

突破 64KB 限制的 8051 程序设计

朱民, 鲍远律, 张旺生
中国科学技术大学自动化系

摘要 :本文讨论了单片机程序超过 64KB 空间后的软硬件处理。详细论述了其软件设计思想, 并提供一个可由 ASM51 宏汇编编译, 链接的程序示例。

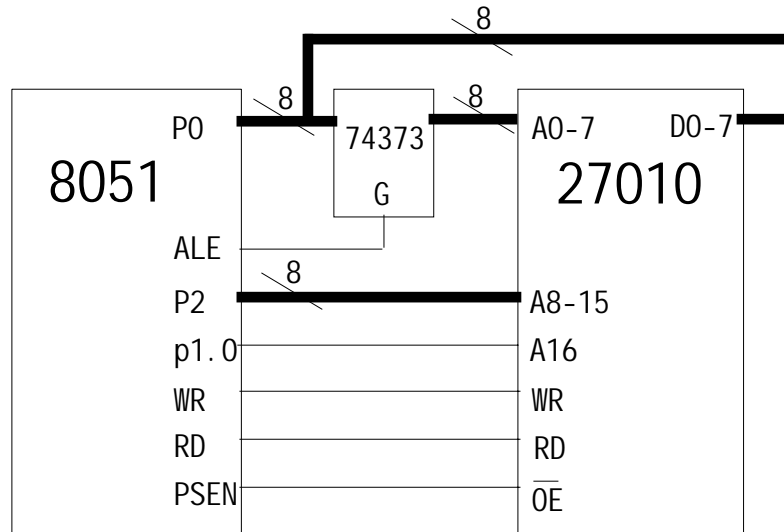
关键词 : 8051 单片机, 区间程序相互调用, 程序空间, ASM51

MCS-51 8 位单片机是应用相当广泛的一类单片机。它由 16 位地址线, 可访问 64KB 的程序和数据空间。对于一般应用而言, 64KB 的容量已够。但是随着应用复杂度增加和对程序功能越来越多的要求, 64KB 的空间有时就显得不够, 必须扩展程序存储器。笔者就遇到这种情况。

一. 硬件设计

程序超过 64KB, 一般有两种处理方法。一是更换 CPU, 选取访问空间更大的 CPU。但这样的话, 程序往往要重写, 费力。第二个方法是再增加一块 EPROM, 占用 8051 的一个 I/O 端口作为片选信号, 从而使 8051 能访问的程序空间超过 64KB。

本文所讨论的系统是用一片 EPROM27010 作为程序存储器, 用 P1.0 作为 EPROM 的高端地址线 A16。其硬件概图如下:



二. 软件设计

与硬件相比, 软件设计难度颇大。P1.0 作为 A16 线, 使 8051 可以访问 128KB 的代码, 但程序计数器 PC 值只有 16 位长, 当在程序中改变 P1.0 的状态时(只有通过改变 P1.0 的状态, 才能访问 128KB 空间), PC 值仅是正常地自增, 下一个取指周期中取出的指令将不是程序地址为: $address = (PC+1)$ 处的代码, 而取的是地址为: $address = (PC+1+10000H) \bmod$

20000H 处的代码，程序变得不可控制。因此必须对软件进行精心设计，使 128KB 程序都能正常运行。

我们将 128KB 的 EPROM 划分为如图 1 所示的 A、B、C、D 四个区域，并让 C、D 两区的代码完全一样。这样当 PC 值介于[0000, 3FFF]时，不论 P1.0 的状态如何程序都能正常运行。也就是说把程序空间分为三块，程序代码空间可达 16KB + 48KB + 48KB = 112KB。

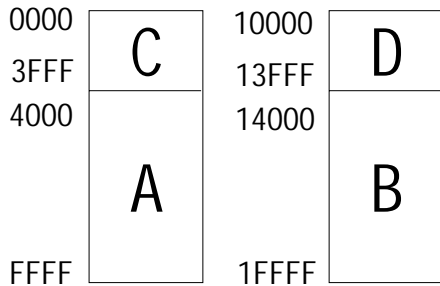


图 1

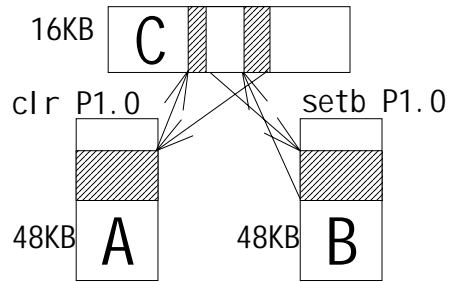


图 2

分区之后，C 区调用 A(B)区的代码只要设置好相应的 P1.0 的状态即可；A(B)区调用 C 区的代码不必特殊处理；A 区与 B 区之间代码不能直接相互调用，必须通过某种机制来连接，这也是划分 C 区的目的。以 A 区调用 B 区代码为例，要遵循 A-C-B-C-A 的调用准则。这样做的不足是降低了程序效率。

要正确实现区间程序调用还必须知道被调用程序的入口地址。一个很自然的想法是在 C 区建立一个地址表，用来存储入口地址，在程序调用前读这个表即可。但是要注意的是：这个表不能建立在程序区，而应让主程序(C 区程序)读取入口地址并建立在片外数据区。否则在程序连接(ASM51)时会因有不能连接确定的地址而出错(除非手工填入地址。这很麻烦)。

在确定了区间代码调用机制后，程序的编写变得相对简单些，但具体的实现还需要一些小技巧。本文提供一个可实现 A、B 区间相互调用，并由 ASM51 宏汇编编译、连接通过的完整例子。该例子由 main.asm, a.asm, b.asm 组成，各相应于 C、A、B 区的代码。main.asm 和 a.asm 编译连接后形成[00000, 0ffff]的写入文件；main.asm 和 b.asm 编译连接后形成[10000, 1ffff]的写入文件。main.asm 集中了中断处理程序、入口地址读入程序(get_address)、区间调用转向(go_abroad)和返回程序(go_home)。另外，划分 a.asm 和 b.asm 时，尽量让它们之间的相关性小，以减少区间代码的相互调用。因为 A、B 区间的

相互调用总是要通过 C 区，付出一定的“过桥费”后实现的。

附程序清单如下：

程序一 . main.asm (C 区程序)

```

name main_asm ;
public go_abroad ; 页间转换子程序
public ramdph,ramdpl ; data
public page1_call_page2,page2_call_page1 ;SRAM 地址, 存储子程序入口地址
NUM equ 20 ; 允许 20 个子程序入口地址
page1_call_page2 xdata 0
page2_call_page1 xdata page1_call_page2+NUM*2

```

```

RamDph  data 30h
RamDpl  data 31h

cseg at 0
org 0
ljmp main
org 0003
ljmp prog_int1      ; 转中段处理
;
main: mov sp,#50h
      mov r7,#0
      mov r6,#0
delay: nop
      djnz r6,delay
      djnz r7,delay  ; 系统延时
      clr p1.0      ; 选择 A 区
      nop
; 读出 A 区子程序入口地址放在 page2_call_page1 处
      mov dptr,#page2_call_page1
      mov b,#0
      lcall get_address
; 读出 B 区子程序入口地址放在 page1_call_page2 处
      setb p1.0
      nop
      mov dptr,#page1_call_page2
      mov b,#0
      lcall get_address
;
      clr p1.0
      nop          ; 选择 A 区
      lcall 4100h  ; lcall page1_main
      sjmp $

;-----
; 入口: 区属性已设置
;      B=0 , dptr = 目标存储地址
get_address: push dph
             push dpl      ; 将目标地址压栈
             mov dptr,#4000h
             mov a,b
             movc a,@a+dptr
             pop dpl
             pop dph
             movx @dptr,a
             inc b

```

```

    mov a,b
    cjne a,#NUM*2,get_address ; 读取 NUM 个入口地址
    ret
;-----
; A,B 区间相互调用子程序
go_abroad:
    mov dptr,#go_home
    push dpl
    push dph ; 准备返回地址
    cpl p1.0
    nop ; A->B 或 B->A
    mov dph,ramdph
    mov dpl,ramdpl ; 目标调用子程序入口地址
    clr a
    jmp @a+dptr ; 到目标子程序
;-----
; A,B 间相互调用后,返回原区子程序
go_home:
    cpl p1.0
    nop ; 返回原区
    ret
;-----
prog_int1:ret
db 'main_end' ; just check
end

```

程序二 . a.asm(A 区程序)

```

name page1_program
extrn code(go_abroad)
extrn data(RamDph,RamDpl)
extrn number(page1_call_page2)

cseg at 4000h
dw page1_main ; 被调用子程序入口表
dw page1_sub

cseg at 4100h
page1_main:
    lcall page1_sub ; 调用本区程序,不必换区
; 调用 B 区程序
    mov dptr,#page1_call_page2
    movx a,@dptr
    mov RamDph,a ; PC 高 8 位
    inc dptr

```

```

    movx a,@dptr
    mov RamDpl,a      ; PC 低 8 位
    lcall go_abroad  ; 返回公共区
    ret
;
page1_sub: ret
db 'page1_end'      ; for check eprom
end

```

程序三 . b.asm(B 区程序)

```

name page2_program

extrn code(go_abroad)
extrn data(RamDph,RamDpl)
extrn number(page2_call_page1)

cseg at 4000h
dw page2_main
dw page2_sub

page2_main:ret
page2_sub: ret
db 'page2_end' ; for check
end

```

参考文献：

- [1]孙涵芳，徐爱卿，《MCS-51,96 单片机原理及应用》，北京航空航天大学出版社
- [2]綦希林，陈嘉庆，魏世明，唐铭新，《ASM51 单片机程序设计语言》，《微计算机信息》编辑部