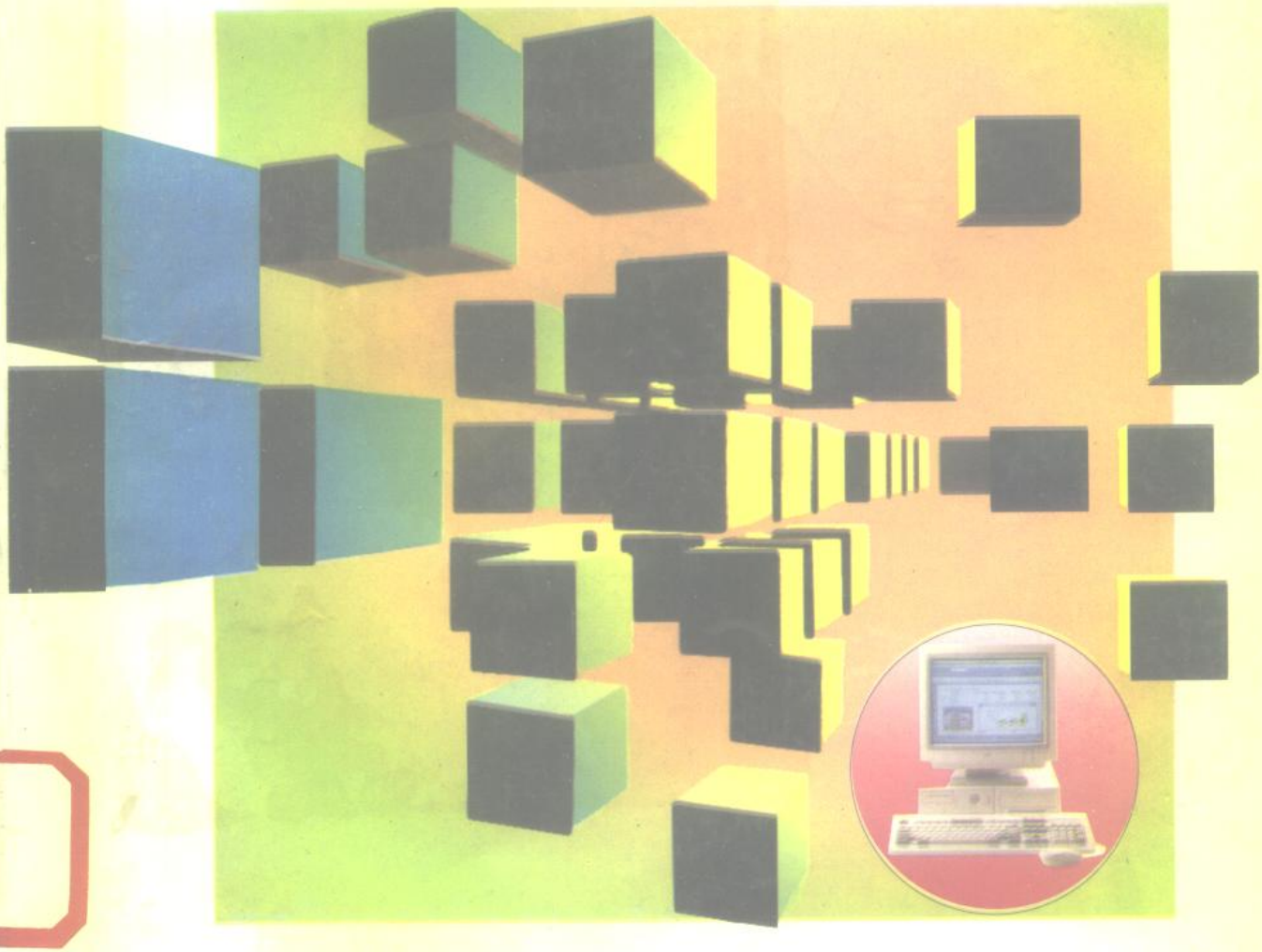
齐秋群　刚寒冰 等

Motorola

单片机实用技巧集萃

（一）



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

Motorola 单片机实用技巧集萃

(一)

齐秋群　刚寒冰等

电子工业出版社

内 容 提 要

本书详细地介绍了几十个实用性很强的 Motorola M68HC05/M6805/M68HC11 单片机应用实例，内容涉及通讯、家用电器、测量/控制/显示系统、仪表、单片机系统设计与系统扩展技术、单片机开发与编程等。每个例子都给出详细的硬件电路和具体软件程序，读者可以直接采用或稍加改动应用到相近的场合或改用其他型号的单片机。

本书实用性强，内容翔实，取材新颖，对开发、应用单片机具有很高的参考价值。本书适用于从事单片机开发与应用的工程技术人员、科研人员和高校师生阅读。

Motorola 单片机实用技巧集萃

(一)

齐秋群　刚寒冰等

责任编辑　郭　晓

*

电子工业出版社出版

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行　各地新华书店经销

电子工业部情报所印刷

*

开本：787×1092 毫米 1/16 印张：22.25　字数：570 千字

1996 年 4 月第一版　　1996 年 4 月北京第一次印刷

印数：3500 册　　定价：38.00 元

ISBN7－5053－3334－8/TP・1270

前　言

Motorola M68HC05、M68HC11、M6805 系列是国际上应用最广泛的单片机，具有功能丰富、性能价格比高、功耗低、系统设计简单、使用方便、速度高等特点，在家电、通讯、测控系统、仪器、汽车领域获得了广泛应用。

目前国内 Motorola 单片机用户迅速增加，但目前市场上大都是主要介绍 Motorola 单片机原理或技术手册的图书，工程技术人员在开发、设计、应用单片机时可参考的有关实际应用例子的资料匮乏。我们编写这本书也正是为了弥补这一缺陷。本书详细介绍几十个成功地应用单片机的例子，每个实例都有具体的硬件电路和详细的软件程序，读者可直接采用或进行移植，对广大开发使用单片机的专家与同行无疑具有很高的参考价值。

本书的例子都是从许多应用文章精选出来的，实用价值高，内容翔实、完整，取材新颖，内容涉及家电、通讯、控制与测量系统、仪器仪表单片机系统设计与系统扩展、单片机开发技术等各个领域。

本书主要由齐秋群、刚寒冰编写，参加编写者还有刚砺韬、姜洪福、常安、文州峰、高京斋、姜朋、刘化亮、刘联、郭小吉、李宇仁。

本书不足和错误之处，请读者批评指正（北京理工大学电子工程系，100081）。

作　者

1995.9.

目　录

1. 具有 128K 字节寻址能力的 M68HC11 系统 …… 1
2. MC68HC11K4 的存储器映象扩展 …… 17
3. M68HC11 浮点运算软件包 …… 25
4. 用 PC 机对 M68HC11 内部或外部 EEPROM 进行编程 …… 57
5. 减小 MCU 中 A/D 转换器的误差 …… 69
6. M68HC11 的自引导模式 …… 77
7. M68HC11 堆栈的使用 …… 109
8. MC68HC68T1 实时时钟的应用 …… 127
9. 用单片机产生脉冲的方法 …… 134
10. 利用单片机的输入捕捉功能检测脉冲宽度 …… 140
11. MC68HC05 SR3 与 MC6805R3 性能比较 …… 146
12. 用 MC68HC705C8A 代替 MC68HC705C8 …… 150
13. MC68HC805B6 EEPROM 单片机编程模块 …… 156
14. MC68HC05F2 DTMF 输出的廉价低压有源滤波器 …… 158
15. 正确选用单片机 …… 160
16. HCMOS 单片机的电磁兼容(EMC)设计 …… 163
17. 用 MC68HC05B4 和 MC14489 测量和显示温度 …… 170
18. MC68HC05T1 的屏幕显示(OSD)在 TV 中的应用 …… 183
19. MC68HC05B6 的 RAM 和 EEPROM1 的串行自引导方式 …… 205
20. MC68HC8085B6 和 MC68HC705B5 串行/并行编程电路 …… 215
21. MC68HC05 E0 EPROM 仿真器 …… 219
22. 用 M6805/M68HC05 驱动 LCD …… 248
23. 采用 MC68HC805B6 的 MCM2814 编程器 …… 258
24. 由 M68HC11 单片机控制步进电机 …… 269
25. M68HC11 全应答方式并行 I/O 通讯 …… 275
26. 由单片机实现的免校准压力传感器系统 …… 277
27. 用单片机提高压力传感器 A/D 转换分辨率的方法 …… 282
28. 补偿型压力传感器与单片机/数字电路的接口电路 …… 285
29. 能与 MCU 接口的 MPX2000 系列压力传感器频率输出系统 …… 290
30. 半导体压力传感器与单片机的接口方法 …… 294
31. MC68HC05C5 SIOP 与 I^2C 外围器件的接口方法 …… 303
32. 采用 MC6805 单片机的电话拨号技术 …… 316
33. M68HC11 单片机与 M6805 单片机利用 SPI 进行同步串行通讯 …… 329
34. 利用 MC6805R3 MCU 构成的温度控制器 …… 339
35. MC68HC05T3 自引导程序/自引导编程电路 …… 346

1. 具有 128K 字节寻址能力的 M68HC11 系统

M68HC11 系列单片机的最大寻址范围为 64K 字节，这在有些应用场合是不够用的。这里介绍两种存储器分页方法，使 M68HC11 的寻址范围扩大到 128K 字节。这种分页技术可以扩展到对多个 EPROM、RAM 或 EEPROM 存储器进行寻址的场合。

1.1 分页原理

M68HC11 MCU 可寻址 64K 字节连续的地址空间。当需要的地址空间大于 64K 字节时，要用另一个具有相同地址空间存储器块代替这一存储器块，这一技术称为分页(Paging)，是扩展地址空间的简单而有效的方法。在这种方法中，有多个数据块(页)重叠，但在某个时刻 CPU 只能访问其中一页。

有两种基本方法：方法 A 只采用软件和一个 I/O 口引脚控制最高位地址 A16；方法 B 采用少量硬件和软件控制最高 3 位地址 A14、A15 和 A16。

在下面的例子中，采用具有非公用地址/数据总线的 MC68HC11G5 说明分页技术。任何其他的 M68HC11 器件均可采用类似的方法。

方法 A 的特点是不需要额外的硬件，对软件的限制少，用户码可达 64K 字节并保持在同一页，但中断等待时间长。方法 B 的优点是不影响中断等待时间，有一份向量表。该例中用户码主程序最大长度为 48K 字节，另有 5 页分别为 16K 字节，用于子程序和表格。

1.2 方法 A——软件方法

EPROM 的地址 A16 直接由 M68HC11 的 PD5 控制，如图 1-1 所示，复位后 D 口设置为输入状态。复位后，控制地址 A16 的引脚状态是很重要的，因此，PD5 接一个 10kΩ 的上拉电阻，以强迫复位后 A16 为逻辑高电平状态。应该注意，在数据方向寄存器写入 1 设置为输出之前，必须先向数据寄存器写入 1。

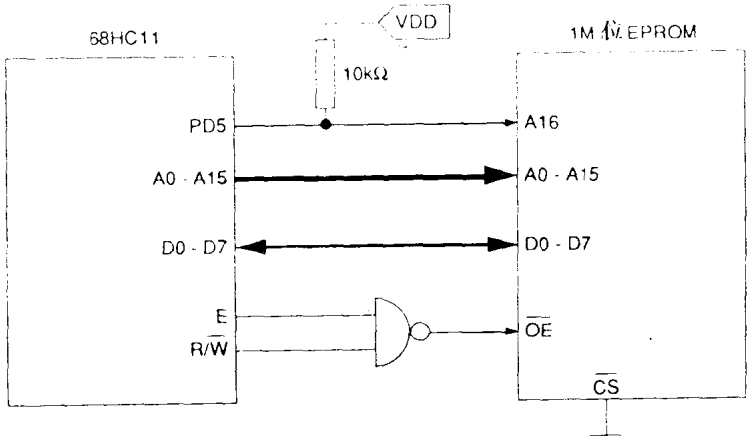


图 1-1 软件分页结构图

这种方法可允许 M68HC11 访问 128K 字节的 EPROM，该 EPROM 就如同两个均为 64K 字节的存储器，通过改变 EPROM 的地址线 A16 的状态分页。要确保口时序在地址选通（MC68HC11G5 的 E 时钟的上升沿）之前改变引脚状态，并且满足建立和保持时间。

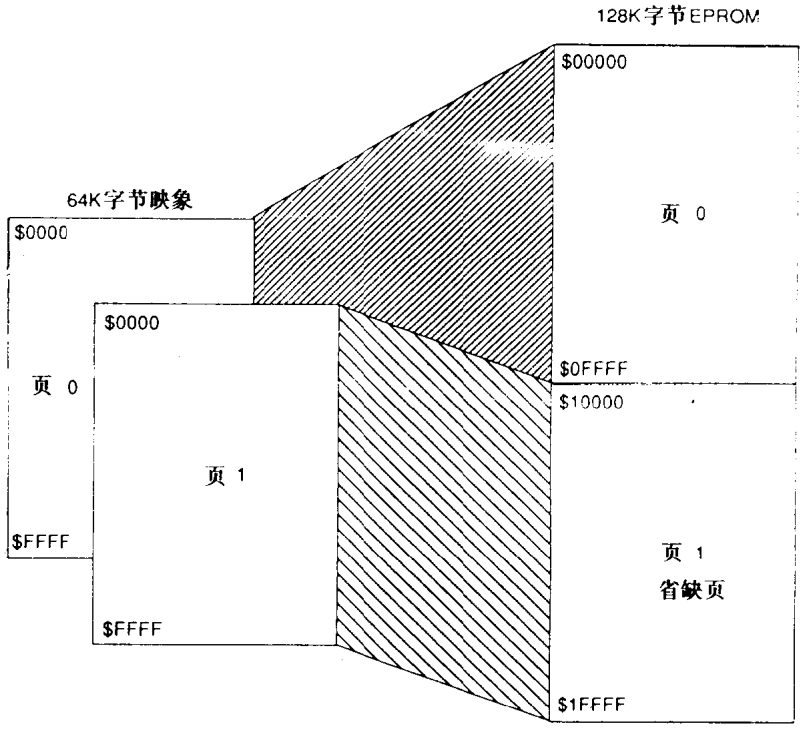


图 1-2 软件分页说明

图 1-2 是这种分页技术的原理图。它不能从一个 64K 页直接跳转到另一个 64K 页。解决该问题有两个方法：用户码在 RAM 中建立一个程序（它由两页公用，因为它位于内部，不受 PD5 的影响）；或在两页的相同地址具有页转换程序。后面的程序采用后一种方法。

1.2.1 中断程序

在 PD5 清零或置位之前，页转换程序存储当前页，而且页转换程序的跳转命令在两个存储器中必须具有相同的地址。因为 PD5 清零或置位将立即改变存储器页，但程序计数器正常加 1。因此，从页 0 变为页 1 时，为保证正确跳转，变址寄存器 X 必须装入在新的一页要执行的程序的地址，图 1-3 是改变页时从页 1 变为页 0 时的执行情况。

从中断返回的程序需要在执行中断程序之前先检查含有存储器页的 RAM 地址。然后，或者立即执行 RTI 命令（若处于同一页）；或者改变 PD5 的状态，再在正确的页执行 RTI 命令。注意，对于 JMP 0,X 指令，RTI 必须在两者的相同地址。CCR 的 I 位（中断禁止位）在该期间置位，以便正确运行，否则返回页可能变化。在当前中断之前用堆栈保持页号可克服这一问题。

1.2.2 其他程序

利用相同的转换程序可在任何时候从一页转换到另一页，无需在 RAM 中存储目前页，因此，变页程序可在 BCLR 或 BSET 开始，节省 4 个周期，可使换页延迟减少到 17 个周期。注意，不能用该例方法执行 JSR 命令进入其他页，因为 RTS 不会返回到原来的一页，但可进行适当

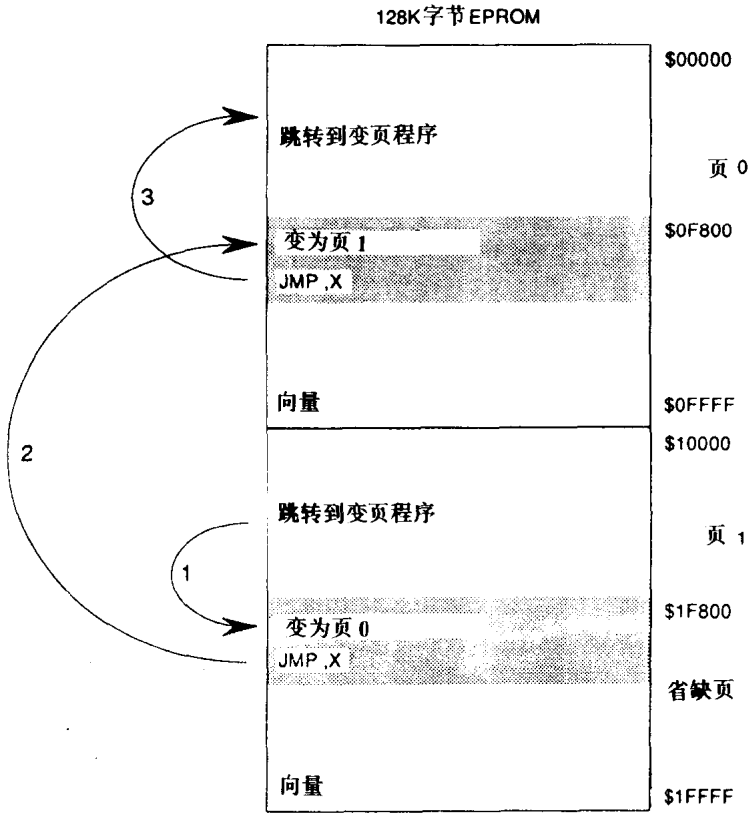


图 1-3　从页 1 变为页 0 的程序流程图

注：1. 跳转到变页程序　2. 变为页 0　3. 跳转到 X 寄存器的地址(页 0)

修改。在这种情况下，应该用堆栈保持正确的返回页，或将 CCR 的 I 位置位防止中断。

1.2.3　重要条件

控制复位后 A16 的状态是很重要的。该例中 PD5 复位后为输入，并用上拉电阻强迫 A16 为高。如果使用固定输出脚，则为逻辑 0(不需上拉电阻)。设置 I/O 口的初始化程序必须在省缺页，否则不能正确地控制 A16。类似地，若用双向 I/O 线，要有下拉电阻来确定复位后的地址线 A16。

RAM、寄存器和内部 EEPROM 若允许时，将总是出现在存储器映象中，因此应避免与这些地址相冲突。也可由改变 INIT 寄存器来改变 RAM 和寄存器的地址。

1.2.4　软件分页方法程序

* 1M 位(128 字节)EPROM 分为 2×64K 字节页
* A16 连至 PD5，其状态决定访问哪一页，复位后 PD5＝1
* 该程序以 68HC11G5 编写，可容易地修改到其他 68HC11 器件

```
PORTA       EQU     $00
DDRA        EQU     $01
PORTB       EQU     $04
```

```
PORTC       EQU     $06
DDRC        EQU     $07
PORTD       EQU     $08
DDRD        EQU     $09
TMSK2       EQU     $24
TFLG2       EQU     $25
RTII        EQU     $40
RTIF        EQU     $40
PACTL       EQU     $26
DDRA7       EQU     $80
REGS        EQU     $1000
 * RAM 定义(从$0000～$01FF)
            ORG     $0000
 PAGE       RMB     1
 TIME       RMB     2
 *
 NPAGE      EQU     $20          PD5 页控制线
 ROMBASE    EQU     $0200
 CHANGE     EQU     $F800
 VECTORS    EQU     $FFCC

 * 主程序开始
 * 页 0(EPROM 的第一部分)
      ORG           ROMBASE
 RESET0 LDX         #RESET
 JMP                CHGPAGE0
 *
LOOPP0  BSET    PORTA,Y,#$10        位 4 翻转
        BCLR    PORTA,Y,#$10
        LDX     #LOOPP1             获得页 1 的返回地址
        JMP     CHGPAGE0            转变页程序
*  实时中断服务程序
RTISRV  BRSET   TFLG2,Y,#RTIF,RTISERV
        RTI                         若不是正确的中断源则返回，
 *                                  在页 1 时,因为中断向量只指向
                                    该处,故为 RTI
RTISERV
        LDAA    #%01000000          零页中断从此开始
        STAA    TFLG2,Y             清 RTI 标志
        LDAA    TIME+1              取时间计数器
        INCA
        STAA    PORTB+REGS          将时间存贮在 B 口
        LDX     TIME
        INX
```

```
        STX     TIME                    并存入 RAM 中
        JMP     RETRTI0                 转 RTI 程序
*  改变页程序
*  执行该程序时必须使 I 位置位,以防止中断。否则可能造成错误
*  该程序在两页的相同地址都出现
*  转移程序
        ORG     CHANGE                  该程序的地址固定
*                                       (周期)
CHGPAGE0
        LDAA    #0              (2)     目前页号置为 0
        STAA    PAGE            (2)     存入 PAGE
        BSET    PORTD,Y,#NPAGE  (8)     由设置 PD5 改变页
        JMP     0,X             (3)     该码在两页中均相同
*  从中断程序返回
RETRTI0
        LDAA    PAGE            (2)     获得发生中断的页
        CMPA    #1              (2)     是否为页 1?
        BEQ     RTIPAGE0        (3)     是,则变页
        RTI                     (12)    否则,从中断返回
RTIPAGE0
        BSET PORTD,Y,#NPAGE     (8)     变页并从中断返回

        RTI                     (12)    该码在两页中相同
*  向量
        ORG     VECTORS
        FDB     RESET0          EVENT 2
        FDB     RESET0          EVENT 1
        FDB     RESET0          TIMER OVERFLOW 2
        FDB     RESET0          INPUT CAPTURE 6 / OUTPUT COMPARE 7
        FDB     RESET0          INPUT CAPTURE 5 / OUTPUT COMPARE 6
        FDB     RESET0          SCI
        FDB     RESET0          SPI
        FDB     RESET0          PULSE ACC INPUT
        FDB     RESET0          PULSE ACC OVERFLOW
        FDB     RESET0          TIMER OVERFLOW 1
        FDB     RESET0          INPUT CAPTURE 4 / OUTPUT COMPARE 5
        FDB     RESET0          OUTPUT COMPARE 4
        FDB     RESET0          OUTPUT COMPARE 3
        FDB     RESET0          OUTPUT COMPARE 2
        FDB     RESET0          OUTPUT COMPARE 1
        FDB     RESET0          INPUT CAPTURE 3
        FDB     RESET0          INPUT CAPTURE 2
        FDB     RESET0          INPUT CAPTURE 1
        FDB     RTISRV          REAL TIME INTRRUPT
        FDB     RESET0          IRQ
```

```
        FDB     RESET0          XIRQ
        FDB     RESET0          SWI
        FDB     RESET0          ILLEGAL OPCODE
        FDB     RESET0          COP
        FDB     RESET0          CLOCK MONITOR
        FDB     RESET0          RESET
* 页1(EPROM的第二部分)
    * 主程序,不在中断控制之下
            ORG   ROMBASE
RESET       LDS   #$01FF
            JSR   SETUP                     初始化RTI中断和DDR
LOOP1       LDAA  #$FF
LOOP        BSET  PORTA,Y,#$08              位3翻转
            BCLK  PORTA,Y,#$08
            LDX   #LOOPP0                   转向另一页
            JMP   CHGPAGE1
LOOPP1
            DECA                            从另一页的返回点
            BNE   LOOP
            BRA   LOOP1                     再次开始循环
            *初始化程序
SETUP       SEI
            LDY                             #$1000  寄存器地址偏移量
            LDAA                            #$FF
            STAA                            DDRA+REGS  使A口均为输出
            STAA                            PORTD+REGS  确保PD5写入1
            STAA                            DDRD+REGS
LDAA        #%01000000
            STAA                            TFLG2+REGS  清RTI标志
            STAA                            TMSK2+REGS  允许RTI中断
            CLI
            RTS
*  Redirect to the Real time interrupt service routine
*  Page 1 routine for service routine located in page 0
INTRTI      BRSET TFLG2,Y,#RTIF,GOODINT
            RTI                             若不是正确的中断源则返回
GOODINT
            LDX #RTISERV                    (3周期)获得页0中断进入点
            JMP CHGPAGE1                    (3周期)转换页程序
* 换页程序
* 跳转程序
            ORG CHANGE
*                                           (周期数)
CHGPAGE1
            LDAA  #$1                       (2)使当前页号置1
```

```
                STAA PAGE                       (2)存储页号
                BCLR PORTD,Y,#NPAGE             (8)清 PD5,变页
                JMP 0,X                         (3)
*   从中断程序返回
RETRTI1
                LDAA PAGE               (2)
                CMPA #0                 (2)
                EBQ   RTIPAGE 1   -     (3)
                RTI                     (12)
RTIPAGE1
                BCLR PORTD,Y,#NPAGE     (8)
                RTI                     (12)
* 向量
        ORG     VECTORS
        FDB     RESET           EVENT 2
        FDB     RESET           EVENT 1
        FDB     RESET           TIMER OVERFLOW 2
        FDB     RESET           INPUT CAPTURE 6 / OUTPUT COMPARE 7
        FDB     RESET           INPUT CAPTURE 5 / OUTPUT COMPARE 6
        FDB     RESET           SCI
        FDB     RESET           SPI
        FDB     RESET           PULSE ACC INPUT
        FDB     RESET           PULSE ACC OVERFLOW
        FDB     RESET           TIMER OVERFLOW 1
        FDB     RESET           INPUT CAPTURE 4 / OUTPUT COMPARE 5
        FDB     RESET           OUTPUT COMPARE 4
        FDB     RESET           OUTPUT COMPARE 3
        FDB     RESET           OUTPUT COMPARE 2
        FDB     RESET           OUTPUT COMPARE 1
        FDB     RESET           INPUT CAPTURE 3
        FDB     RESET           INPUT CAPTURE 2
        FDB     RESET           INPUT CAPTURE 1
        FDB     INTRTI          REAL TIME INTRRUPT
        FDB     RESET           IRQ
        FDB     RESET           XIRQ
        FDB     RESET           SWI
        FDB     RESET           ILLEGAL OPCODE
        FDB     RESET           COP
        FDB     RESET           CLOCK MONITOR
        FDB     RESET           RESET
        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
        END
```

1.3 方法 B——软硬件结合的方法

这种方法与前一种方法类似,但用一些硬件取代部分软件。M68HC11 的 A14 和 A15 经

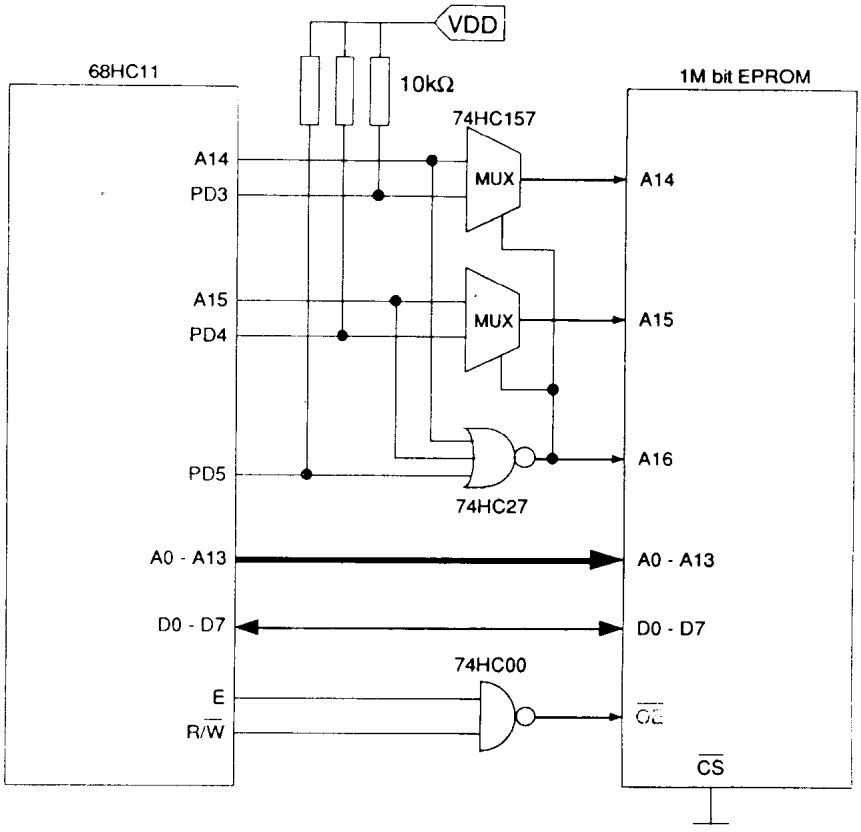


图 1-4　软硬件分页方法

或非后控制 EPROM 的 A16，如图 1-4 所示。它分为一个 48K 字节公共页和 5 个 16K 字节页。

例如，PD3 和 A14 地址线连到二选一多路器，或非门的输出决定这两个信号中哪个信号连到 EPROM 的 A14。类似地，或非门的输出还决定 PD4 和 A15 地址线这两个信号之中哪一个连到 EPROM 的 A15。若 A14 或/和 A15 为 1，则或非门输出(74HC27)为逻辑 0，即 EPROM 的 A16 为 0。这种情况下，HC11 的 A14 和 A15 分别控制 EPROM 的 A14 和 A15。这样，48K 字节主程序部分(EPROM)在任何时间都可在 $4000～$FFFF 访问。当 PD5、A14 和 A15 均为 0 时，(地址范围 $0000～$3FFF)，PD3 和 PD4 决定 EPROM 的 A14 和 A15(A16=1)。PD3、PD4 共有 4 种组合，即有 4 个 16K 字节页。这种结构共有 5 个 16K 字节页和一个 48K 字节页，该 48K 字节页总是处于存储器映象中，如表 1-1 所示。

因 PD3～PD5 均有上拉电阻，复位后为逻辑高状态(PD 口为输入)故复位后为主程序页和页 0。

1.3.1　C 语言的实现

程序由汇编语言写成，也可由 C 语言编写。后面给出了 C 语言程序和相应的汇编程序。

1.3.2　中断

当中断程序位于主 48K 页时，具有正常的等待时间(latency)，因为 CPU 总是可以访问这

48K 字节页，翻页中断程序可造成 25 个周期的延迟。

1.3.3 重要条件

这种方法要求向量必须指向换页程序位于的存储器的 48K 主页。处于其他页的程序只能通过 48K 页才能转移到另一页。

表 1-1 软硬件结合方法的页选择

PD5	A14	A15	A16	存储器页
0	0	0	1	由 PD3 和 PD4 决定页 1～页 4(每页 16K)
	0	1	0	48K 主页
	1	0	0	
	1	1	0	
1	0	0	0	页 0(每页 16K)
	0	1	0	48K 主页
1	0	0		
	1	1	0	

象方法 A 一样，RAM、寄存器以及 EEPROM，都出现在存储器映象中，因此，必须注意避免这些地址，或将 RAM 和寄存器地址移到不同的地址。

图 1-5 是软硬件方法分页示意图

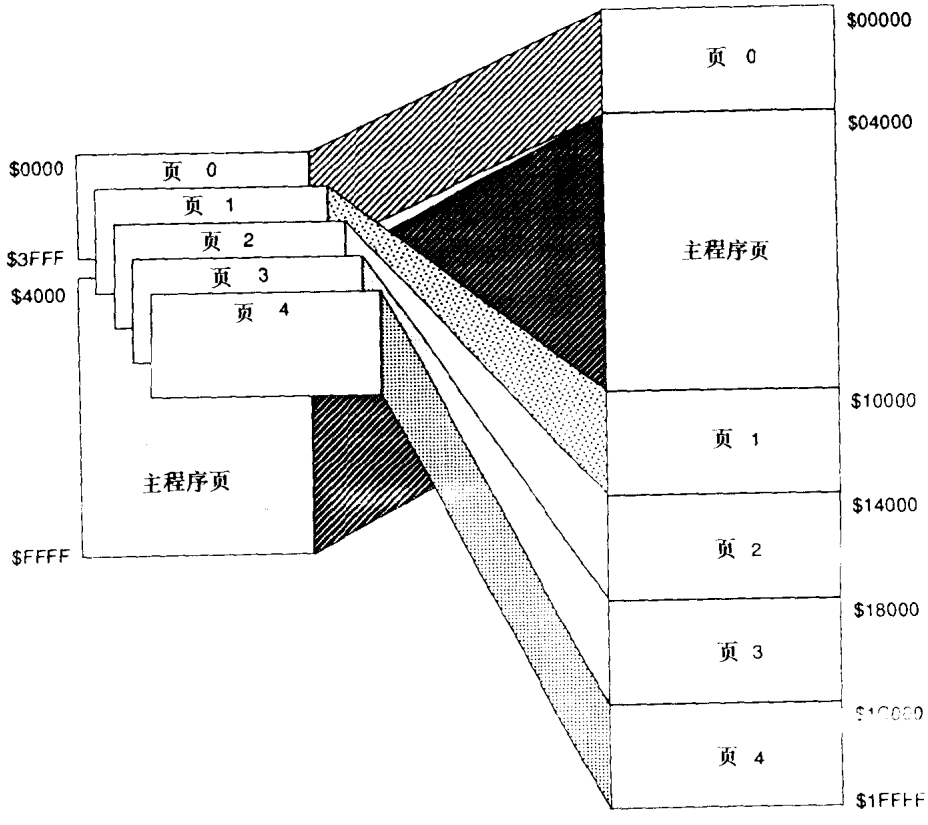


图 1-5 软硬件结合分页方法示意图

可以容易地改变主程序页和各页的大小，以适于不同的应用场合。例如，不用 PD3 控制 EPROM 的地址线 A14，则主程序页为 32K 字节，另外 3 页也是 32K 字节。类似地，增加 I/O 口

线 PD2 来控制地址线 A13，则主程序页为 56K 字节，另有 9 页，每页为 8K 字节。

1.3.4 软硬件结合分页方法程序清单

```
*  128K 字节 EPROM 分为 48K 字节公共页 $4000～$FFFF 和页 0～页 1
*  复位后 PD3～PD5 均为 1，为主程序页和页 0

PROTA     EQU  $00
DDRA      EQU  $01              68HC11G5  only
PORTB     EQU  $04
PORTC     EQU  $06
DDRC      EQU  $07
PORTD     EQU  $08
DDRD      EQU  $09
TMSK2     EQU  $24
TFLG2     EQU  $25
RTII      EQU  $40
RTIF      EQU  $40
PACTL     EQU  $26
DDRA7     EQU  $80              68HC11E9  only
REGS      EQU  $1000
*  RAM 定义
          ORG $0000
TIME      RMB  2          实时中断程序计数器
*
ROMBASE0  EQU  $0200      避免 RAM($0～$1FF)
ROMBASE1  EQU  $4000
VECTORS   EQU  SFFCC
    *     PAGE 0=$00000～$03FFF    (A16=0,15=0,A14=0)
    *     MAIN  =$04000～$0FFFF    (A16=0)
    *     PAGE 1=$10000～$13FFF    (A16=1,A15=0,A14=0)
    *     PAGE 2=$14000～$17FFF    (A16=1,A15=0,A14=1)
    *     PAGE 3=$18000～$1BFFF    (A16=1,A15=1,A14=0)
    *     PAGE 4=$1C000～$1FFFF    (A16=1,A15=1,A14=1)
  START   EQU  $00
  PAGE0   EQU  $20
  PAGE1   EQU  $00
  PAGE2   EQU  $08
  PAGE3   EQU  $10
  PAGE4   EQU  $18
   * 页 0
          org                          ROMBASE0
LOOPP0 BSET PORTA,Y,#$08
   BCLR   PORTA,Y,#$08                 口 A－3 翻转
   JMP    MAIN0                        返回主程序
*  主程序开始
          ORG  ROMBASE1
   RESET  LDS #$01
```

```
            JSR SETUP                初始化 RTI 中断和 DDR
 LOOP       BSET PORTD,Y,# $40
            BCLR PORTD,Y,# $40       主程序使 PD2 翻转
            JSR CHGPAGE0             选页 0
            JMP LOOPP0               口 A－3 翻转
 MAIN0      JSR CHGPAGE1             选页 1
            JSR LOOPP1               口 A－4 翻转
            JSR CHGPAGE2             选页 2
            JSR LOOPP2               口 A－5 翻转
            JSR CHPAGE3              选页 3
            JMP LOOPP3               口 A－6 翻转
 MAIN3      JSR CHGPAGE4             选页 4
            JMP LOOPP4               口 A－7 翻转
 MAIN4      BRA LOOP                 再次开始循环
*初始化程序
 SETUP      SEI
            LDY  # $1000             寄存器地址偏移
            LDAA # $FF
            STAA DDRA+REGS           A 口均为输出(HC11G5)
            STAA DDRD+REGS           D 口均为输出
            CLR  TIME
            CLR  TIME+1
            CLRA
            STAA  PORTA+REGS
            STAA  PORTD+REGS
            LDAA  #%01000000
            STAA  TFLG2+REGS         RTI 标志清零
            STAA   TMSK2+REGS        允许 RTI 中断
            CLI
            RTS
* 实时中断服务程序
RTISRV      LDAA#%01000000
            STAA TFLG2+REGS          RTI 标志清零
            LDAA TIME+1
            STAA PORTB+REGS          计数器存入 B 口
            LDX TMIE                 取时间计数器
            INX
            STX TIME                 计数器值存入 RAM
            RTI                      从中断返回
*换页
CHGPAGE0
            LDAA PORTD+REGS          取 D 口数据
            ANDA #%11000111          使中间 3 位为低电平
            ADDA #PAGE0
            STAA PORTD+REGS          写回 D 口(只有位 3～5 变化)
            RTS
```

```
CHGPAGE0
        LDAA    PORTD+REGS
        ANDA    #%11000111
        ADDA    #PAGE0
        STAA    PORTD+REGS
        RTS
CHGPAGE1
        LDAA    PORTD+REGS
        ANDA    #%11000111
        ADDA    #PAGE1
        STAA    PORTD+REGS
        RTS
CHGPAGE2
        LDAA    PORTD+REGS
        ANDA    #%11000111
        ADDA    #PAGE2
        STAA    PORTD+REGS
        RTS
CHGPAGE3
        LDAA    PORTD+REGS
        ANDA    #%11000111
        ADDA    #PAGE3
        STAA    PORTD+REGS
        RTS
CHGPAGE4
        LDAA    PORTD+REGS
        ANDA    #%11000111
        ADDA    #PAGE4
        STAA    PORTD+REGS
        RTS
```

* 向量

```
ORG     VECTORS
FDB     RESET       EVENT 2
FDB     RESET       EVENT 1
FDB     RESET       TIMER OVERFLOW 2
FDB     RESET       INPUT CAPTURE 6 / OUTPUT COMPARE 7
FDB     RESET       INPUT CAPTURE 5 / OUTPUT COMPARE 6
FDB     RESET       SCI
FDB     RESET       SPI
FDB     RESET       PULSE ACC INPUT
FDB     RESET       PULSE ACC OVERFLOW
FDB     RESET       TIMER OVERFLOW 1
FDB     RESET       INPUT CAPTURE 4 / OUTPUT COMPARE 5
FDB     RESET       OUTPUT COMPARE 4
FDB     RESET       OUTPUT COMPARE 3
```