.1 存储器间的运算

(1) 加法运算（助记符 ADC）

ADC 指令在 X 变址模式标志 T 为 1 时将 M(X)、存储器 M 和进位标志 C 的内容相加，结果存入 M(X) 和进位标志 C。此时，累加器 A 的内容不变，而状态标志发生变化。其中，M(X) 为变址寄存器 X 指向地址的存储器。

例)

CLC		①
SET		②
LDX	#ADDATA	③
ADC	\$10, X	④
CLT		⑤

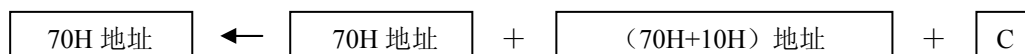
(说明)

①将进位标志 C 的内容置 0。

②将 X 变址模式标志 T 的内容置 1。

③将#ADDATA（例：70H）装入变址寄存器 X。

④将存储器的 70H 地址的内容、80H 地址（70H+10H）的内容和进位标志 C 的内容相加，结果存入 70H 地址和进位标志 C。



⑤将 X 变址模式标志 T 的内容置 0。

例如：70H 地址的内容为 53H、80H 地址的内容为 21H，在①～⑤的指令执行后 70H 地址的内容变为 74H、进位标志 C 的内容变为 0。

(2) 減法运算 (助记符 SBC)

SBC 指令在 X 变址模式标志 T 为 1 时将 M(X) 的内容减去存储器 M 的内容和进位标志 C 的内容的补码, 结果存入 M(X) 和进位标志 C。此时, 累加器 A 的内容不变, 而状态标志发生变化。其中, M(X) 为变址寄存器 X 指向地址的存储器。

例)

SEC		①
SET		②
LDX	#SBDATA	③
SBC	\$10, X	④
CLT		⑤

(说明)

①将进位标志 C 的内容置 1。

②将 X 变址模式标志 T 的内容置 1。

③将#SBDATA (例: \$80) 装入变址寄存器 X。

④将存储器的 80H 地址的内容减去 90H 地址 (80H+10H) 的内容和进位标志 C 的内容的补码, 结果存入 80H 地址和进位标志 C。

80H 地址	←	80H 地址	−	(80H+10H) 地址	−	C 的补码
--------	---	--------	---	--------------	---	-------

⑤将 X 变址模式标志 T 的内容置 0。

例如: 80H 地址的内容为 53H、90H 地址的内容为 21H, 在①~⑤的指令执行后 80H 地址的内容变为 32H、进位标志 C 的内容变为 1。

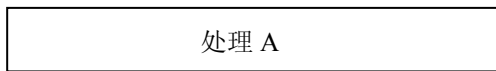
2.2 数据的位判定转移

(1) 置位时转移 (助记符 BBS)

BBS 指令测试累加器 A 或者存储器 M 指定的位 i。如果该位为 1，就转移到指定地址。转移目标地址为相对地址。如果该位为 0，就执行下一条指令。

例)

BBS 0, P0, START

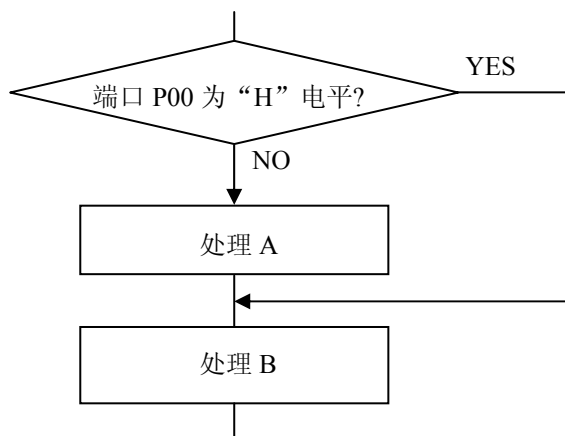


START:



(说明)

如果端口 P0 的位 0 为 1，就转移到 START。如果端口 P0 的位 0 为 0，就执行下一条指令。



(2) 清除时转移 (助记符 BBC)

BBC 指令测试累加器或者存储器 M 指定的位 i。如果该位为 0，就转移到指定地址。转移目标地址为相对地址。如果该位为 1，就执行下一条指令。

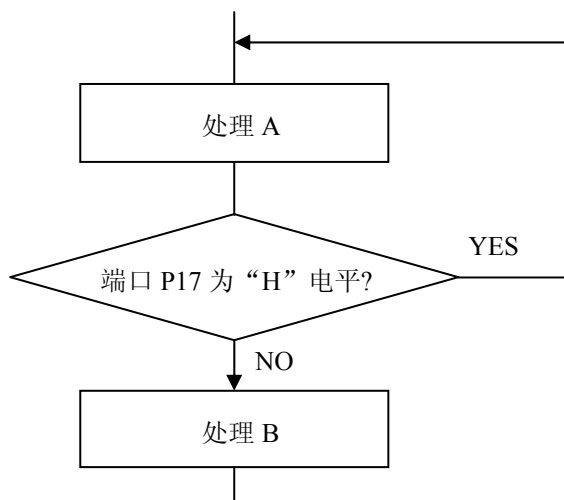
例)

RESET:



(说明)

如果端口 P1 的位 7 为 0，就转移到 RESET。如果端口 P1 的位 7 为 1，就执行下一条指令。



2.3 数据的位处理（置位/复位）

（1）置位（助记符 SEB）

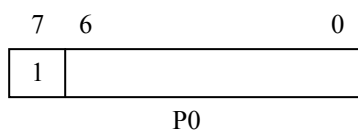
SEB 指令将累加器 A 或者存储器 M 指定的位 i 的内容置 1。

例）

SEB 7, P0

（说明）

将端口 P0 的位 7 的内容置 1。



例如：端口 P0 的内容为 53H，在指令执行后端口 P0 的内容变为 D3H。

（2）位清除（助记符 CLB）

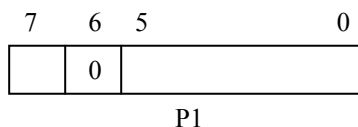
CLB 指令将累加器 A 或者存储器 M 指定的位 i 的内容清 0。

例）

CLB 6, P1

（说明）

将端口 P1 的位 6 的内容清 0。



例如：端口 P1 的内容为 53H，在指令执行后端口 P1 的内容变为 13H。

2.4 数据的循环移位

(1) 左循环（助记符 ROL）

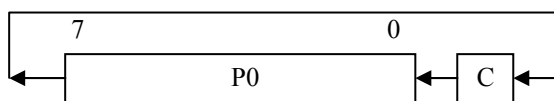
ROL 指令将累加器 A 或者存储器 M 和进位标志 C 相连，其内容左循环 1 位。累加器 A 或者存储器 M 的位 0 存放进位标志 C 的内容，进位标志 C 存放累加器 A 或者存储器 M 的位 7 的内容。

例)

ROL P0

(说明)

将端口 P0 和进位标志 C 相连，其内容左循环 1 位。



例如：端口 P0 的内容为 53H、进位标志 C 的内容为 1，在指令执行后端口 P0 的内容变为 A7H、进位标志 C 的内容变为 0。

(2) 移位（助记符 ASL）

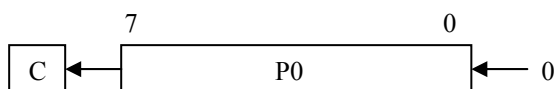
ASL 指令将累加器 A 或者存储器 M 的所有位左移 1 位。此时，累加器 A 或者存储器 M 的位 0 变为 0。另外，进位标志 C 存放累加器 A 或者存储器 M 的位 7 的内容。

例)

ASL P0

(说明)

将端口 P0 的所有位左移 1 位。



例如：端口 P0 的内容为 53H、进位标志 C 的内容为 1，在指令执行后端口 P0 的内容变为 A6H、进位标志 C 的内容变为 0。

(3) 右循环（助记符 ROR）

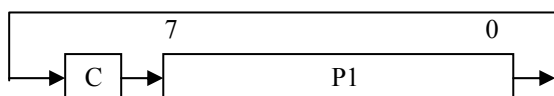
ROR 指令将累加器 A 或者存储器 M 和进位标志 C 相连，其内容右循环 1 位。累加器 A 或者存储器 M 的位 7 存放进位标志 C 的内容，进位标志 C 存放累加器 A 或者存储器 M 的位 0 的内容。

例)

ROR P1

(说明)

将端口 P1 和进位标志 C 相连，其内容右循环 1 位。



例如：端口 P1 的内容为 53H、进位标志 C 的内容为 1，在指令执行后端口 P1 的内容变为 A9H、进位标志 C 的内容变为 1。

(4) 右移位（助记符 LSR）

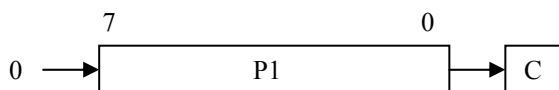
LSR 指令将累加器 A 或者存储器 M 的所有位右移 1 位。此时，累加器 A 或者存储器 M 的位 7 变为 0。另外，进位标志 C 存放累加器 A 或者存储器 M 的位 0 的内容

例)

LSR P1

(说明)

将端口 P1 的所有位右移 1 位。



例如：端口 P1 的内容为 53H、进位标志 C 的内容为 1，在指令执行后端口 P1 的内容变为 29H、进位标志 C 的内容变为 1。

(5) 高位和低位的交换 (助记符 RRF)

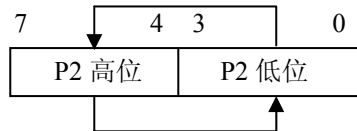
RRF 指令将存储器 M 的内容右循环 4 位。其结果，存储器 M 的内容的高 4 位和低 4 位进行了交换。

例)

RRF P2

(说明)

将端口 P2 的内容的高 4 位和低 4 位进行交换。



例如：端口 P2 的内容为 53H，在指令执行后端口 P2 的内容变为 35H。

3. 基本处理的程序例子

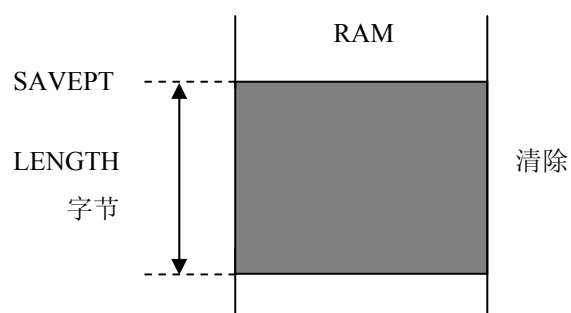
3.1 RAM 清除

(1) 概要

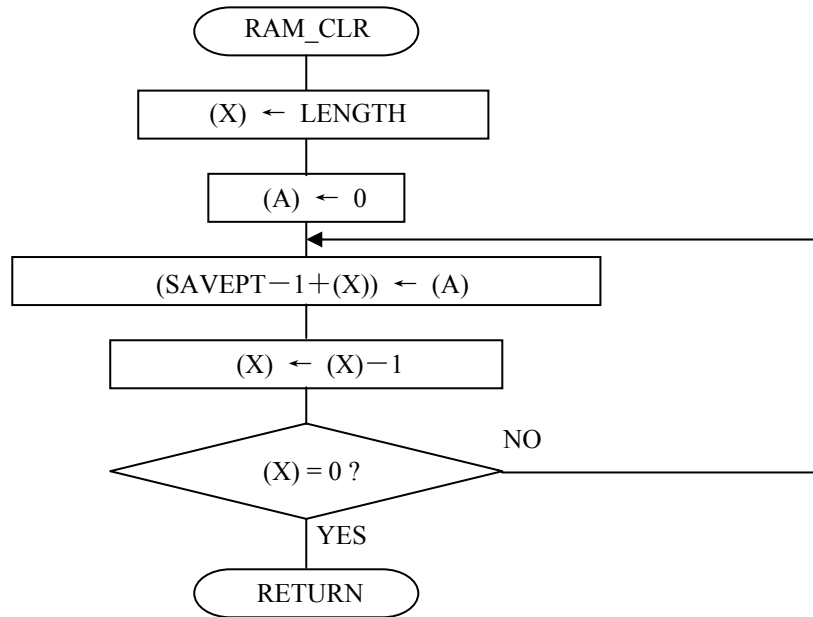
指定区域，清除 RAM。

(2) 说明

从 RAM 地址 SAVEPT 开始清除 LENGTH 字节的 RAM。



(3) 流程图



(4) 程序清单

```

;*****
;      RAM clear routine
;*****
RAM_CLR:
    LDX    #LENGTH          ;RAM length
    LDA    #$00
RAMCL1:  STA    SAVEPT-1,X    ;Clear from SAVEPT
        DEX                ; - to SAVEPT+LENGTH-1
        BNE    RAMCL1        ;Clear end ?
        RTS                 ;Yes
  
```

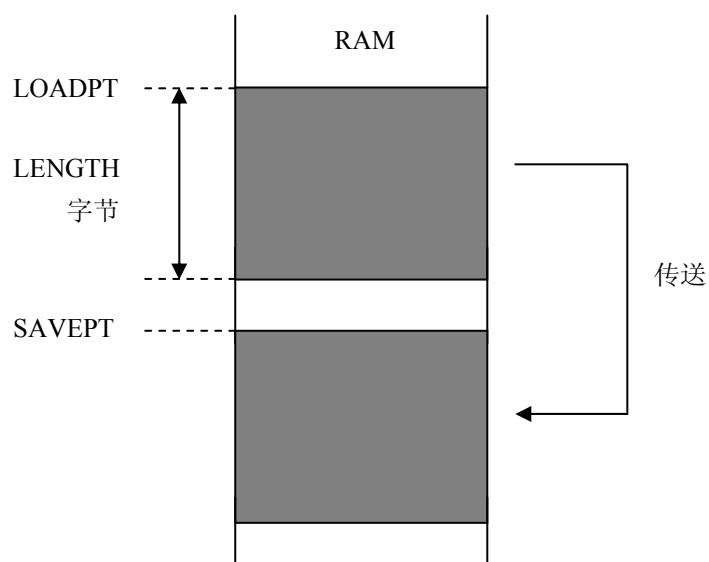
3.2 数据传送 (RAM)

(1) 概要

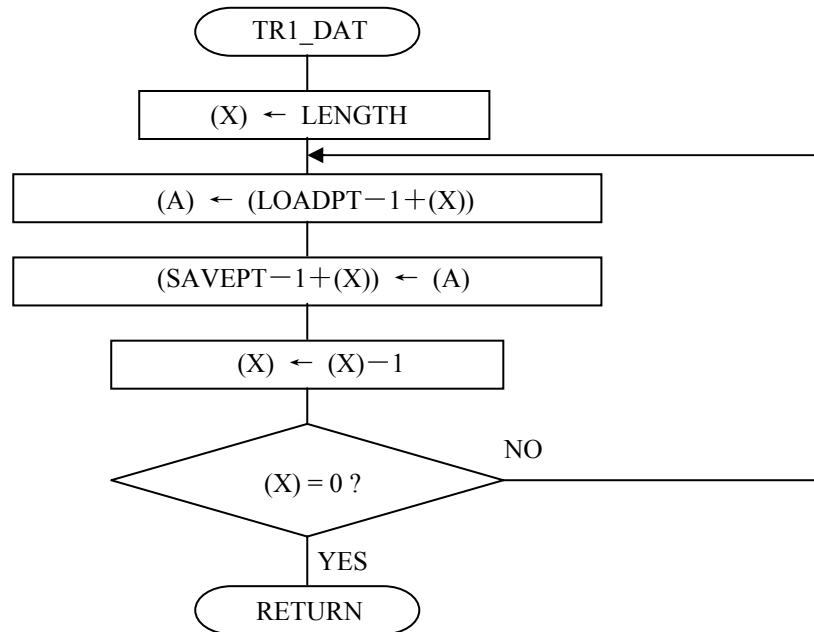
在 RAM 区内传送数据。

(2) 说明

从 RAM 地址 LOADPT 开始将 LENGTH 字节的数据传送到 SAVEPT 开始的连续地址。



(3) 流程图



(4) 程序清单

```

;*****
;      RAM data transfer routine
;*****
;
TR1_DAT:
    LDX    #LENGTH          ;RAM length
TR1_01:  LDA    LOADPT-1,X    ;Transfer data from LOADPT
    STA    SAVEPT-1,X        ; - to SAVEPT
    DEX
    BNE    TR1_01           ;Transfer end ?
    RTS                      ;Yes
  
```

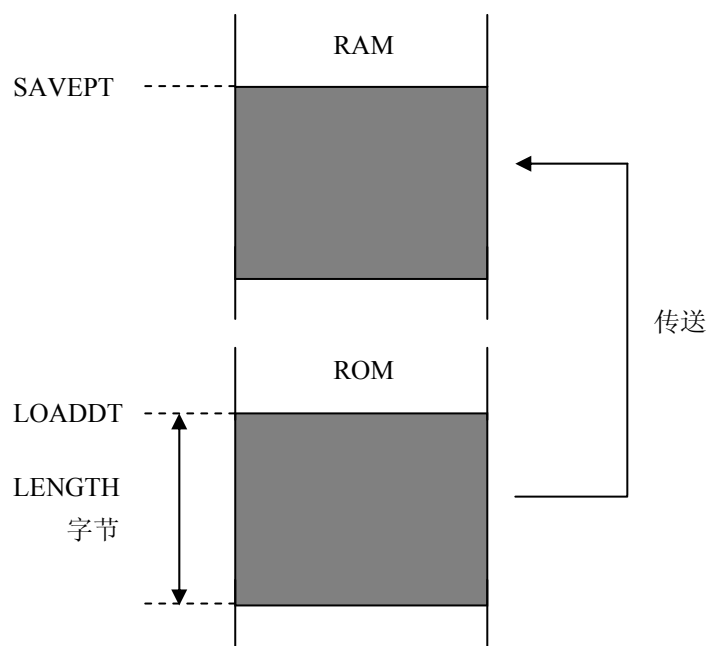
3.3 数据传送（固定 ROM 地址）

（1）概要

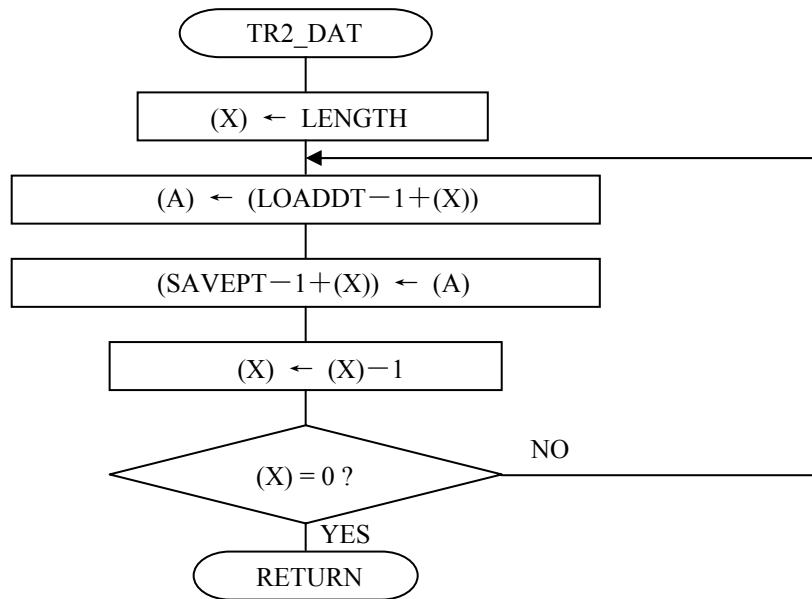
传送 ROM 区内的数据（固定地址）。

（2）说明

从 ROM 地址 LOADDT 开始将 LENGTH 字节的数据传送到 SAVEPT 开始的连续 RAM 地址。



(3) 流程图



(4) 程序清单

```

;*****
;      ROM data transfer routine ( address fixed )
;*****
;
TR2_DAT:
    LDX    #LENGTH          ;ROM length
TR2_01:  LDA    LOADDT-1,X    ;Transfer data from LOADDT
    STA    SAVEPT-1,X        ; - to SAVEPT
    DEX
    BNE    TR2_01           ;Transfer end ?
    RTS                      ;Yes
  
```

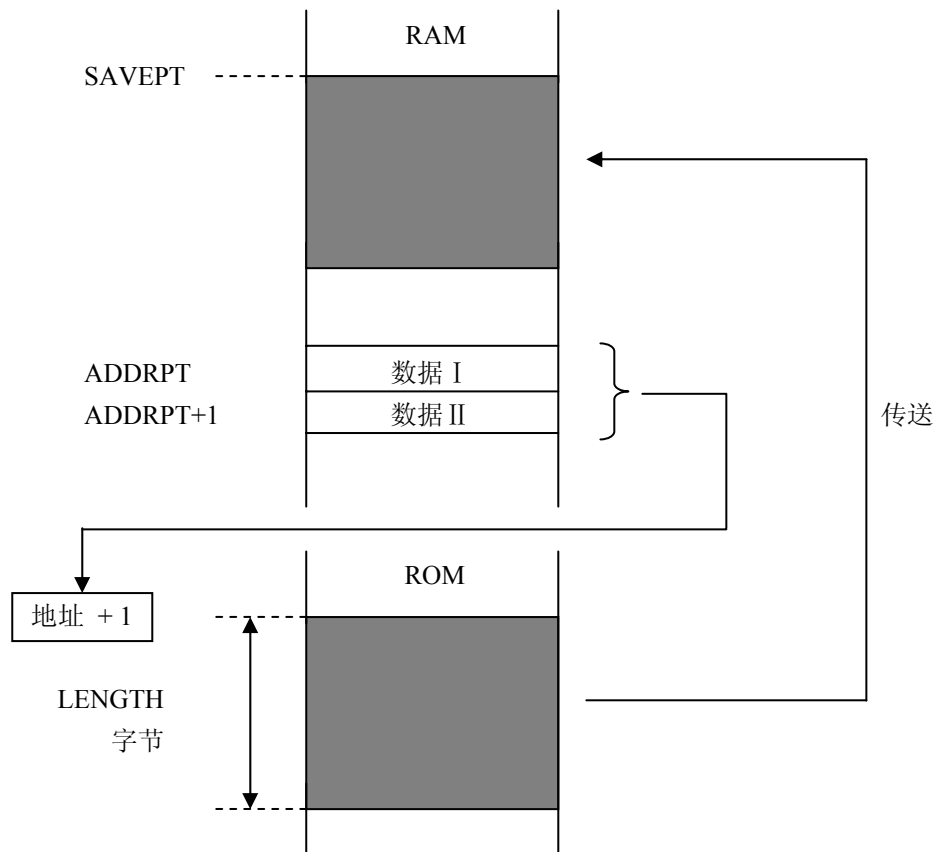
3.4 数据传送（可变 ROM 地址）

（1）概要

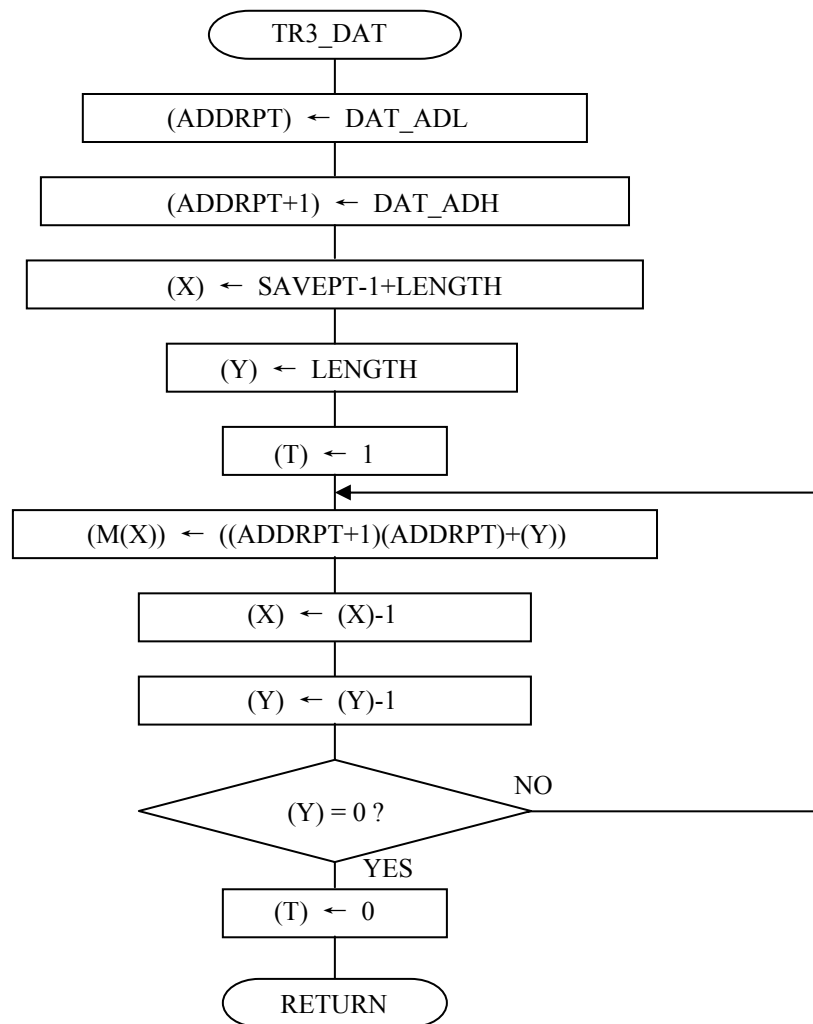
传送 ROM 区内的数据（可变地址）。

（2）说明

从 RAM 地址 ADDRPT+1 和 ADDRPT 的内容指向的 ROM 地址+1 开始将 LENGTH 字节的数据传送到 SAVEPT 开始的连续 RAM 地址。



(3) 流程图



注：M(X) 为变址寄存器 X 指向地址的存储器。

(4) 程序清单

```
;*****
;          ROM data transfer routine ( address float )
;*****
;
TR3_DAT:
        LDM    #DAT_ADL,ADDRPT
        LDM    #DAT_ADH,ADDRPT+1
;-----
        LDX    #SAVEPT-1+LENGTH
        LDY    #LENGTH          ;ROM length
        SET                     ;Transfer data from
TR3_01: LDA    (ADDRPT),Y        ; - (ADDRPT+1) (ADDRPT)+1
        DEX                     ; - to SAVEPT
        DEY                     ;
        BNE    TR3_01           ;Transfer end ?
        CLT                     ;Yes
        RTS
```

3.5 数据的重新排序

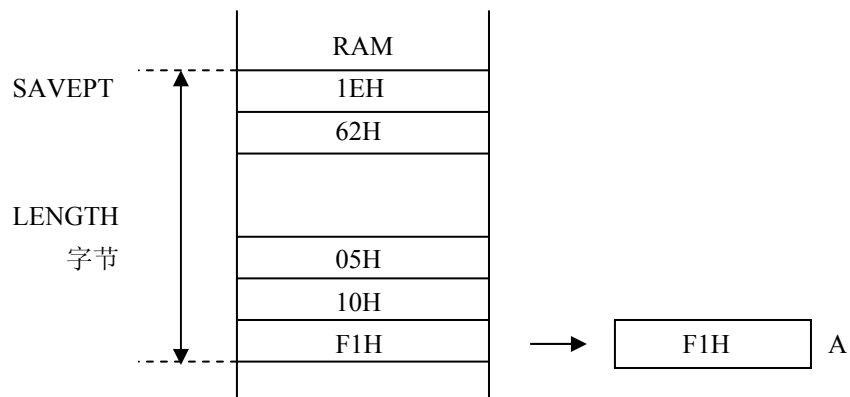
(1) 概要

按降序重新排列 RAM 区内的数据。

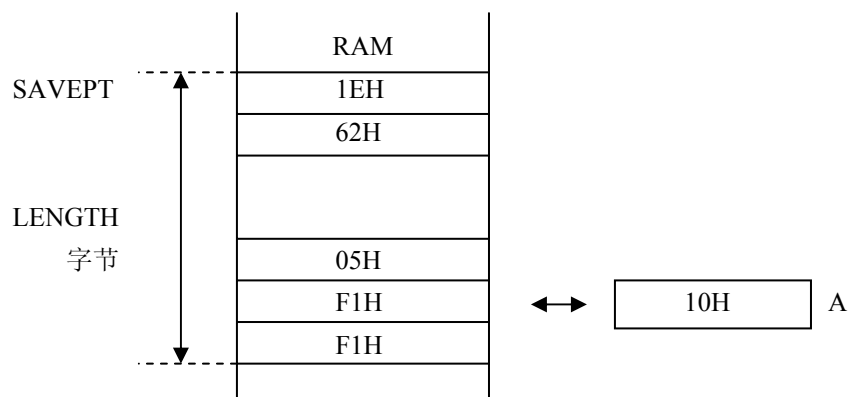
(2) 说明

从 RAM 地址 SAVEPT 按降序重新排列 LENGTH 字节的数据。

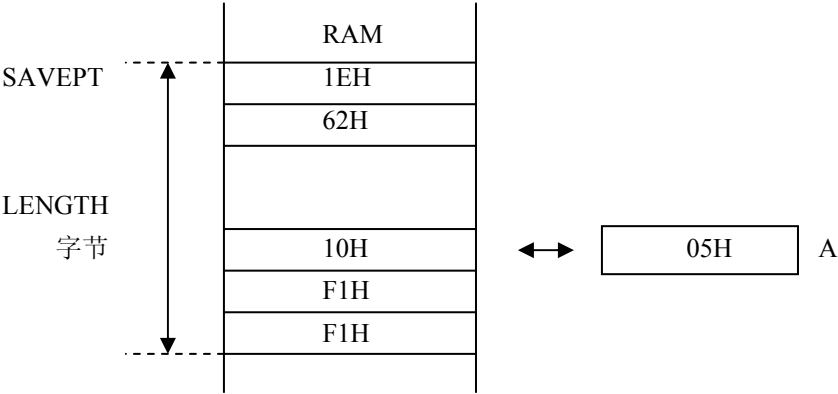
①首先，将要重新排序的区域最高位地址的存储器内容存入累加器 A。



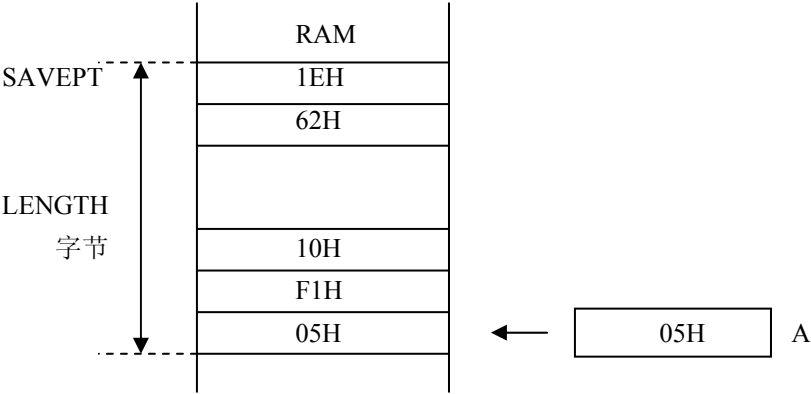
②然后，将下一位地址的存储器的内容和累加器 A 的内容进行比较。此时，如果累加器 A 的内容大于等于存储器的内容，就交换存储器和累加器 A 的内容。



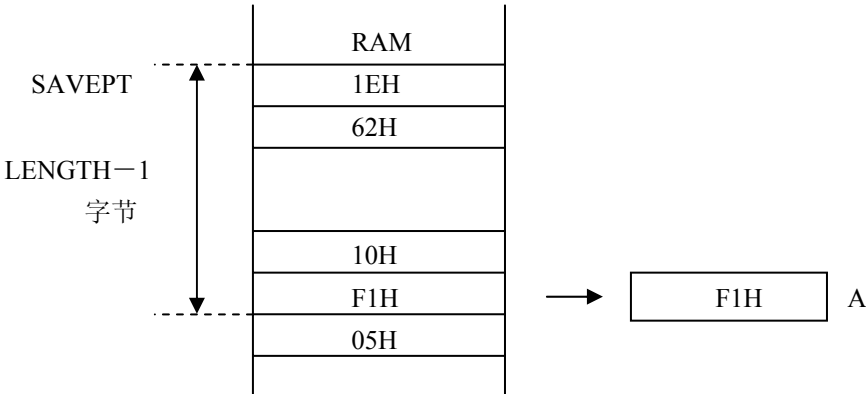
③接着，再将下一位地址的存储器的内容和累加器 A 的内容进行比较。此时，如果累加器 A 的内容大于等于存储器的内容，就交换存储器和累加器 A 的内容。



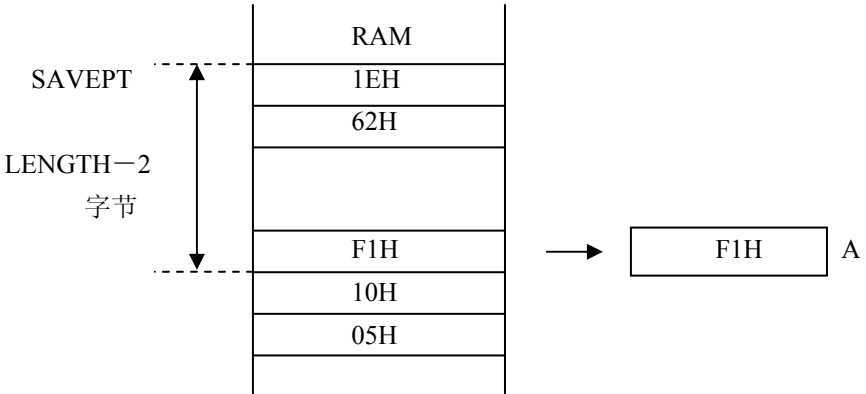
④按此，重复到要重新排序的区域的最低位地址为止，就能将最小值存入累加器 A，累加器 A 的内容存入最高位地址的存储器。



⑤在最高位地址递减的区域内（从 SAVEPT 的 LENGTH-1 字节）执行①~④。



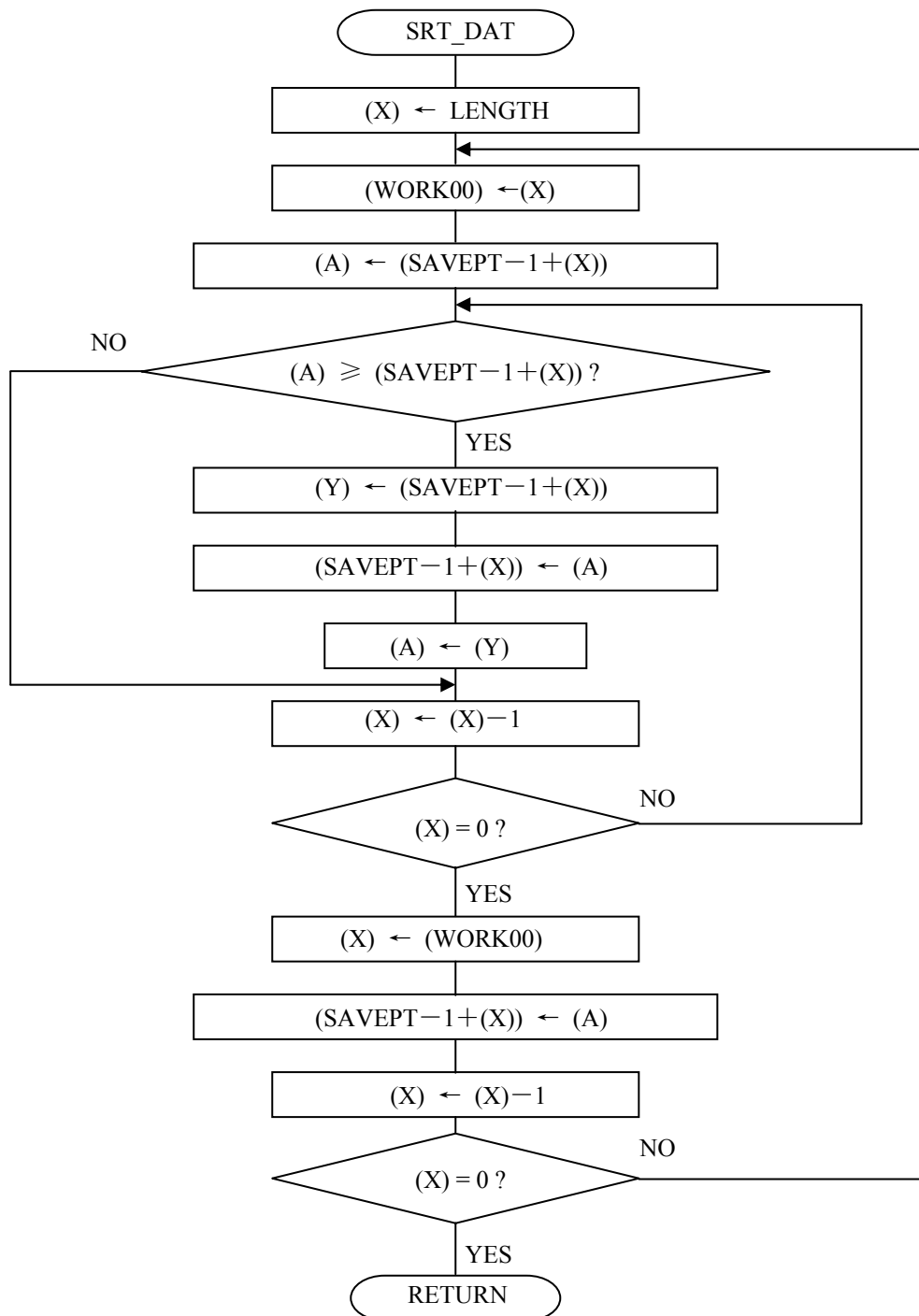
⑥接着，在最高位地址递减的区域内（从 SAVEPT 的 LENGTH-2 字节）执行①~④。



⑦按此，将最高位地址逐渐递减且执行①~④，直到最高位地址和最下位地址相等为止，就能按降序重新排列。

另外，只要在累加器 A 的内容小于存储器的内容时进行②③的存储器和累加器 A 的内容交换，就能按升序重新排序。

(3) 流程图



(4) 程序清单

```

;*****
;          Sort routine
;*****
;
SRT_DAT:
    LDX    #LENGTH          ;Data length
SRT_01:  STX    WORK00
    LDA    SAVEPT-1,X        ;SAVEPT <--> SAVEPT-1+WORK00
SRT_02:  CMP    SAVEPT-1,X    ;If use (BCS)
    BCC    SRT_03            ; - then negative
    LDY    SAVEPT-1,X
    STA    SAVEPT-1,X
    TYA
SRT_03:  DEX                ;Minimum data set
    BNE    SRT_02            ; - to A & Y
    LDX    WORK00
    STA    SAVEPT-1,X        ;Minimum data set
    DEX                ;Next area
    BNE    SRT_01            ;Sort end ?
    RTS                ;Yes

```

3.6 16 位数据加法运算（二进制）

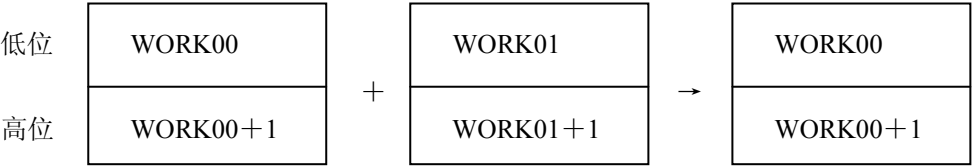
（1）概要

将 16 位二进制数据进行加法运算。

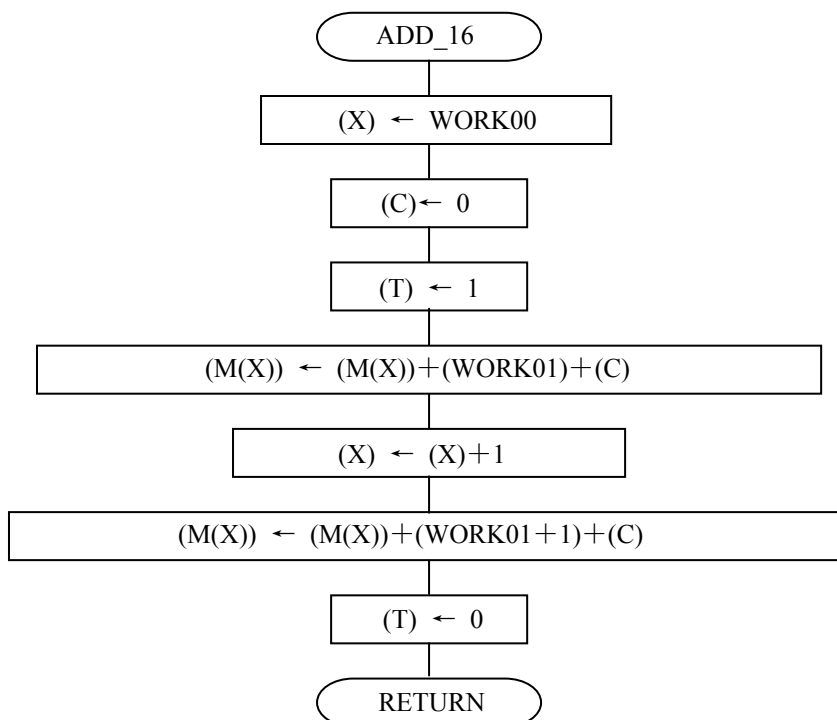
（2）说明

将 WORK00+1、WORK00 的内容和 WORK01+1、WORK01 的内容相加，结果存入 WORK00+1、WORK00。

将 X 变址模式标志 T 置 1，不破坏累加器 A 的内容进行运算。



(3) 流程图



注：M (X) 为变址寄存器 X 指向地址的存储器。

(4) 程序清单

```

;*****
;      16 bits BIN. data addition routine
;*****
;
ADD_16:
    LDX    #WORK00
    CLC                    ;C flag clear
    SET     T              ;T flag set
    ADC     WORK01
    INX                    ; (WORK00+1) (WORK00) +
    ADC     WORK01+1        ; (WORK01+1) (WORK01)
    CLT                    ;T flag clear
    RTS
  
```

3.7 16 位数据减法运算（二进制）

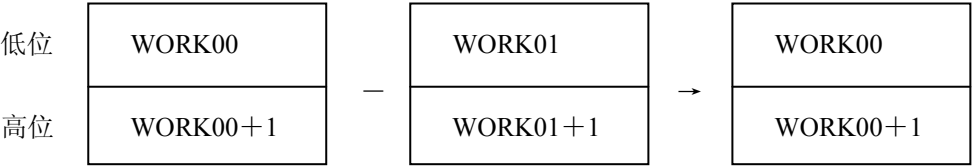
（1）概要

将 16 位二进制数据进行减法运算。

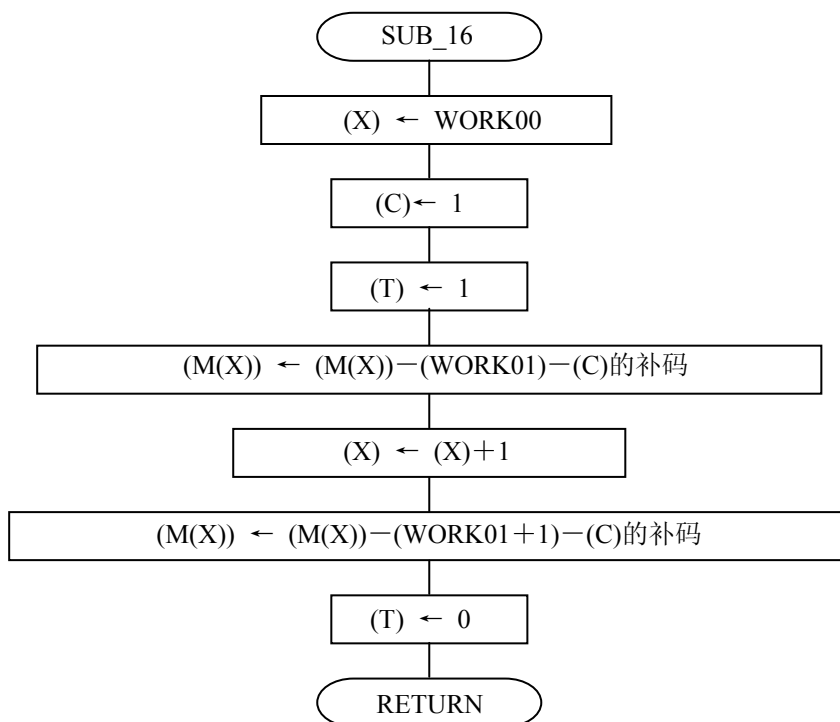
（2）说明

将 WORK00+1、WORK00 的内容减去 WORK01+1、WORK01 的内容，结果存入 WORK00+1、WORK00。

将 X 变址模式标志 T 置 1，不破坏累加器 A 的内容进行运算。



(3) 流程图



注：M (X) 为变址寄存器 X 指向地址的存储器。

(4) 程序清单

```

;*****
;      16 bits BIN. data subtraction routine
;*****
;
SUB_16:
    LDX    #WORK00
    SEC                      ;C flag set
    SET                      ;T flag set
    SBC    WORK01
    INX                      ; (WORK00+1) (WORK00) -
    SBC    WORK01+1          ; (WORK01+1) (WORK01)
    CLT                      ;T flag clear
    RTS
  
```

3.8 16 位数据加法运算（BCD）

（1）概要

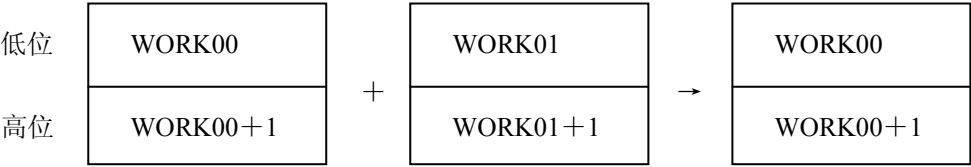
将 16 位 BCD 数据进行加法运算。

（2）说明

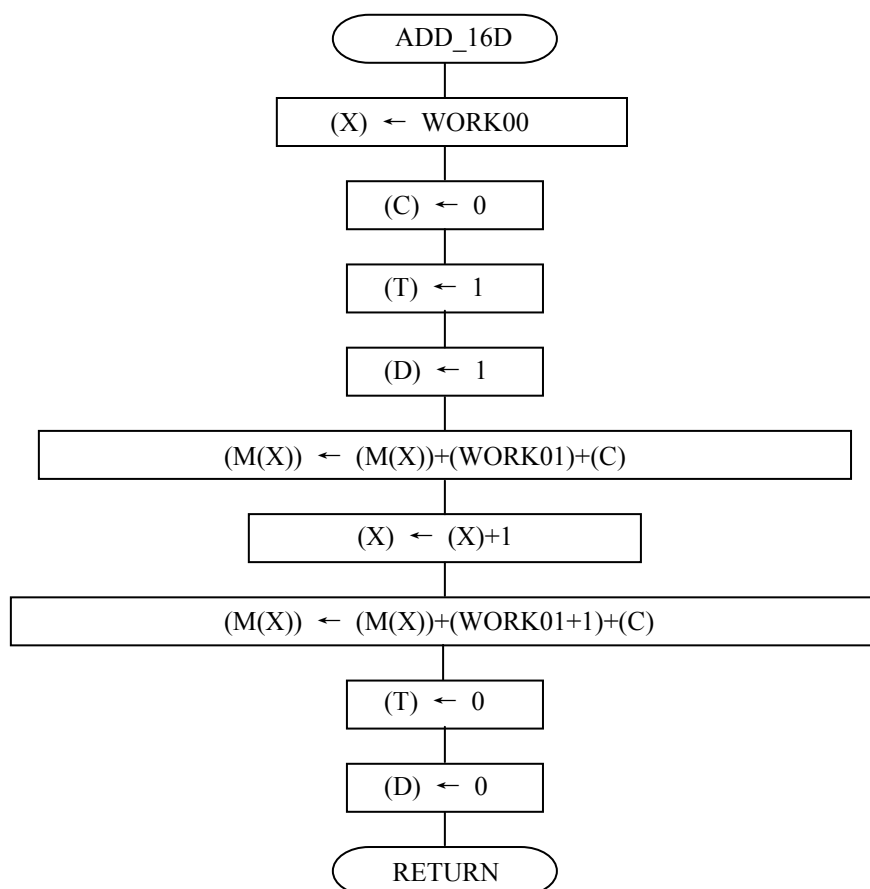
将 WORK00+1、WORK00 的内容和 WORK01+1、WORK01 的内容相加，结果存入 WORK00+1、WORK00。

通过将 10 进制模式标志 D 置 1，能使用 10 进制加法运算的 ADC 指令。但是，此时进位标志 C 的确定较慢，所以请不要紧接在 ADC 指令的后面执行 SEC、CLC、CLD 指令。

将 X 变址模式标志 T 置 1，不破坏累加器 A 的内容进行运算。



(3) 流程图



注：M (X) 为变址寄存器 X 指向地址的存储器。

(4) 程序清单

```

;*****
;      16 bits BCD data addition routine
;*****
;
ADD_16D:
    LDX    #WORK00
    CLC                    ;C flag reset
    SET                     ;T flag set
    SED                     ;Decimal mode set
    ADC     WORK01
    INX                    ; (WORK00+1) (WORK00) +
    ADC     WORK01+1        ; (WORK01+1) (WORK01)
    CLT                    ;T flag reset
    CLD                    ;Decimal mode clear
    RTS
  
```

3.9 16 位数据减法运算（BCD）

（1）概要

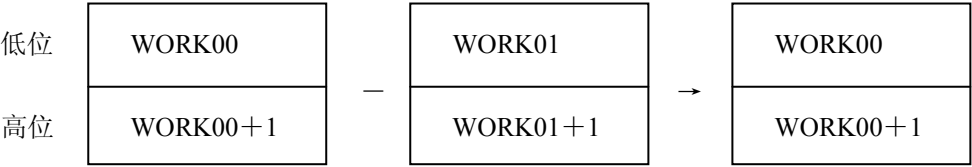
将 16 位 BCD 数据进行减法运算。

（2）说明

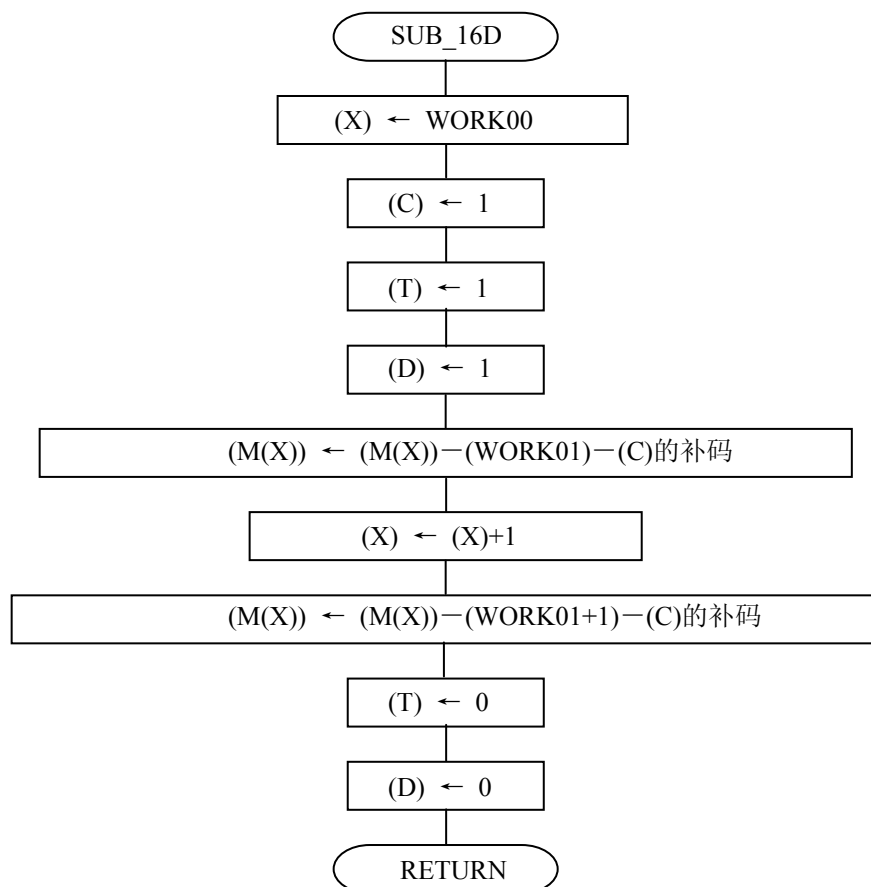
将 WORK00+1、WORK00 的内容减去 WORK01+1、WORK01 的内容，结果存入 WORK00+1、WORK00。

通过将 10 进制模式标志 D 置 1，能使用 10 进制加法运算的 SBC 指令。但是，此时进位标志 C 的确定较慢，所以请不要紧接在 SBC 指令的后面执行 SEC、CLC、CLD 指令。

将 X 变址模式标志 T 置 1，不破坏累加器 A 的内容进行运算。



(3) 流程图



注：M (X) 为变址寄存器 X 指向地址的存储器。

(4) 程序清单

```

;*****
;      16 bits BCD data subtraction routine
;*****
;
SUB_16D:
    LDX    #WORK00
    SEC                      ;C flag set
    SET                      ;T flag set
    SED                      ;Decimal mode set
    SBC    WORK01
    INX                      ; (WORK00+1) (WORK00) -
    SBC    WORK01+1          ; (WORK01+1) (WORK01)
    CLT                      ;T flag reset
    CLD                      ;Decimal mode reset
    RTS
  
```

3.10 16 位数据乘法运算（二进制）

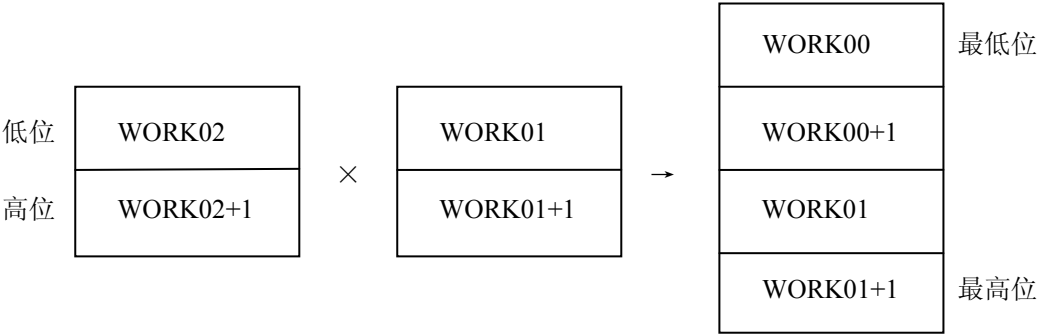
（1）概要

将 16 位二进制数据进行乘法运算。

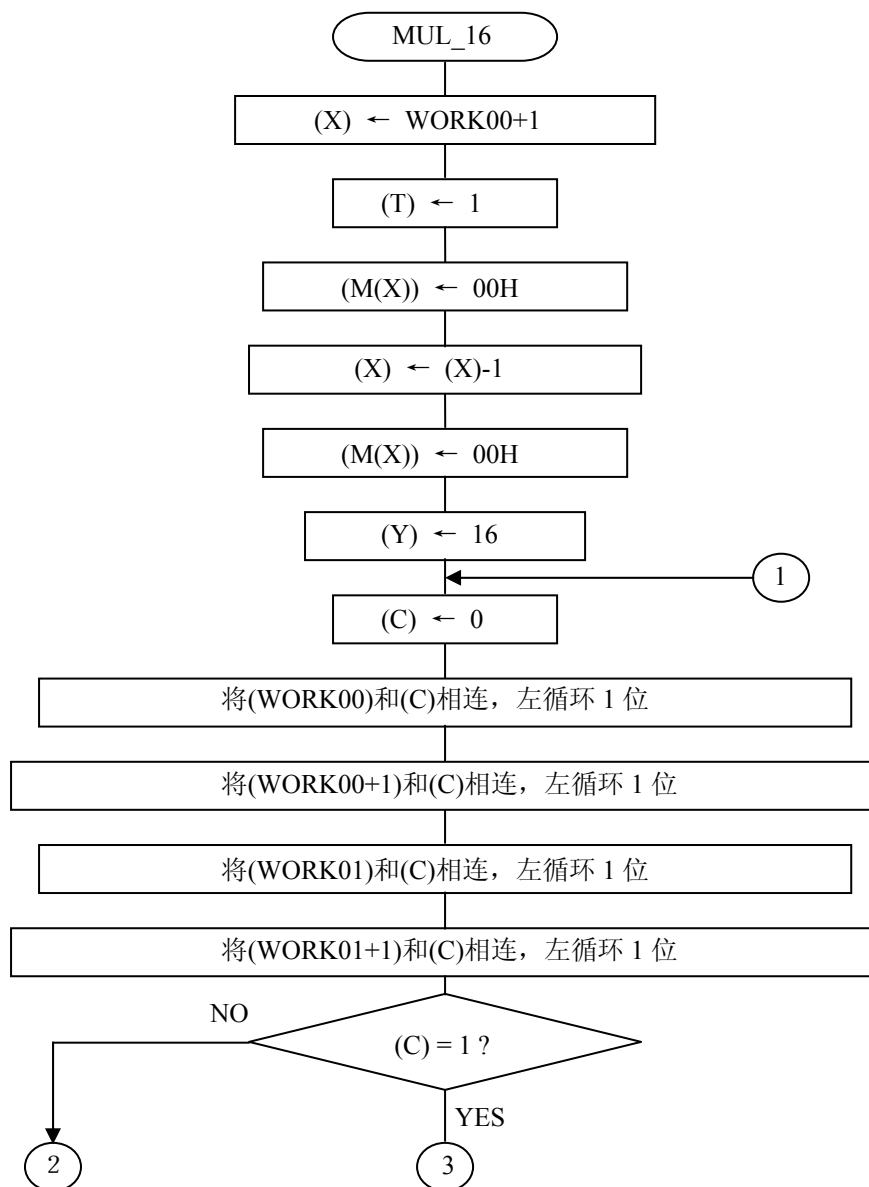
（2）说明

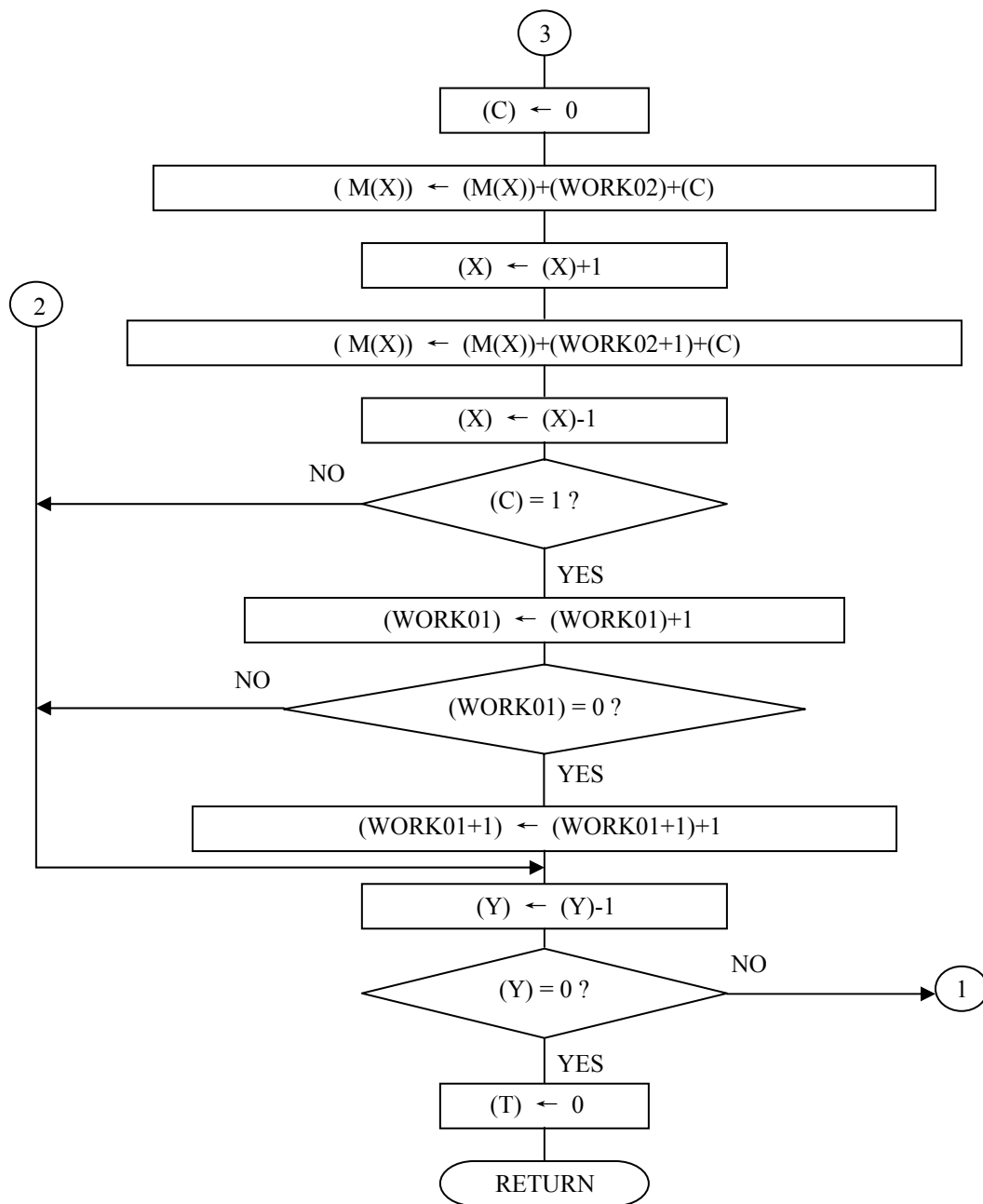
将 WORK02+1、WORK02 的内容和 WORK01+1、WORK01 的内容相乘，结果存入 WORK01+1、WORK01、WORK00+1、WORK00。

将 X 变址模式标志 T 置 1，不破坏累加器 A 的内容进行运算。



(3) 流程图





注：M (X) 为变址寄存器 X 指向地址的存储器。

(4) 程序清单

```
;*****
;      16 bits BIN. data multiplication routine
;*****
;
MUL_16:
    LDX    #WORK00+1      ;Product L addr. set
    SET    ;T flag set
    LDA    #$00           ;Clear product L
    DEX
    LDA    #$00
    LDY    #16            ;Bit counter set
;-----
MUL_01:  CLC
        ROL    WORK00      ;Rotate product L
        ROL    WORK00+1
        ROL    WORK01      ;Rotate product H
        ROL    WORK01+1
        BCC    MUL_02      ;C flag 1 ?
        CLC                ;Yes
        ADC    WORK02      ;Multiplicand + product L
        INX
        ADC    WORK02+1
        DEX
        BCC    MUL_02      ;Over flow ?
        INC    WORK01      ;Yes
        BNE    MUL_02      ;Over flow ?
        INC    WORK01+1    ;Yes
;-----
MUL_02:  DEY
        BNE    MUL_01      ;Multiple end ?
        CLT                ;Yes
        RTS
```

3.11 16 位数据除法运算（二进制）

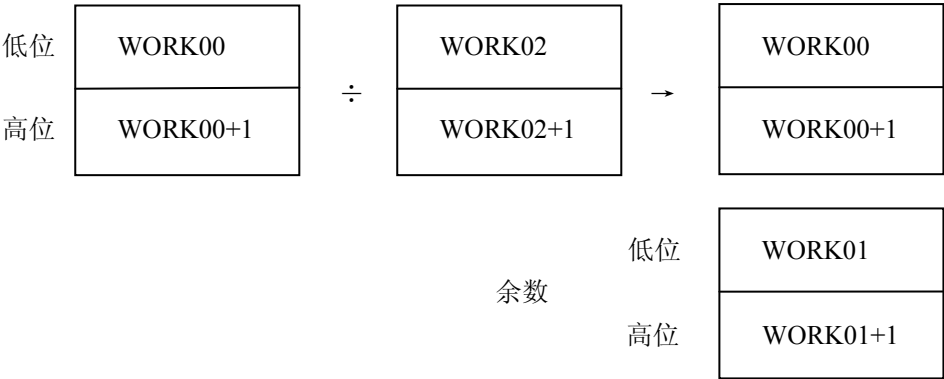
（1）概要

将 16 位二进制数据进行除法运算。

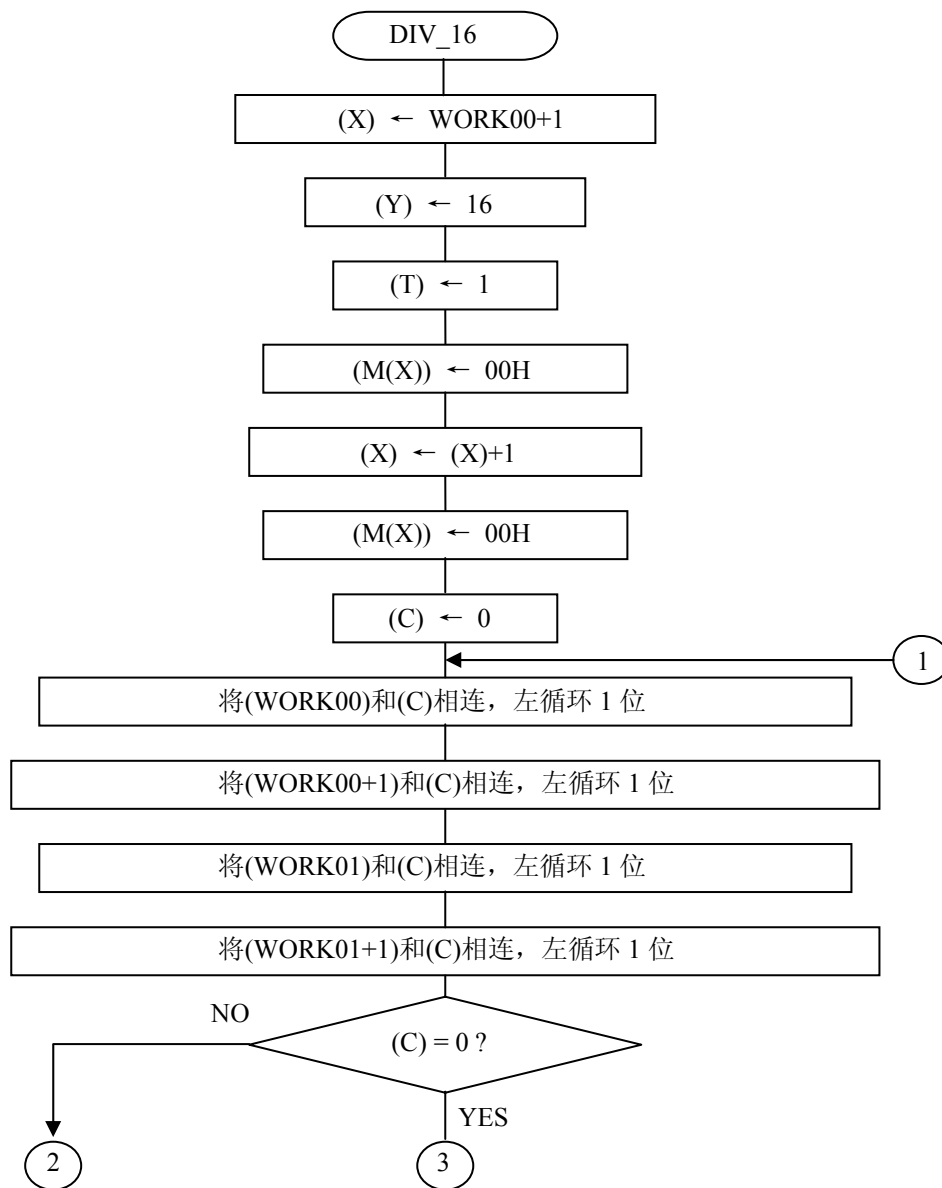
（2）说明

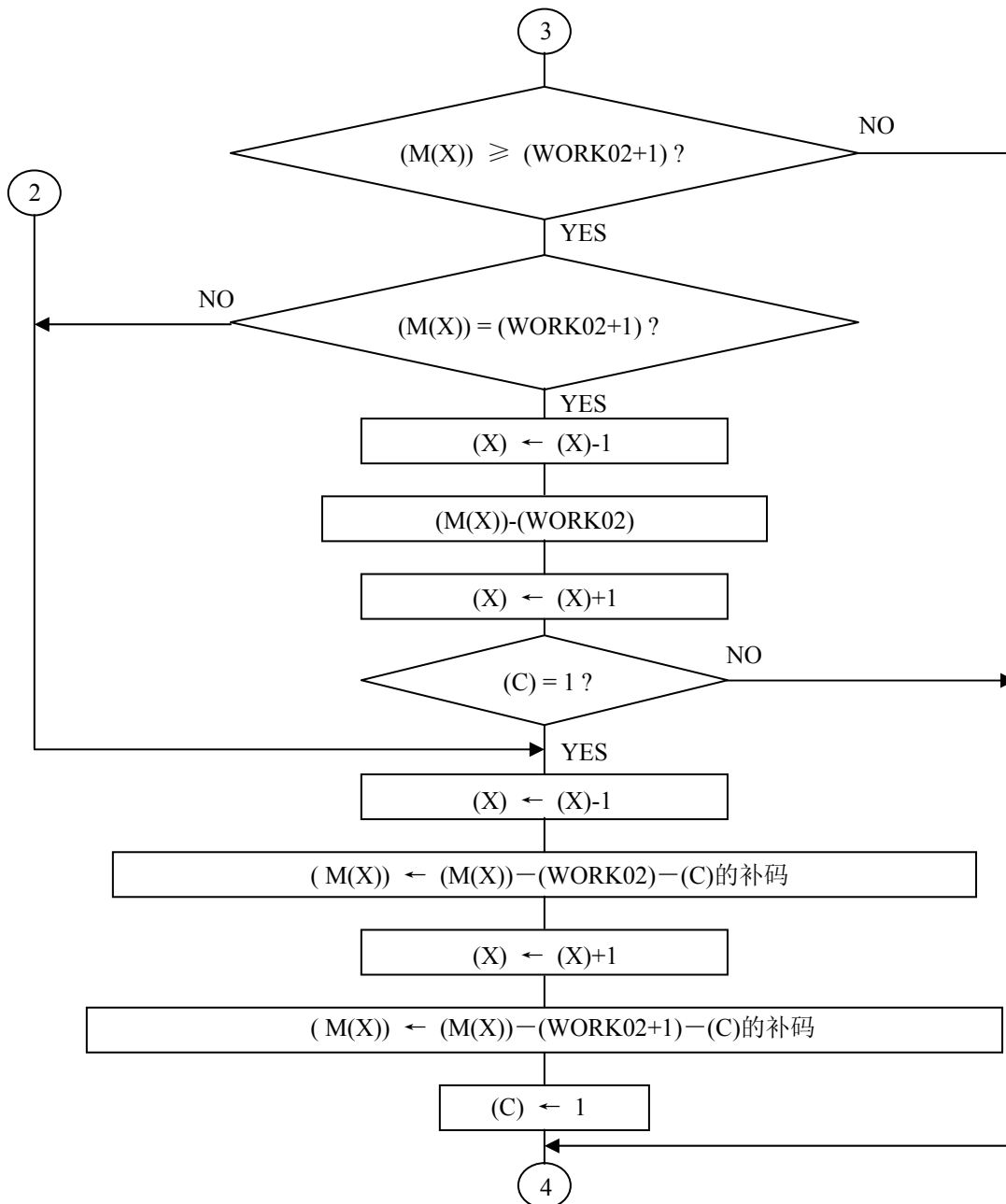
将 WORK00+1、WORK00 的内容除以 WORK02+1、WORK02 的内容，商存入 WORK00+1、WORK00，余数存入 WORK01+1、WORK01。

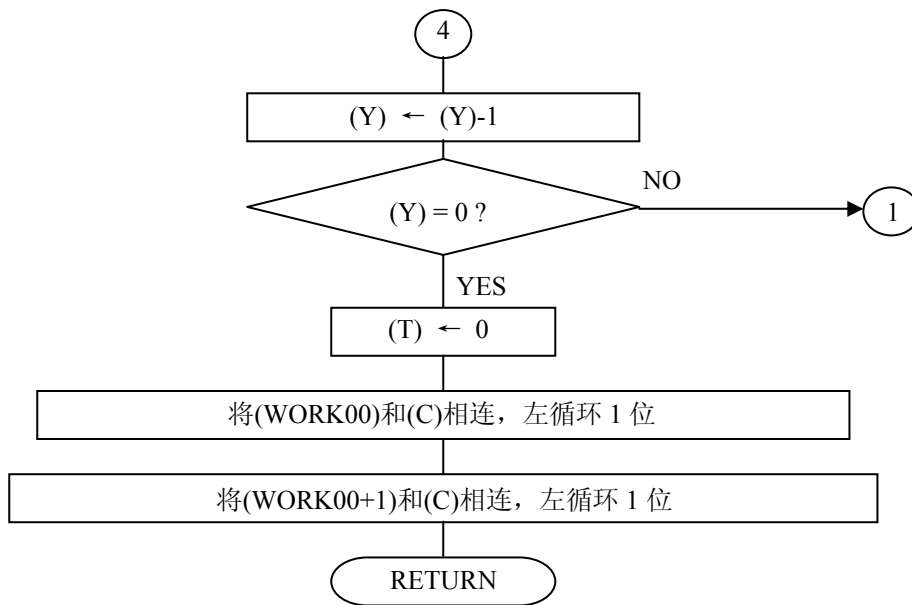
将 X 变址模式标志 T 置 1，不破坏累加器 A 的内容进行运算。



(3) 流程图







注：M (X) 为变址寄存器 X 指向地址的存储器。

(4) 程序清单

```

;*****
;          16 bits BIN. data division routine
;*****
;
DIV_16:
    LDX    #WORK01          ;Surplus addr. set
    LDY    #16              ;Bit counter set
    SET    T                ;T flag set
    LDA    #$00             ;Clear surplus
    INX
    LDA    #$00
    CLC
DIV_01:  ROL    WORK00          ;Rotate quotient
    ROL    WORK00+1
    ROL    WORK01          ;Rotate surplus
    ROL    WORK01+1
    BCS    DIV_02            ;C flag 1 ?
;-----
    CMP    WORK02+1          ;No
    BCC    DIV_03            ;Cannot divide ?
    BNE    DIV_02            ;No
    DEX
    CMP    WORK02
    INX
    BCC    DIV_03            ;Cannot divide ?
;-----
DIV_02:  DEX                ;No
    SBC    WORK02            ;Surplus - divisor
    INX
    SBC    WORK02+1
    SEC
DIV_03:  DEY
    BNE    DIV_01            ;Divide end ?
;-----
    CLT                    ;Yes
    ROL    WORK00            ;Rotate quotient
    ROL    WORK00+1
    RTS

```

4. 应用程序例子

4.1 文件处理（传送）

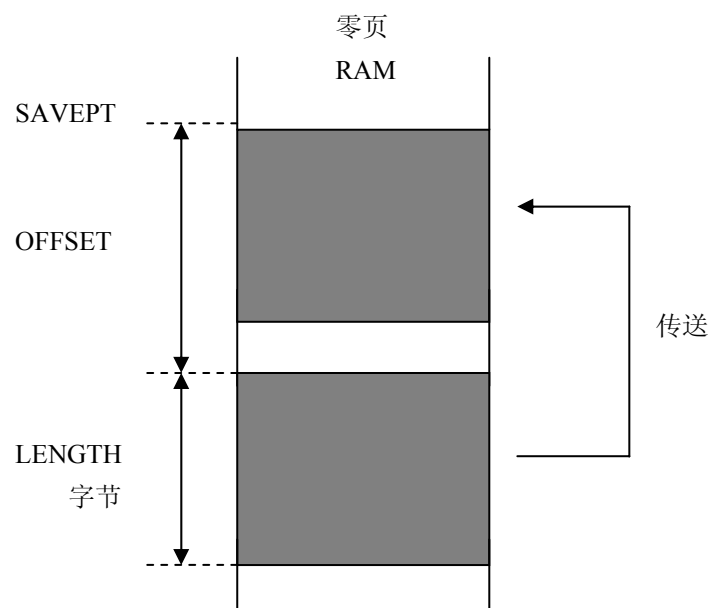
（1）概要

将零页 RAM 部分视为文件存储器，进行数据传送处理。

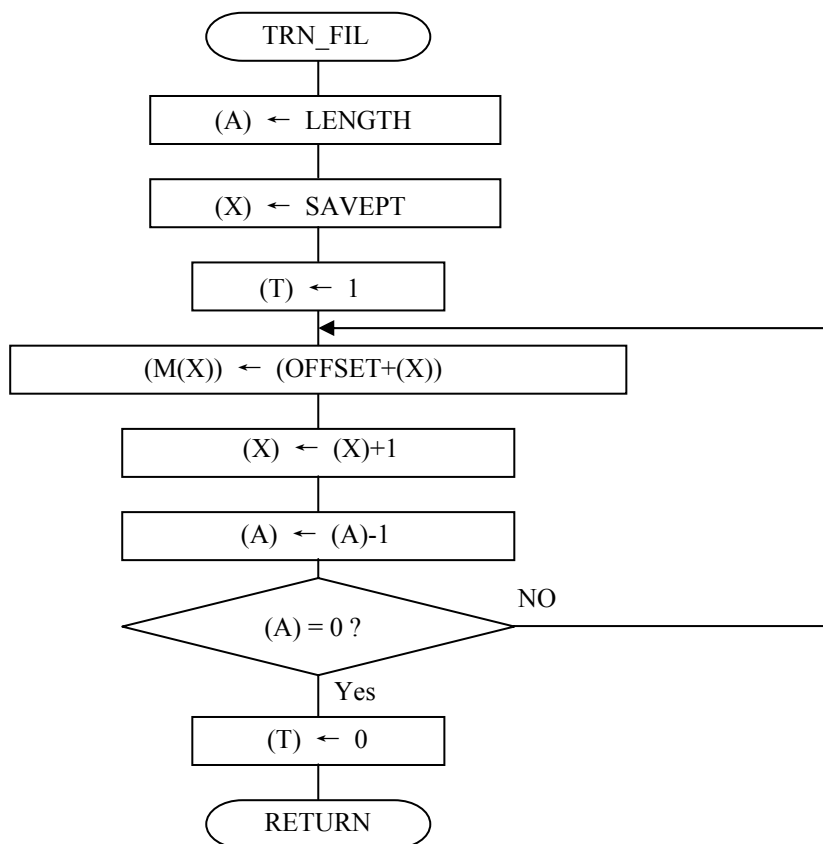
（2）说明

从零页 RAM 地址 SAVEPT+OFFSET 开始将 LENGTH 字节的文件存储器数据传送到 SAVEPT 开始的连续文件存储器。

将 X 变址模式标志 T 置位，进行存储器间的传送。另外，指针并非双指针（源/目标）而为单指针+偏移量。



(3) 流程图



注：M (X) 为变址寄存器 X 指向地址的存储器。

(4) 程序清单

```

;*****
;      File handling (transfer)
;*****
;
TRN_FIL:
    LDA    #LENGTH          ;File length
    LDX    #SAVEPT
    SET    T                 ;T flag set
TRN_01:
    LDA    OFFSET,X         ;Transfer data from
    INX                    ; - SAVEPT + OFFSET
    DEC    A                ; - to SAVEPT
    BNE    TRN_01           ;Transfer end ?
    CLT
    RTS                    ;Yes
  
```

4.2 文件处理（交换）

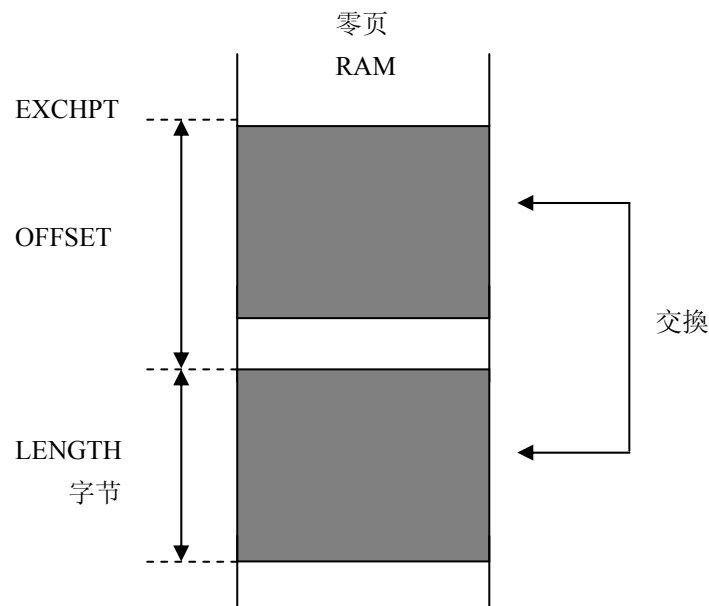
（1）概要

将零页 RAM 部分视为文件存储器，进行数据交换处理。

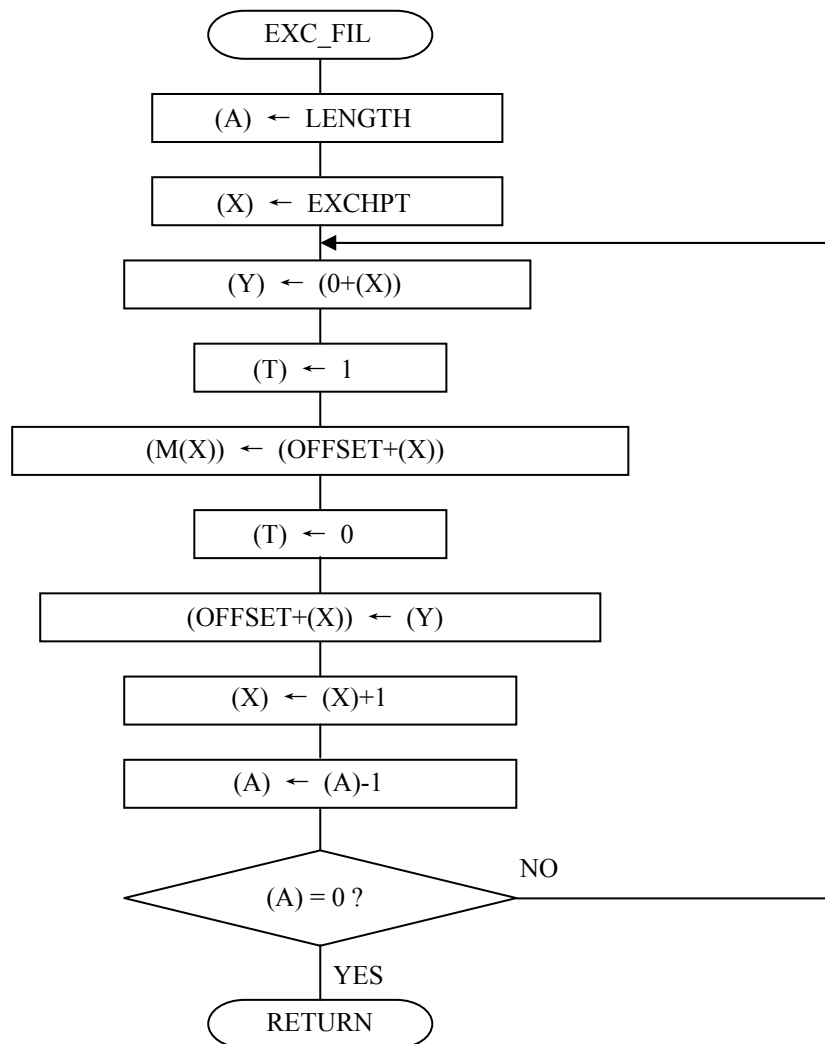
（2）说明

将零页 RAM 地址 EXCHPT+OFFSET 开始的 LENGTH 字节的文件存储器数据和 EXCHPT 开始的 LENGTH 字节的文件存储器数据进行交换。

将 X 变址模式标志 T 置位，进行存储器间的传送。另外，指针并非双指针（源/目标）而为单指针+偏移量。



(3) 流程图

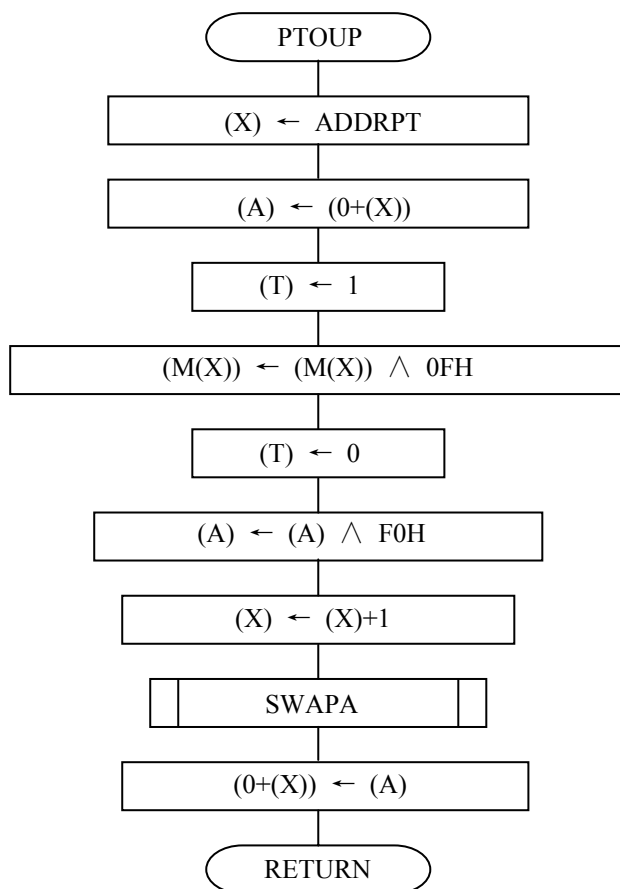


注：M (X) 为变址寄存器 X 指向地址的存储器。

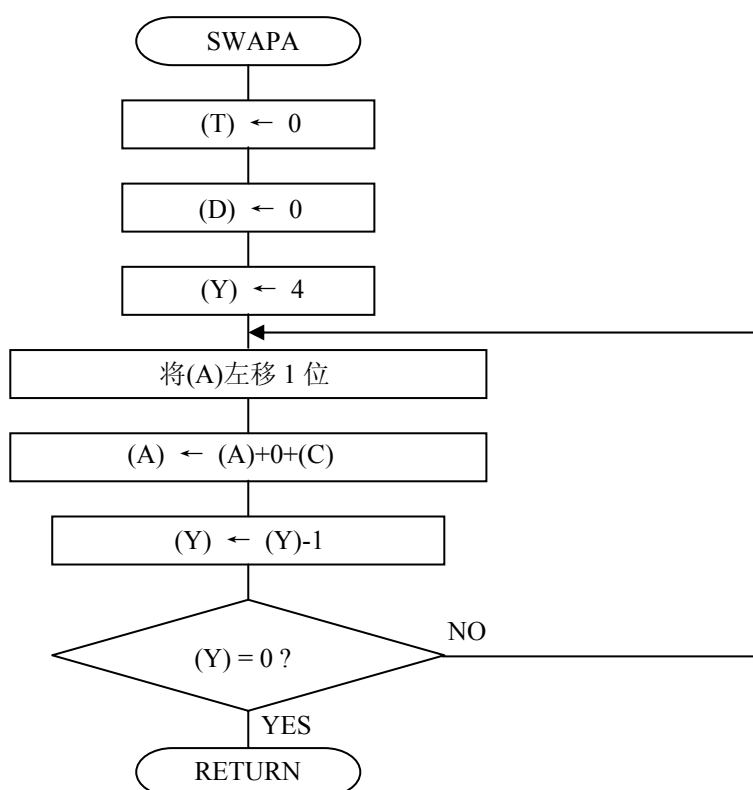
(4) 程序清单

```
;*****
;      File handling (exchange)
;*****
;
EXC_FIL:
    LDA    #LENGTH          ;File length
    LDX    #EXCHPT
EXC_01:
    LDY    0,X              ;Exchange data of
    SET                     ; - EXCHPT + OFFSET
    LDA    OFFSET,X         ; - with EXCHPT
    CLT
    STY    OFFSET,X
    INX
    DEC    A
    BNE    EXC_01           ;Exchange end ?
    RTS                    ;Yes
```


(3) 流程图



注：M (X) 为变址寄存器 X 指向地址的存储器。



(4) 程序清单

```
;*****
;          Packed BCD -> unpacked BCD
;*****
;
PTOUP:
    LDX    #ADDRPT
    LDA    0,X                ;Get packed BCD data
    SET                     ;T flag set
    AND    #0FH              ;Unpacked BCD data L
    CLT                     ;T flag clear
    AND    #0F0H
    INX
    JSR    SWAPA              ;Swap A
    STA    0,X                ;Unpacked BCD data H
    RTS

;*****
;          Swap A register
;*****
;
SWAPA:
    CLT                     ;T flag clear
    CLD                     ;Decimal mode clear
    LDY    #4
SWAPA1:
    ASL    A
    ADC    #0
    DEY
    BNE    SWAPA1
    RTS
```

4.4 码转换（非压缩 BCD→非压缩 BCD）

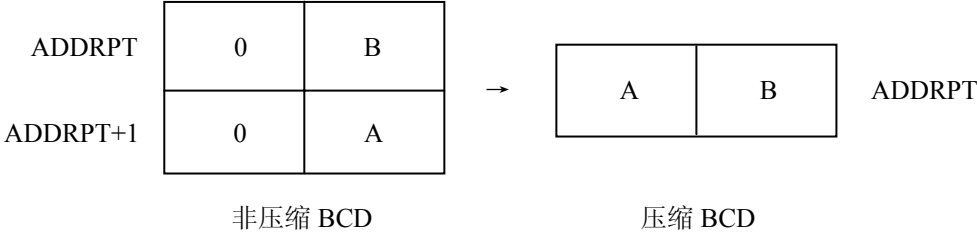
（1）概要

将非压缩 BCD 数据转换成压缩 BCD 数据。

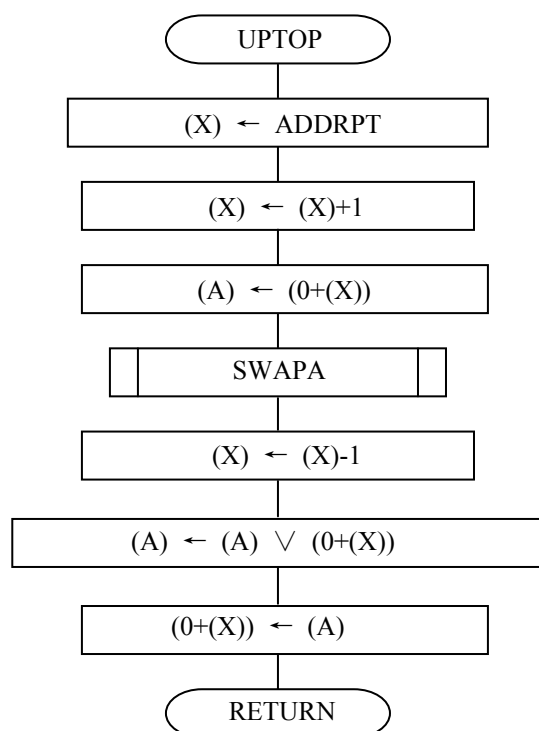
（2）说明

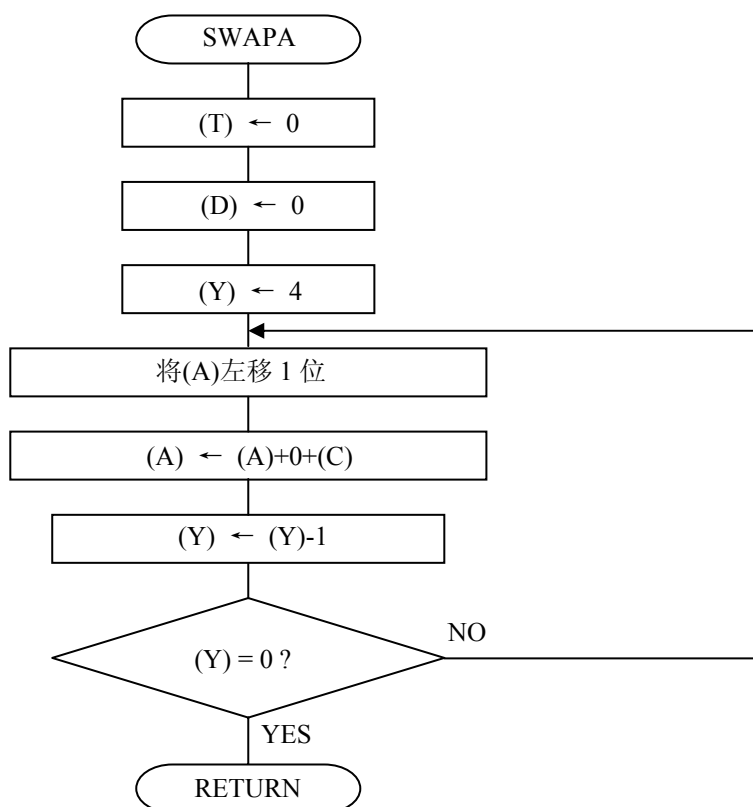
将零页 RAM 地址 ADDRPT+1 和 ADDRPT 的非压缩 BCD 数据转换成压缩 BCD，存入 ADDRPT。

如下所示，压缩 BCD 是将 2 位 BCD 数值压缩成 1 字节的数据。另外，非压缩 BCD 是将压缩 BCD 分成高 4 位为 0 的 2 字节数据。



(3) 流程图





(4) 程序清单

```

;*****
;      Unpacked BCD -> packed BCD
;*****
;
UPTOP:
    LDX    #ADDRPT
    INX
    LDA    0,X      ;Get unpacked BCD data H
    JSR    SWAPA     ;Swap A
    DEX
    ORA    0,X      ;Packed BCD data
    STA    0,X
    RTS

;*****
;      Swap A register
;*****
;
SWAPA:
    CLT                ;T flag clear
    CLD                ;Decimal mode clear
    LDY    #4
SWAPA1:
    ASL    A
    ADC    #0
    DEY
    BNE    SWAPA1
    RTS

```

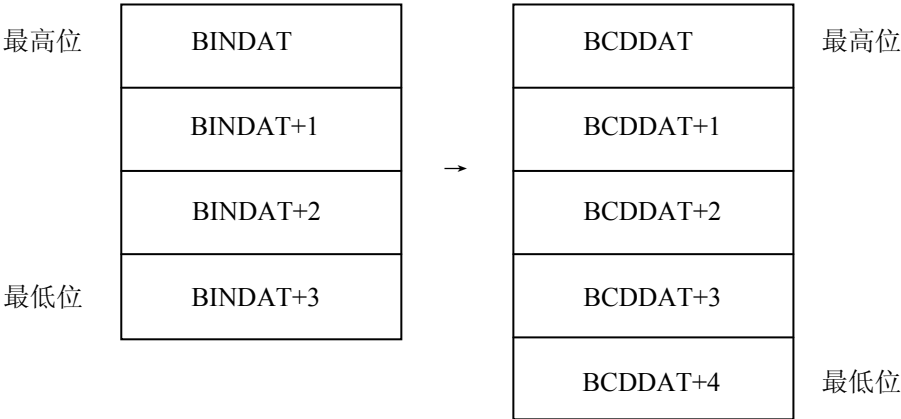
4.5 码转换 (BIN→BCD)

(1) 概要

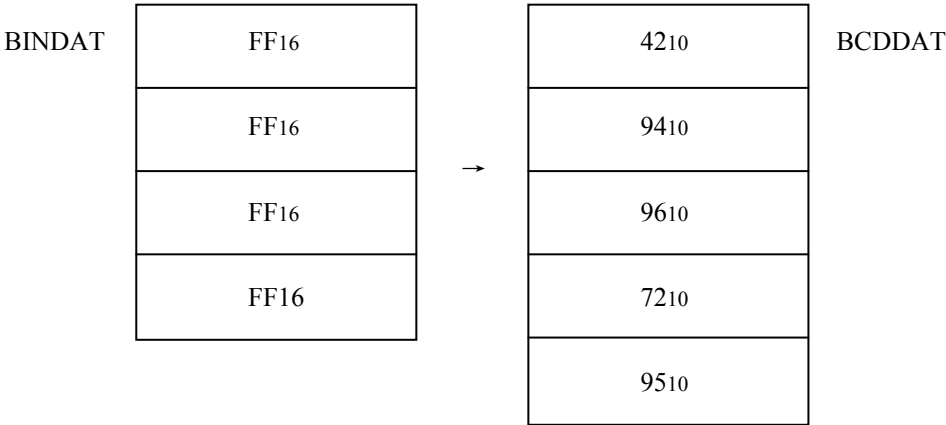
将 4 字节 BIN 数据转换成 5 字节 BCD 数据。

(2) 说明

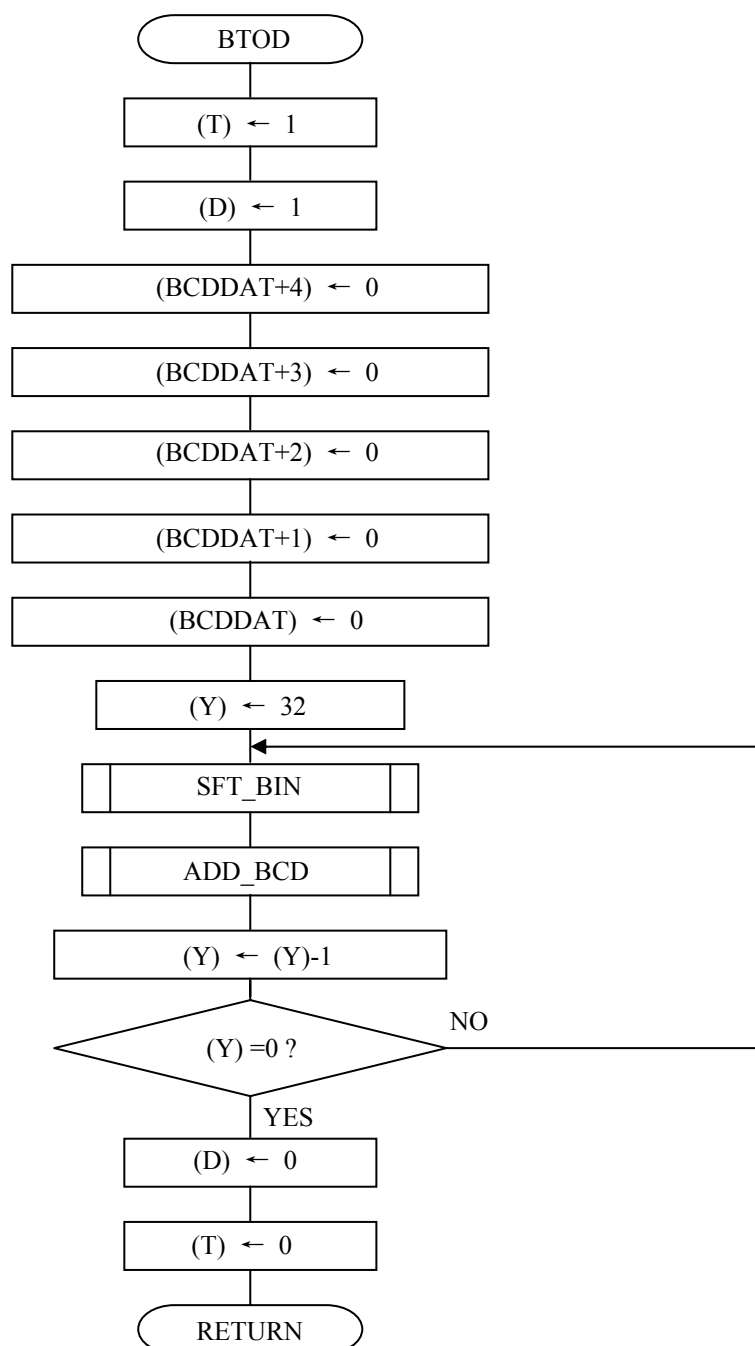
将零页 RAM 地址 BINDAT、BINDAT+1、BINDAT+2、BINDAT+3 的 4 字节 BIN 数据转换成 5 字节的 BCD 数据，存入 BCDDAT、BCDDAT+1、BCDDAT+2、BCDDAT+3、BCDDAT+4。

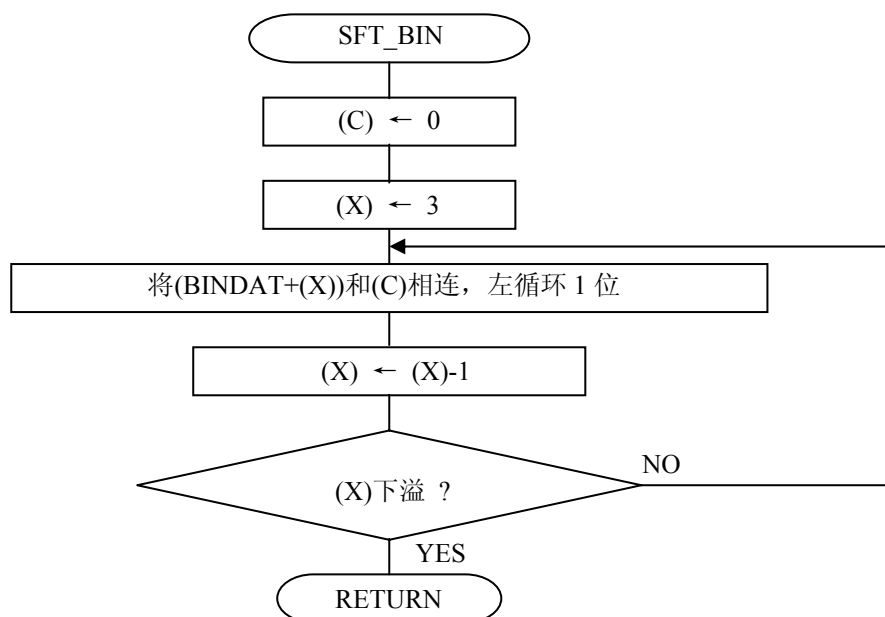


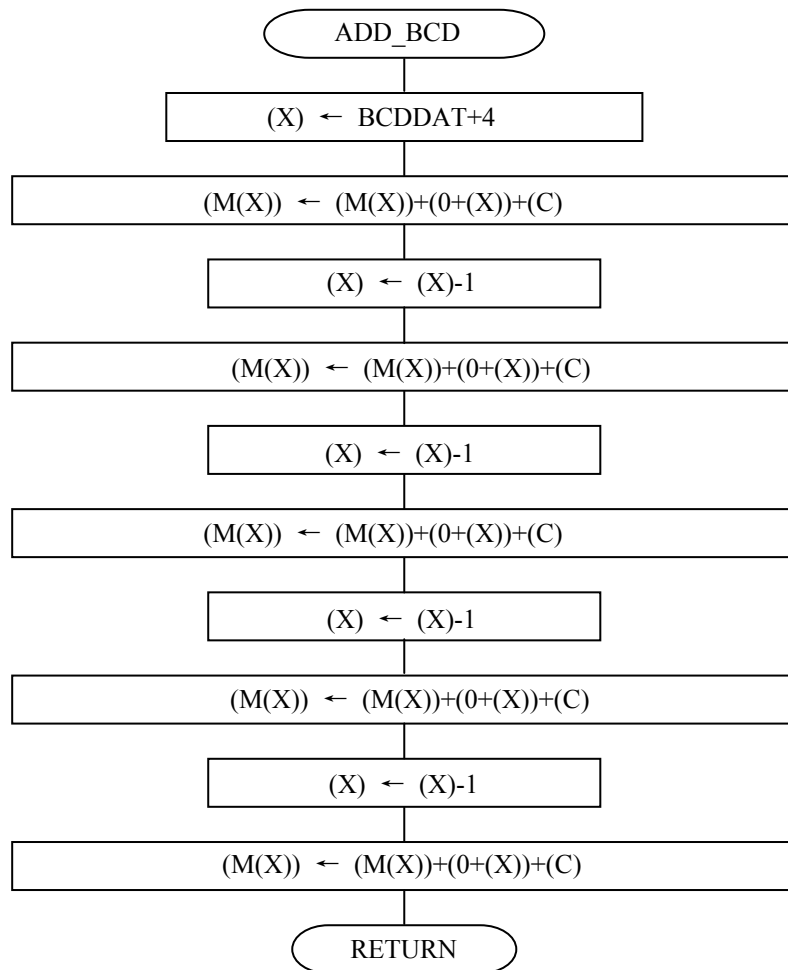
例) 在 BIN 数据为 FFFFFFFFH 的情况下，BCD 数据变为 4294967295。



(3) 流程图







注：M (X) 为变址寄存器 X 指向地址的存储器。

(4) 程序清单

```
;*****
;          BIN -> BCD
;*****
;
BTOD:
    SET                ;T flag set
    SED                ;Decimal mode set
    LDM    #0,BCDDAT+4 ;Clear BCD result area
    LDM    #0,BCDDAT+3
    LDM    #0,BCDDAT+2
    LDM    #0,BCDDAT+1
    LDM    #0,BCDDAT
    LDY    #32          ;Yes
BTOD_01:
    JSR    SFT_BIN      ;Left shift BIN data
    JSR    ADD_BCD      ;2*(BCD)+C -> (BCD)
    DEY
    BNE    BTOD_01      ;Convert end ?
    CLD
    CLT                ;T flag clear
    RTS
;*****
;          Left shift BIN data
;*****
;
SFT_BIN:
    CLC                ;C flag clear
    LDX    #3
SFT_01:
    ROL    BINDAT,X
    DEX
    BPL    SFT_01      ;Shift end ?
    RTS                ;Yes
;*****
;          2*(BCD)+C -> (BCD)
;*****
;
ADD_BCD:
    LDX    #BCDDAT+4
    ADC    0,X
    DEX
    ADC    0,X
    DEX
    ADC    0,X
    DEX
    ADC    0,X
    DEX
    ADC    0,X
    RTS
```

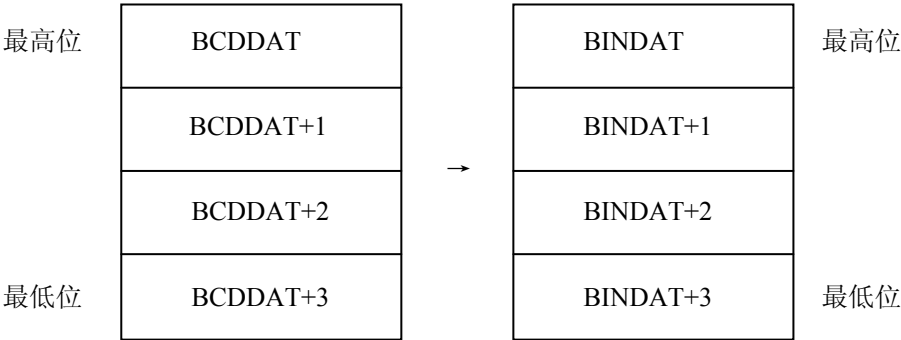
4.6 码转换（BCD→BIN）

（1）概要

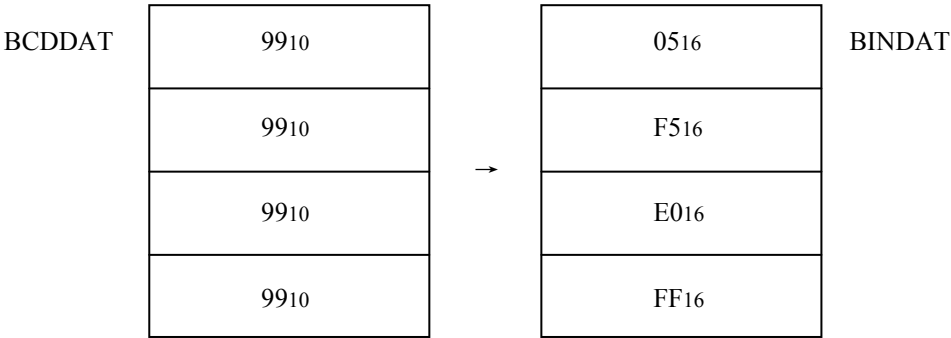
将 4 字节 BCD 数据转换成 4 字节 BIN 数据。

（2）说明

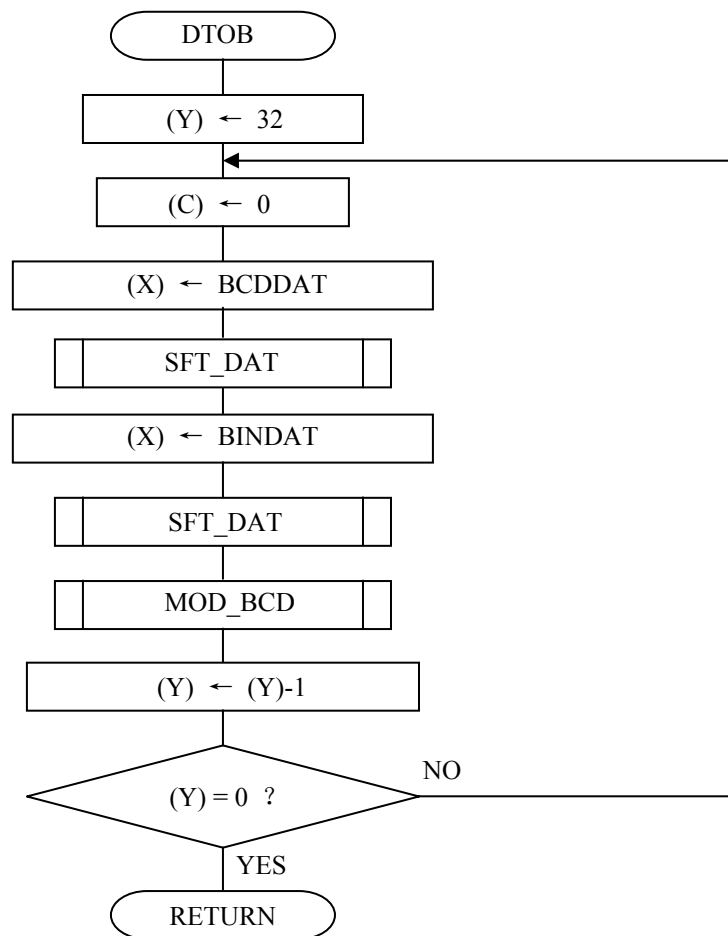
将零页 RAM 地址 BCDDAT、BCDDAT+1、BCDDAT+2、BCDDAT+3 的 4 字节 BCD 数据转换成 4 字节的 BIN 数据，存入 BINDAT、BINDAT+1、BINDAT+2、BINDAT+3。

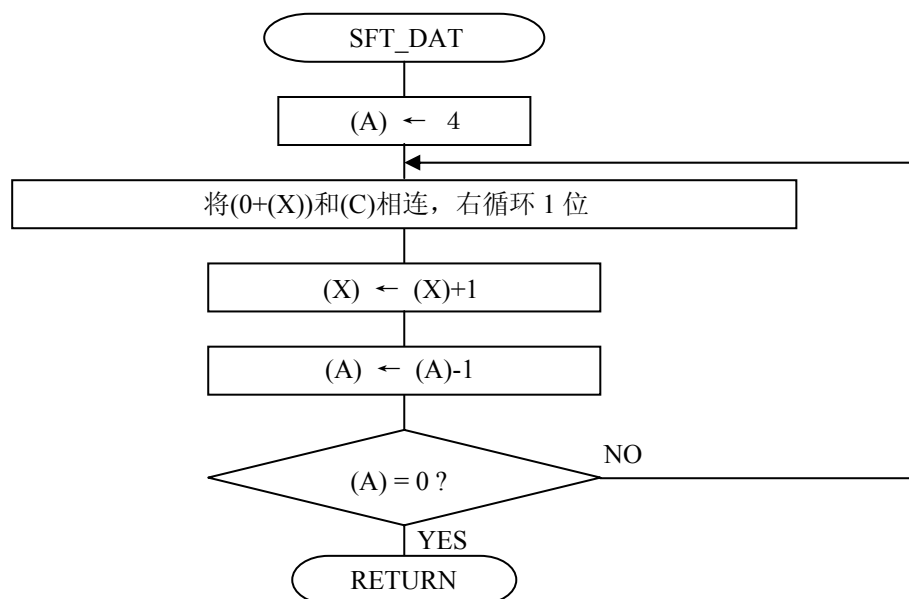


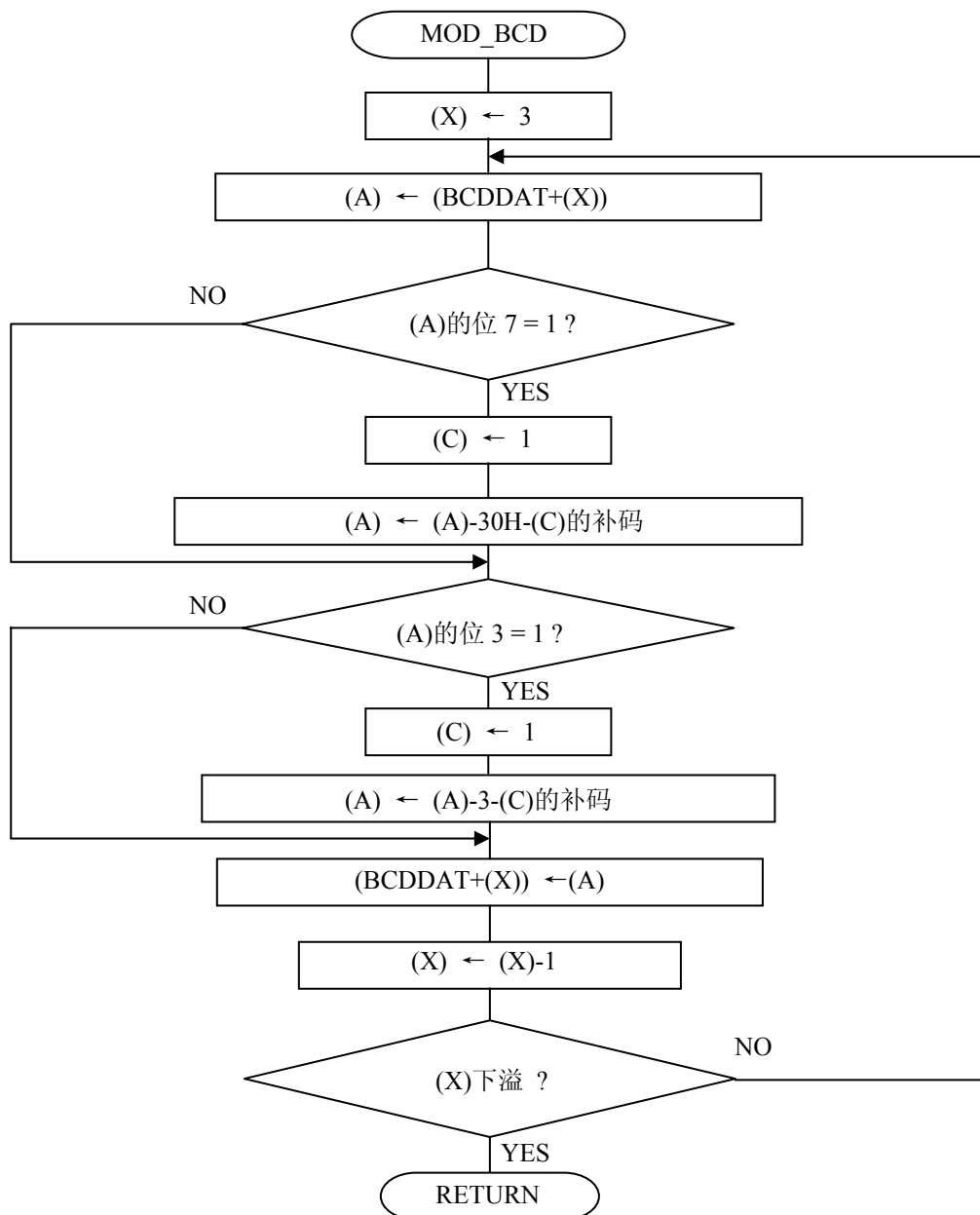
例）在 BCD 数据为 99999999 的情况下，BIN 数据变为 05F5E0FFH。



(3) 流程图







(4) 程序清单

```
;*****
;          BCD -> BIN
;*****
;
DT0B:
        LDY    #32
DT0B_01:
        CLC
        LDX    #BCDDAT           ;Point to BCDDAT
        JSR    SFT_DAT           ;Right shift BCD data
        LDX    #BINDAT          ;Point to BINDAT
        JSR    SFT_DAT           ;Right shift BIN data
        JSR    MOD_BCD          ;Get modified BCD data
        DEY
        BNE    DT0B_01          ;Shift end ?
        RTS                    ;Yes !
;*****
;          Right shift data
;*****
;
SFT_DAT:
        LDA    #4
SFT_02:
        ROR    0,X
        INX
        DEC    A
        BNE    SFT_02
        RTS
;*****
;          Modify BCD data
;*****
;
MOD_BCD:
        LDX    #3
MOD_01:
        LDA    BCDDAT,X
        BBC    7,A,MOD_02
        SEC
        SBC    #30H
MOD_02:
        BBC    3,A,MOD_03
        SEC
        SBC    #3
MOD_03:
        STA    BCDDAT,X
        DEX
        BPL    MOD_01
        RTS
```

4.7 SGN 函数

(1) 概要

是累加器 A 的 SGN 函数。

(2) 说明

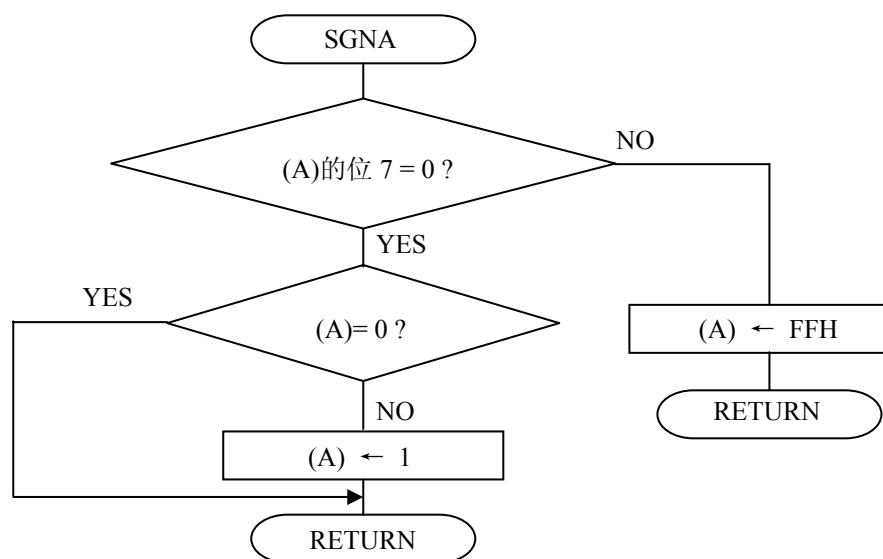
求累加器 A 的内容的 SGN 值，其结果存入累加器 A。累加器 A 的位 7 为符号位。此时的 SGN 函数如下：

(A)>0 时 SGN(A)=1

(A)=0 时 SGN(A)=0

(A)<0 时 SGN(A)=-1

(3) 流程图



(4) 程序清单

```

;*****
;      SGN(A)
;*****
;
SGNA:
    BBS    7,A,SGN_01
    CMP    #0
    BEW    SGN_02
    LDA    #1

SGN_02:
    RTS

SGN_01:
    LDA    #$FF
    RTS
  
```

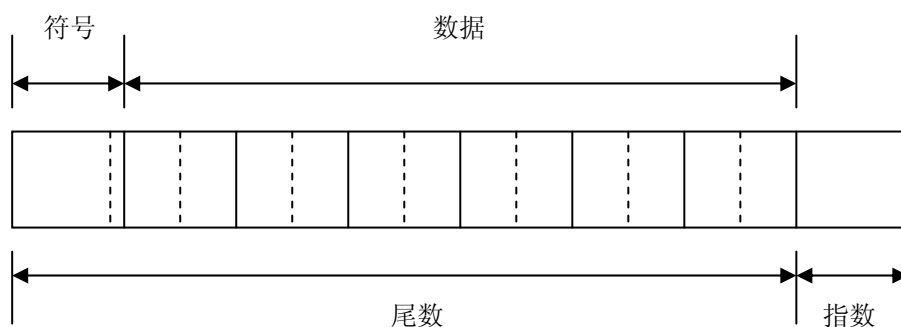
4.8 DCB 12 位浮点四则运算

(1) 概要

进行 BCD 12 位的浮点四则运算。

(2) 说明

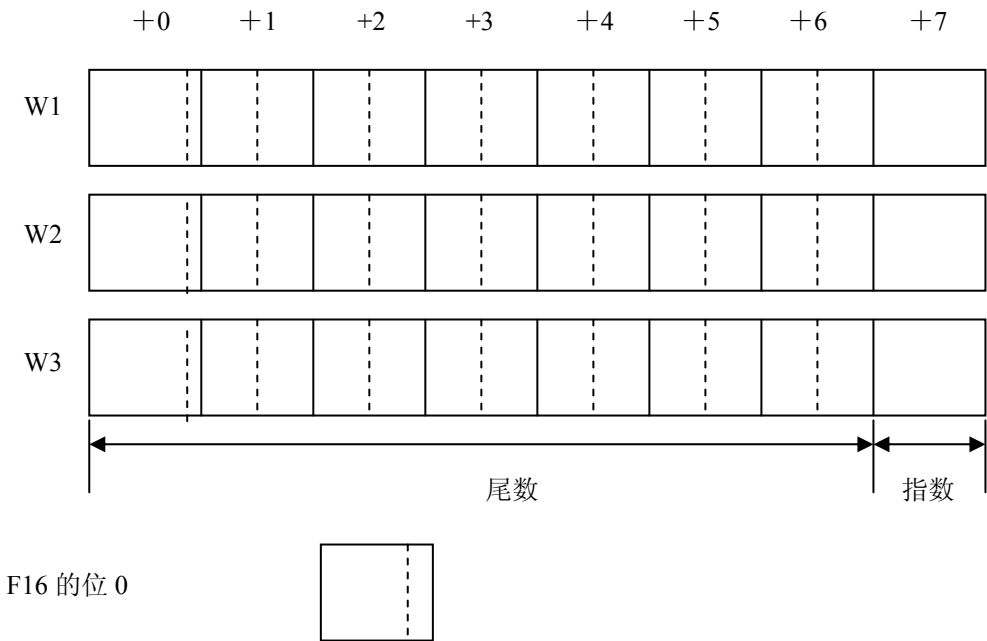
数据格式如下所示。此格式用 7 字节的尾数（1 字节符号位、6 字节数据）和 1 字节的指数表现。尾数的符号：0 为正、1 为负，尾数的数据为 12 位的有效数据（BCD）。指数只能取 00H~0CH 的数值，如果设定 0DH~FFH 的值，就不保证运算结果。



用此格式表现数的例子如下所示：

1234	0						1	2	3	4	00
1.234	0						1	2	3	4	03
-15000	1					1	5	0	0	0	00
-0.999999999999	1	9	9	9	9	9	9	9	9	9	0C

此格式被用于如下所示的 W1、W2、W3 的 3 个文件（零页 RAM 区）。



- W1: 保存被运算数和结果。
- W2: 保存运算数。地址为 $W2=W1+8$ 。
- W3: 在进行乘除法运算时，用作工作文件。地址为 $W3=W2+8$ 。
- F16 的位 0: 在发生错误时，为 1。

各运算的处理顺序如下所示。加減运算用 W2 的位 3 的值判断，请注意。

- 乘法运算
 - ①给 W1 设定被运算数。
 - ②给 W2 设定运算数。
 - ③JSR MULT

- 除法运算
 - ①②与乘法运算相同
 - ③JSR DIV

- 加法运算
 - ①②与乘法运算相同
 - ③JSR ADSB

- 减法运算
 - ①②与乘法运算相同
 - ③将 W2 的位 3 置 1。
 - ④JSR ADSB

在执行这些处理后，能用零页 RAM 地址 F16 的位 0 的数据检查运算结果是否有错。如果 (F16 的位 0) = 1，就为错误的运算结果。对于各运算的错误如下所示：

运算名	错误
加法运算	运算结果超过了 999999999999
减法运算	运算结果超过了 -999999999999
乘法运算	运算结果超过了 ± 999999999999
除法运算	• 被 0 除 • 运算结果超过了 ± 999999999999

(3) 程序清单

```
;*****
;      12 digits BCD number addition,subtraction
;      ,multiplication,division
;*****
;
;      RAM ASSIGN
;
;      0      1      2      3      4      5      6      7
; W1: SIGN MSD<-----LSD      INDX
; W2: SIGN MSD<-----LSD      INDX
; W3: SIGN MSD<-----LSD      INDX
;
;*****
;      W1 <- W1/W2
;*****
DIVI:
      JSR      BFITL          ;W3<-W1 copy,W1<-0
      SBC      8,X            ;D=0,T=1,W1 INDX<-W1 INDX-W2 INDX
      AND      #0FH          ;W1 INDX MSD clear
      SEB      0,F16
      BCC      DIV1          ;W1 INDX<W2 INDX?
      CLB      0,F16          ;No
DIV1:  CLT                    ;T<-0
      SED                    ;Decimal mode
      JSR      SUB0          ;W1<-W1-W2
      BCS      DIV2          ;Borrow arise?
      TST      W1            ;Yes,then test W1
      BEQ      DIV3
      DEC      W1
DIV2:  CLD                    ;Binary mode
      RRF      W3+6
      CLC
      LDA      W3+6
      ADC      #10H
      STA      W3+6          ;W3 LSD<-W3 LSD+1
      RRF      W3+6
      BCC      DIV1          ;W3 LSD>=15?
ERROR: SEB      0,F16          ;0 Division error
      CLD                    ;Decimal mode clear
      RTS
;
DIV3:  JSR      ADD0          ;W1<-W1+W2,add back
      LDA      W3+7
      CMP      #12
      BCS      DIV6
      INC      W3+7
;
;      W1 MSD<--W1 LSD<--W3 MSD<--W3 LSD
DIV4:  LDY      #4            ;4 bits left rotate
DIV5:  LDX      #6
      CLC
DIV51: ROL      W3,X
      DEX
      BNE      DIV51
      LDX      #6
      LDA      #7
```

```

DIV52:  ROL    W1,X
        DEX
        DEC    A
        BNE    DIV52
        DEY
        BNE    DIV5          ;Rotate end?
        INX
        SET     ;T flag set
        AND    #0FH
        BRA    DIV1
;
DIV6:   LDA    W3+1
        AND    #0F0H
        BNE    DIV10
        BBS    0,F16,DIV7
        LDA    W1+7
        CMP    #12
        BCS    DIV10
DIV7:   INC    W1+7
        BBC    4,W1+7,DIV41
        CLB    4,W1+7
        CLB    0,F16
DIV41:  BRA    DIV4
ERROR1: BRA    ERROR
;
DIV10:  BBS    0,F16,ERROR    ;Over flow error
        LDA    W1+7
        CMP    #13
        BCS    ERROR        ;Under flow error
DIV12:  LDY    #7            ;W1<-W3 copy
        LDX    #W1
        SET     ;T<-1
DIV15:  LDA    16,X
        INX
        DEY
        BNE    DIV15
;
        CLT          ;T<-0
        JSR    INDX    ;"0" condense
        CLD          ;Decimal mode clear
        CLC
        LDA    W1
        ADC    W2
        STA    W1        ;Get sign bit
        CLB    0,F16     ;Normal return
        RTS
;
;*****
;      W1 <- W1*W2
;*****
;
MULT:   JSR    BFITL      ;W3 <- W1 copy,W1 clear
        CLC          ;C flag clear
        ADC    8,X       ;W1_INDX <- W1_IND+X+2_IND

```

```

        CLB      0,F16
        BBC      4,W1+7,MULT2      ;W1_INDX <- W_INDX+W2_INDX
        SEB      0,F16            ;Yes
        CLB      4,W1+7

MULT2:
        CLT
                                ;T flag clear
MULT7:
        LDA      W3+6
        AND      #$0F
        BNE      MULT3
        INC      W3+7              ;W1_MSD->W1_LSD->W3_MSD->W3_LSD

MULT9:
        LDY      #4
MULT6:
        LDX      #0
        LDA      #7
        CLC

MULT4:
        ROR      W1,X
        INX
        DEC      A
        BNE      MULT4
        LDX      #1
        LDA      #6

MULT5:
        ROR      W3,X
        INX
        DEC      A
        BNE      MULT5
        DEY
        BNE      MULT6            ;4 bits right rotate end ?
        LDA      W3+7
        CMP      #12
        BCC      MULT7
        BBS      0,F16,MULT8
        LDX      #W1+1            ;W1 "0" test

MULT81:
        LDA      0,X
        BNE      MULT8
        INX
        CPX      #W1+7
        BNE      MULT81
        BRA      DIV12            ;Get result (to division routine)

MULT8:
        LDA      W1+7
        BNE      MULT10
        BBC      0,F16,ERROR1      ;Overflow error
        CLB      0,F16

MULT10:
        DEC      W1+7
        LDA      W1+7
        AND      #$0F
        STA      W1+7
        BRA      MULT9

MULT3:
        DEC      W3+6
        SED

```

```

        JSR    ADD0
        BCC    MULT31
        INC    W1
MULT31:
        BRA    MULT2
;*****
;        W1 <- W1-W2
;*****
;
SUB0:
        SEC
        LDA    #6
        LDX    #W1+6
        SET
                                ;T flag set
SUB01:
        SBC    8,X
        DEX
        DEC    A
        BNE    SUB01
        CLT
                                ;T flag clear
        RTS
;*****
;        W1 <- W1+W2
;*****
;
ADD0:
        CLC
        LDA    #6
        LDX    #W1+6
        SET
                                ;T flag set
ADD01:
        ADC    8,X
        DEX
        DEC    A
        BNE    ADD01
        CLT
                                ;T flag clear
        RTS
;*****
;        Following "0" condense,then modify INDX and data
;*****
;
INDX:
        TST    W1+7                                ;Test W1 INDX
        BEQ    INDX1
        LDA    W1+6
        AND    #$0F
        BNE    INDX1                                ;Valid data remain ?
        LDX    #W1                                ;No
        JSR    SFR4                                ;Condense to LSB direction
        DEC    W1+7                                ;Modify INDX
        BRA    INDX                                ;again
INDX1:
        RTS
;*****
;        W1 -> W3 ,and 0 -> W1
;*****
;

```

```

BFITL:
    CLT
    LDX    #W1
BFITL1:
    LDA    0,X
    STA    16,X
    LDA    #0
    STA    0,X
    INX
    CPX    #W1+7
    BNE    BFITL1
    LDM    #0,W3+7           ;W3_INDX initial
    CLD                     ;Decimal mode clear
    SET                     ;T flag set
    RTS
;*****
;      WN 4 bits left shift
;*****
;
SFL4:
    LDA    #4
SFL41:
    ASL    6,X
    ROL    5,X
    ROL    4,X
    ROL    3,X
    ROL    2,X
    ROL    1,X
    DEC    A
    BNE    SFL41
    RTS
;*****
;      WN 4 bits right shift
;*****
;
SFR4:
    LDA    #4
SFR41:
    LSR    1,X
    ROR    2,X
    ROR    3,X
    ROR    4,X
    ROR    5,X
    ROR    6,X
    DEC    A
    BNE    SFR41
    RTS
;*****
;      Adjust INDX to W1 to W2
;*****
;
ADIX:
    LDA    W1+7
    CMP    W2+7
    BEQ    ADIX1
    BCC    ADIX23
ADIX22:

```

```

        LDX    #W2
ADIX2:
        LDA    1,X
        AND    #$F0
        BEQ    ADIX21
        CPX    #W1
        BEQ    ADIX51
        LDX    #W1
ADIX5:
        JSR    SFR4
        DEC    7,X
        BRA    ADIX
ADIX23:
        LDX    #W1
        BRA    ADIX2
ADIX51:
        LDX    #W2
        BRA    ADIX5
ADIX21:
        JSR    SFL4
        INC    7,X
        BRA    ADIX
ADIX1:
        RTS
;*****
;      Execute addition,subtraction
;*****
;
ADSB:
        SED                                ;Decimal mode set
        CLT                                ;T flag clear
        JSR    ADIX                        ;Adjust INDX
ADSB1:
        BBS    3,W2,ADSB6
        BBS    0,W1,ADSB7
        BBS    0,W2,ADSB10
ADSB9:
        JSR    ADD0                        ;W2 <- W1+W2
        BCC    ADD2                        ;C arise ?
        SEB    0,F16                       ;Yes
        JSR    SFR4                        ;1 digit right shift
        DEC    W1+7                        ;INDX -1
        BMI    ADD3                        ;FF ?
        CLC                                ;No
        LDA    W1+1
        ADC    #$10
        STA    W1+1
ADD2:
        CLB    0,F16
        JSR    INDX
ADD3:
        CLD                                ;Decimal mode clear
        RTS
ADSB7:
        BBS    0,W2,ADSB9
ADSB10:
        JSR    SUB0                        ;W1 <- W1-W2

```

```

        BCS     SUB2
        INX
SUB3:
        LDA     #$99
        SEC
        SBC     0,X
        STA     0,X           ;Get 99-X
        INX
        CPX     #W1+7
        BNE     SUB3
        DEX
        SET
        LDY     #6           ;C=1
SUB4:
        ADC     #0
        DEX
        DEY
        BNE     SUB4
        CLT           ;Get 100's complement
        COM     W1         ;Get sign bit in bit0
SUB2:
        CLB     0,F16
        JSR     INDX
        CLD           ;Decimal mode clear
        RTS
ADSB6:
        BBS     0,W1,ADSB11
        BBS     0,W2,ADSB9
        BRA     ADSB10
ADSB11:
        BBS     0,W2,ADSB10
        BRA     ADSB9

```

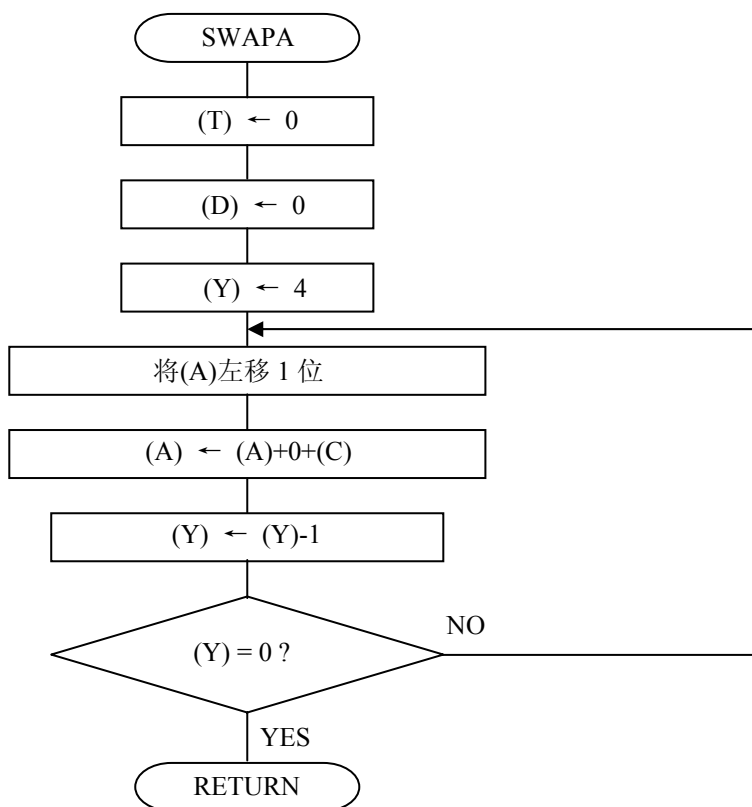
5. 替代指令

5.1 交换累加器

(1) 概要

将累加器的高 4 位和低 4 位进行替换。

(2) 流程图



(3) 程序清单

```

;*****
;      Swap A register
;*****
;
SWAPA:
    CLT                ;T flag clear
    CLD                ;Decimal mode clear
    LDY    #4
SWAPA1:
    ASL    A
    ADC    #0
    DEY
    BNE    SWAPA1
    RTS

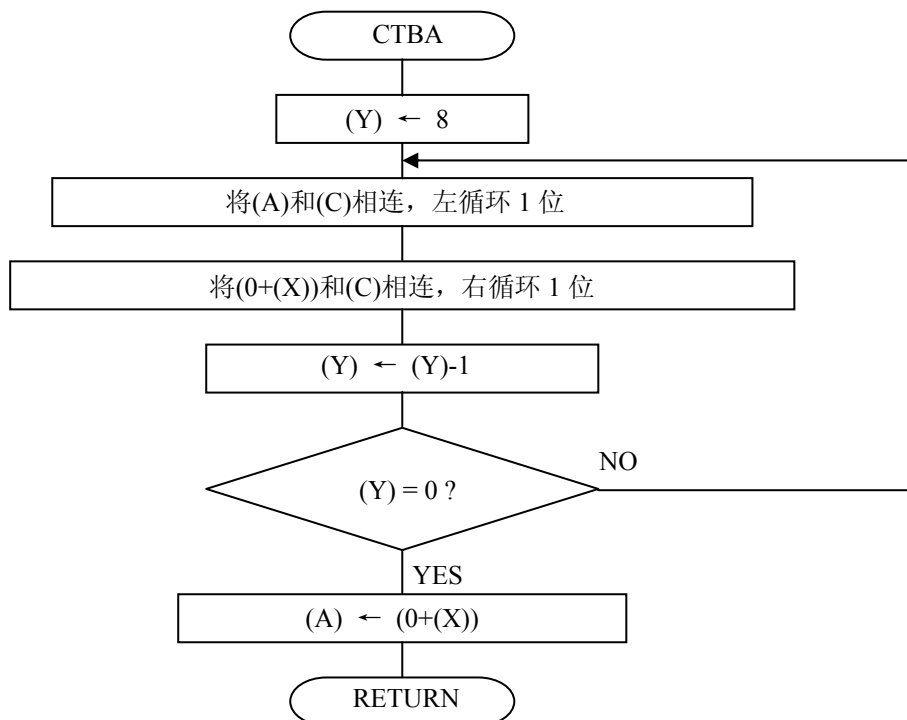
```

5.2 颠倒累加器的位顺序

(1) 概要

将累加器的位数据颠倒排列。使用变址寄存器 X 指定的 1 字节工作存储器。

(2) 流程图



(3) 程序清单

```

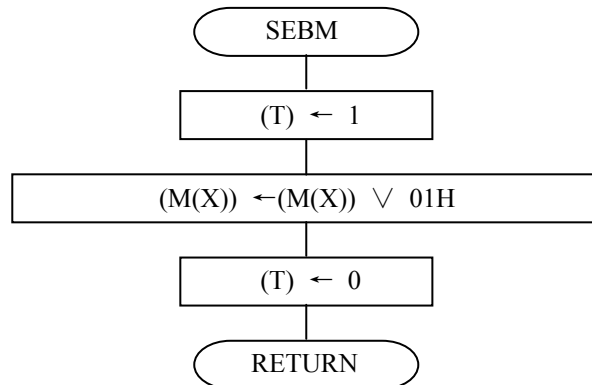
;*****
;   Counter bit A register
;*****
;
CTBA:
    LDY    #8
CTBA1:
    ROL    A
    ROR    0,X
    DEY
    BNE    CTBA1
    LDA    0,X
    RTS
  
```

5.3 存储器的置位

(1) 概要

将变址寄存器 X 指定的存储器的位 0 置“1”。

(2) 流程图



注：M (X) 为变址寄存器 X 指向地址的存储器。

(3) 程序清单

```

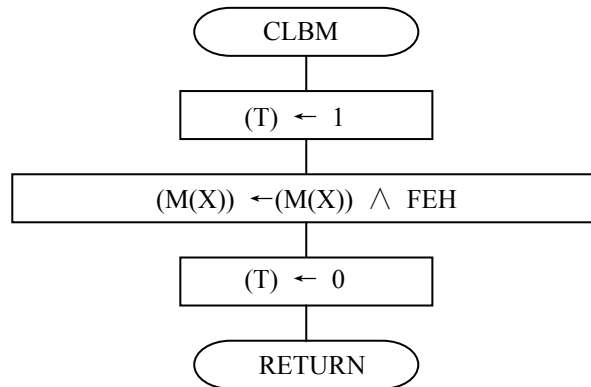
;*****
;      Set bit 0 of M(X)
;*****
;
SEBM:
    SET     #1          ;T flag set
    ORA     #1          ;Bit 0 set
    CLT     #1          ;T flag clear
    RTS
  
```

5.4 存储器的位清除

(1) 概要

将变址寄存器 X 指定的存储器的位 0 清 “0”。

(2) 流程图



注：M (X) 为变址寄存器 X 指向地址的存储器。

(3) 程序清单

```

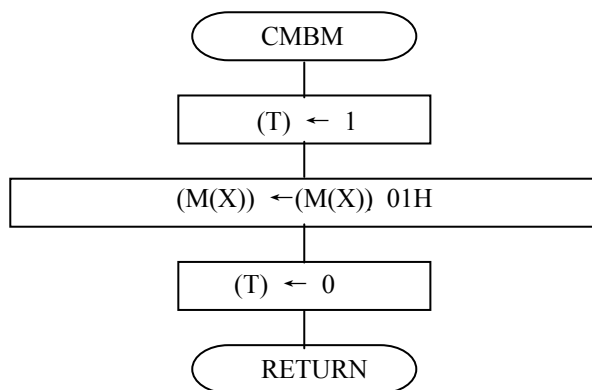
;*****
;      Clear bit 0 of M(X)
;*****
;
CLBM:
    SET             ;T flag set
    AND    #$FE     ;Bit 0 clear
    CLT             ;T flag clear
    RTS
  
```

5.5 存储器的位反转

(1) 概要

将变址寄存器 X 指定的存储器的位 0 进行反转。

(2) 流程图



注：M (X) 为变址寄存器 X 指向地址的存储器。

(3) 程序清单

```

;*****
;      Complement bit 0 of M(X)
;*****
;
CMBM:
    SET     #1      ;T flag set
    EOR     #1      ;Bit 0 complement
    CLT     ;T flag clear
    RTS
  
```

6. 程序利用的注意事项

程序利用时，必须注意以下几点。同时请参考 740 族软件手册。

- (1) 必须对客户的系统进行充分的评价。
- (2) 本程序通常将 X 变址模式标志 T 和 10 进制模式标志 D 处理为“0”。请不要在将 X 变址模式标志 T 或 10 进制模式标志 D 置“1”后调用本程序。
- (3) 如果在 10 进制模式时 (D=1) 执行 ADC 和 SBC 指令，处理器状态寄存器的负标志 N、溢出标志 V 和零标志 Z 就无效。另外，如果运算结果发生进位，进位标志 C 就被置“1”；如果发生借位，进位标志 C 就被清“0”。
- (4) 在 10 进制模式时 (D=1)，SEC 指令、CLC 指令或 CLD 指令不能紧接在 ADC 指令或 SBC 指令后使用。

7. 参考文献

软件手册：740 族软件手册

740 Family Software Manual

最新版本请从瑞萨科技网页取得

<http://www.cn.renesas.com>

修订记录

Rev.	发行日	修订内容	
		页	修订要点
1.00	2005.09.30	—	初版发行

Cautions

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.

注意

本文只是参考译文，前页所载英文版“Cautions”具有正式效力。

请遵循安全第一进行电路设计

1. 虽然瑞萨科技尽力提高半导体产品的质量和可靠性，但是半导体产品也可能发生故障。半导体的故障可能导致人身伤害、火灾事故以及财产损害。在电路设计时，请充分考虑安全性，采用合适的如冗余设计、利用非易燃材料以及故障或者事故防止等的安全设计方法。

关于利用本资料时的注意事项

1. 本资料是为了让用户根据用途选择合适的瑞萨科技产品的参考资料，不转让属于瑞萨科技或者第三者所有的知识产权和其它权利的许可。
2. 对于因使用本资料所记载的产品数据、图、表、程序、算法以及其它应用电路的例子而引起的损害或者对第三者的权力的侵犯，瑞萨科技不承担责任。
3. 本资料所记载的产品数据、图、表、程序、算法以及其它所有信息均为本资料发行时的信息，由于改进产品或者其它原因，本资料记载的信息可能变动，恕不另行通知。在购买本资料所记载的产品时，请预先向瑞萨科技或者经授权的瑞萨科技产品经销商确认最新信息。
本资料所记载的信息可能存在技术不准确或者印刷错误。因这些错误而引起的损害、责任问题或者其它损失，瑞萨科技不承担责任。
同时也请通过各种方式注意瑞萨科技公布的信息，包括瑞萨科技半导体网站。
(<http://www.renesas.com>)
4. 在使用本资料所记载部分或者全部数据、图、表、程序以及算法等信息时，在最终做出有关信息和产品是否适用的判断前，务必对作为整个系统的所有信息进行评价。由于本资料所记载的信息而引起的损害、责任问题或者其它损失，瑞萨科技不承担责任。
5. 瑞萨科技的半导体产品不是为在可能和人命相关的环境下使用的设备或者系统而设计和制造的产品。在研讨将本资料所记载的产品用于运输、交通车辆、医疗、航空宇宙用、原子能控制、海底中继器的设备或者系统等特殊用途时，请与瑞萨科技或者经授权的瑞萨产品经销商联系。
6. 未经瑞萨科技的书面许可，不得翻印或者复制全部或者部分资料的内容。
7. 如果本资料所记载的某产品或者技术内容受日本出口管理限制，必须在得到日本政府的有关部门许可后才能出口，并且不准进口到批准目的地国家以外的国家。
禁止违反日本和（或者）目的地国家的出口管理法和法规的任何转卖、挪用或者再出口。
8. 如果需要了解本资料所记载的信息或者产品的详细，请与瑞萨科技联系。