

# CYGNAL 应用笔记

## AN006 — 外部 SRAM 与 C8051F000 接口

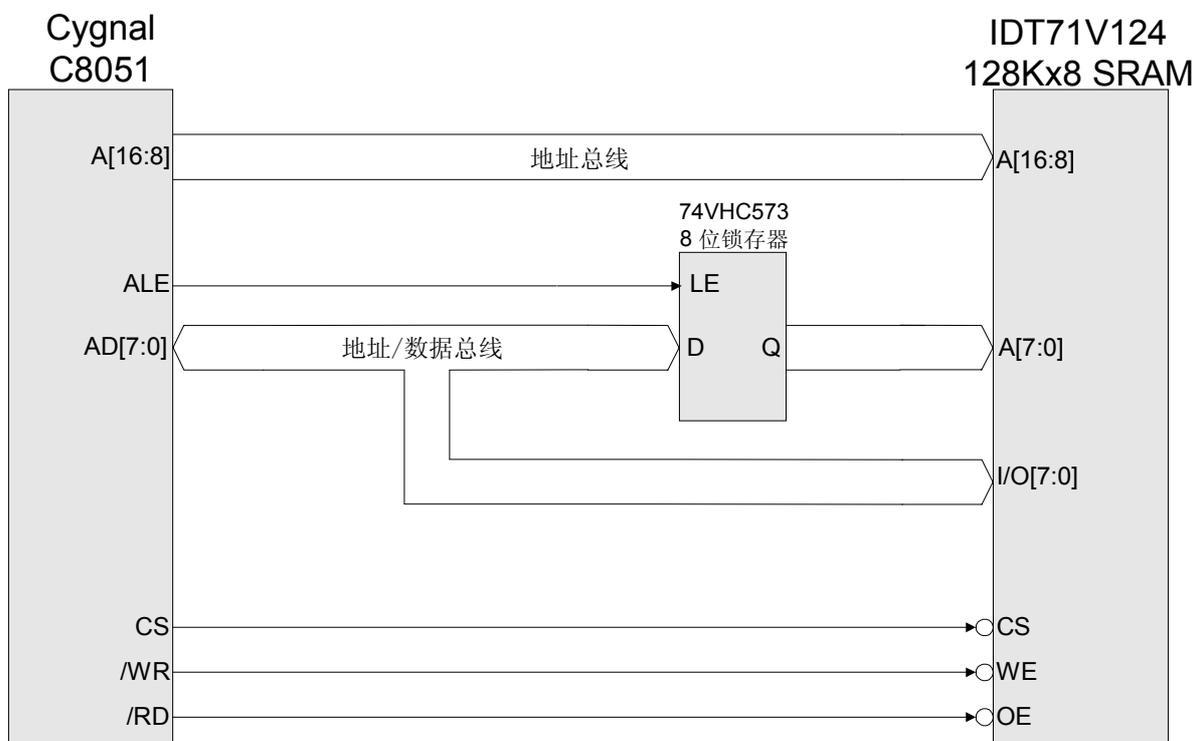


图 1. 外部 SRAM 接口框图

### 引言

本应用笔记的目的是介绍如何将一个通用 SRAM 芯片或一个按存储器寻址的外设部件与 C8051 器件通过通用 I/O 引脚连接。本文给出了硬件连接框图、电路原理图、时序图、示例代码和性能评价。

这种接口方式的应用包括 A/D 转换数据的采集、数据记录或任何其它涉及大量数据存储的应用。

### 关键点

- 该参考设计假定使用 10ns 的 SRAM。如果所用 SRAM 的访问时间大于 45ns，则需要加入 NOP 指令以延长地址建立时间及读/写选通时间。
- 需要的引脚数取决于所支持的地址空间。对于本设计的 128K 字节地址空间，需要 21 个引脚。

CYGNAL Integrated Products, Inc.

4301 Westbank Drive  
Suite B-100  
Austin, TX 78746  
[www.cygnal.com](http://www.cygnal.com)

Copyright ©2001 Cygnal Integrated Products, Inc.  
(版权所有)

沈阳新华龙电子有限公司  
沈阳市和平区青年大街 284 号 58 号信箱  
电话: (024) 23930366, 23940230  
电邮: [longhua@mail.sy.ln.cn](mailto:longhua@mail.sy.ln.cn)  
网址: [www.xhl.com.cn](http://www.xhl.com.cn)

## AN006 — 外部 SRAM 与 C8051F000 接口

---

- 如果设计中使用 SRAM，一定要向供应商查询产品的供货情况。生产厂家正逐步淘汰很多低容量的 SRAM 器件。

### 说明

这个外部 SRAM 接口示例使用一个 Integrated Device Technologies ([www.idt.com](http://www.idt.com)) 生产的 IDT71V124SA10PH (128K X 8 位) 3V SRAM，任何一个通用 SRAM 器件都可以在类似的方式下工作。接口部分采用地址和数据总线复用的方式以减少所需要的端口引脚数。当传输数据时，低位地址保持在一个锁存器中。第 5 页中的图 4 给出了该实现方案经过验证的配置图。

### 双向端口操作

‘Data1’用作数据输入总线、输出总线和部分地址总线。对总线的复用需要对端口的配置进行动态改变，使端口按需要设置为输入或输出。

为了将一个端口引脚配置为输入，必须将其相应的端口配置寄存器位 (PRTnCF.x) 设置为 ‘0’，使其输出方式为 ‘漏极开路’，寄存器锁存位 (Pn.x) 必须设置为 ‘1’，使其输出状态为高阻态。例如，下面的代码将端口 0 的所有引脚配置为输入：

```
mov    PRT0CF, #00h    ;漏极开路输出方式
mov    P0, #0ffh      ;高阻抗
```

下面的代码将端口 0 的所有引脚配置为推挽输出方式：

```
mov    PRT0CF, #0ffh  ;推挽输出方式
```

SRAM\_Read 子程序 (见示例代码部分) 给出改变端口方向的一个例子。在程序执行的前一阶段，‘DATA1’口被配置为输出，将低字节地址输出到端口锁存器。在程序执行的第二阶段，‘DATA1’口被配置为输入，从外部 SRAM 读取数据。

### 信号和接线

图 1 给出了 C8051F000、SRAM、地址锁存器之间的硬件连接框图。整个电路原理图见图 4。连线、标识及信号名说明如下：

在示例代码中以 ‘DATA1’ 为标识的复用地址/数据总线 ‘AD[7..0]’ 支持低 8 位地址和 8 位数据。这种配置使得在 SRAM 与 C8051 之间传输数据时低位地址保持在锁存器 ‘573’ 中，这样就不需要再为数据传输增加 8 位端口。

在示例代码中以 ‘ADDR’ 为标识的 ‘A[15..8]’ 提供高 8 位地址。

在示例代码中以 ‘A16’ 为标识的 ‘A16’ 作为两个 64KB 存储块的选择信号，为 ‘0’ 时选择 1 块，为 ‘1’ 时选择 2 块。

‘RD’、‘WR’、‘ALE’ 和 ‘CS’ 为控制信号，在示例代码中使用相同的名字。‘RD’ 是读选通信号 (低电平有效)。「WR」是写选通信号 (低电平有效)。「ALE」是地址锁存信号，用于在数据传输期间保持低 8 位地址。「CS」是 SRAM 的片选信号 (低电平有效)。

### 软件操作

‘SRAM\_Init’、‘SRAM\_Read’和‘SRAM\_Write’是用于访问外部 SRAM 的三个子程序。

‘SRAM\_Init’程序初始化 SRAM 接口逻辑和端口配置。该程序只在器件的初始化过程中被调用。该程序假定交叉开关已经被允许 (XBR2.6=1)。例如：

```
mov    XBR@, #40h    ; 允许交叉开关
acall  SRAM_Init    ; 初始化 SRAM
```

‘SRAM\_Read’程序从外部 SRAM 读一个字节。调用该程序之前，先将 DPTR 装入 16 位的待读地址，然后调用 ‘SRAM\_Read’。该程序将从 DPTR 所指向的地址读到的数据返回到 ACC。例如：

```
mov    DPH, #00h    ; 装入高字节地址
mov    DPL, #00h    ; 装入低字节地址
acall  SRAM_Read    ; 进行读操作，数据返回到 ACC
```

‘SRAM\_Write’程序将 ACC 中的一个字节写入到外部 SRAM，地址由 DPTR 指定。调用该程序之前，将待写数据装入 ACC，将 DPTR 装入 16 位的待写地址，然后调用 ‘SRAM\_Write’。例如：

```
mov    DPH, #00h    ; 装入高字节地址
mov    DPL, #00h    ; 装入低字节地址
mov    a, #55h      ; 装入待写数据
acall  SRAM_Write   ; 进行写操作
```

示例代码中的主程序概述了如何对该外部 128KB SRAM 的每一个字节进行读写。该程序向外部 SRAM 写入一个字节，再从写入的地址读回，然后比较回读的值与写入的值是否一致。程序接着处理下一个地址，直到整个 64K 的存储块写完。一旦低存储块写完，程序将 ‘A16’ 位置 1（见示例代码中“常数和声明”一节），切换到高存储块。程序将接着对高存储块的每个字节进行同样的读、写和校验操作。

### 时序说明

图 2 和 3 分别给出按示例代码实现的读和写时序波形。表 1 给出这些图形中的时间值。

#### 读时序说明

‘tRDSU’（表 1）是指从读选通信号有效到输出数据有效的的时间。与这一时序相对应的代码为：

```
clr    RD    ;使读选通有效
;NOP      ;加入 NOP 指令延长 tRDSU
mov    a, DATA ;读数据
```

如上所示，为了满足 SRAM 的建立时间，需要在 ‘clr RD’ 之后加入 NOP 指令。

### 写时序说明

如表 1 所示，‘tWR’ 是指 ‘/WR’ 脉冲的宽度。下面的代码段用于产生这一脉冲。

```
clr    WR    ;使写选通有效
;NOP          ;加入 NOP 指令延长 tWRSU
setb   WR    ;使写选通无效
```

如上所示，为了满足 SRAM 的建立时间，需要在 ‘clr WR’ 之后加入 NOP 指令。

### 性能

这一多路复用的并行接口实现方案使用一般的 I/O 端口，却达到了较高的数据传输率。从读或写过程的进入到返回，一个字节读或字节写占用 34 个 SYSCLK 周期，对于 20MHz 的 SYSCLK，相当于 1.7 $\mu$ s。这就是说可以达到的最大传输率为 588K 字节/秒。一个 64K 的存储块可以在 137 $\mu$ s 内读（或写）完。

## AN006 — 外部 SRAM 与 C8051F000 接口

符号	参数	周期	SYSCLK=20MHz
<b>读周期</b>			
tALE	锁存脉冲宽度	2	100ns
trDSU	数据建立时间	2	100ns
<b>写周期</b>			
tALE	锁存脉冲宽度	2	100ns
twASU	地址建立时间	3	300ns
twDSU	数据建立时间	4	200ns
tWR	写脉冲宽度	2	100ns

表 1. 读和写周期时序

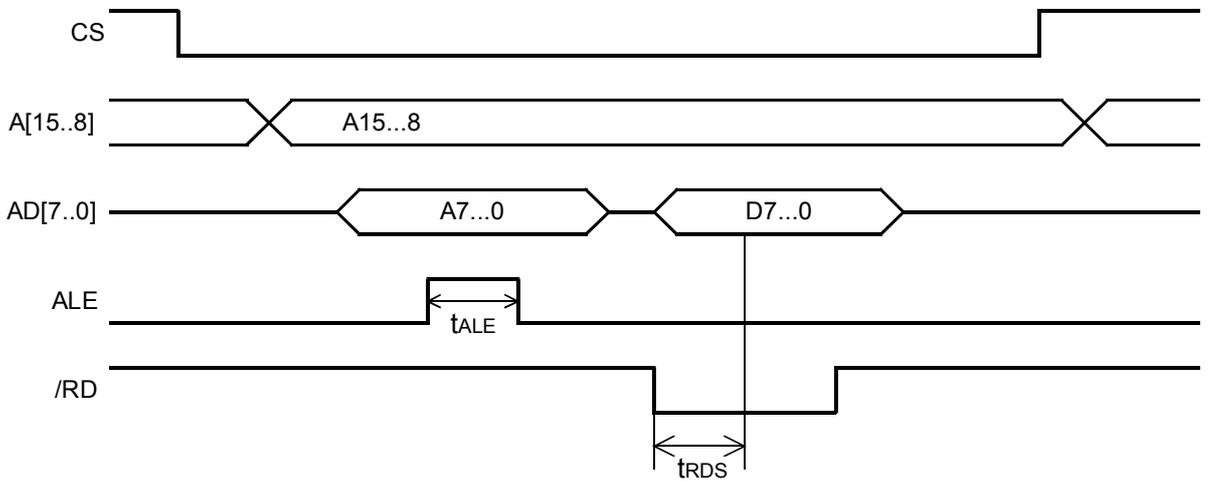


图 2. 读周期时序波形

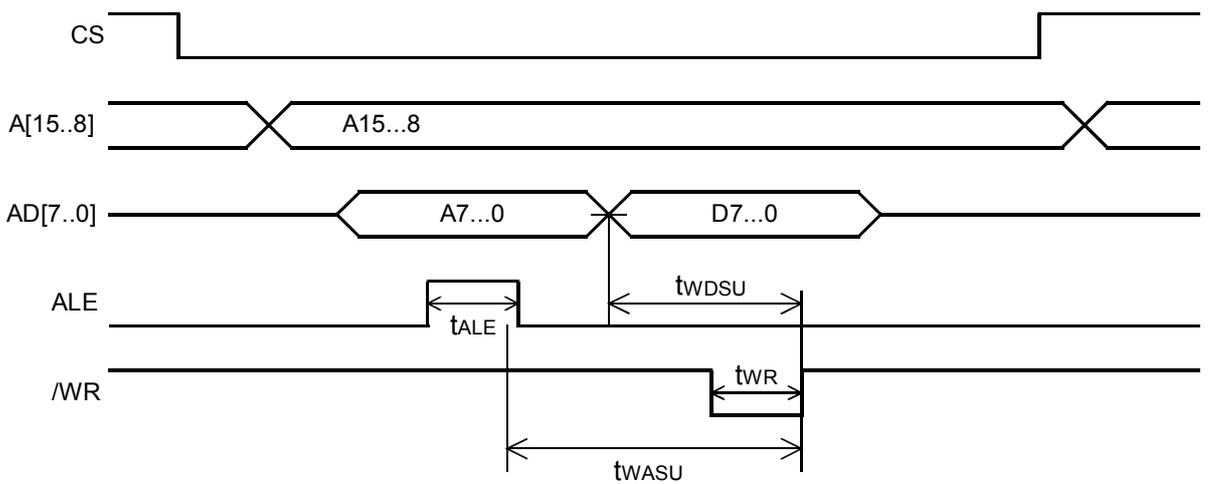


图 3. 写周期时序波形

## AN006 — 外部 SRAM 与 C8051F000 接口

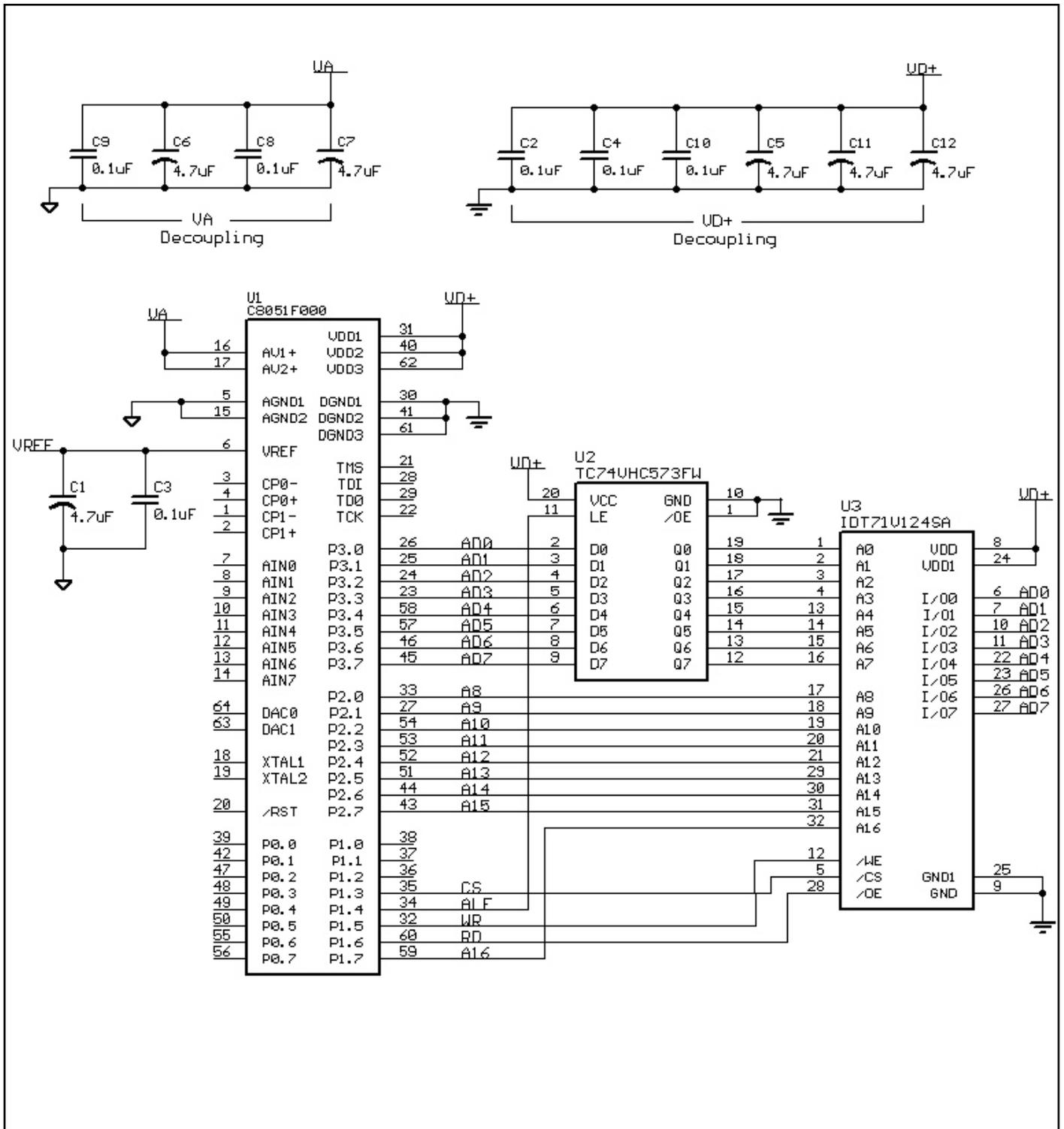


图 4. 经过测试的 C8051F000、128k x 8 SRAM 和地址锁存器配置

## AN006 — 外部 SRAM 与 C8051F000 接口

---

### 示例代码

```
-----  
; Copyright (C) 2000 CYGNAL INTEGRATED PRODUCTS, INC.  
; 版权所有  
;  
;  
; 文件名      : Sram.ASM  
; 目标 MCU    : C8051F000  
; 说明        : 外部 SRAM 读/写验证程序 (对 IDT 71V124SA).  
;  
-----  
; 等价定义  
-----  
$NOLIST  
$MOD8F000  
$LIST  
  
-----  
; 常量和声明  
-----  
  
DATA1      EQU    P3           ; 数据端口引脚 (AD7..0)  
DATA1CF    EQU    PRT3CF      ; 端口配置寄存器 (对数据)  
ADDR       EQU    P2           ; 地址端口引脚 (A15..8)  
ADDRCF     EQU    PRT2CF      ; 端口配置寄存器 (对地址)  
A16        EQU    P1.5        ; 最高地址位 (地址块选择)  
RD         EQU    P1.4        ; 读选通 (低电平有效)  
WR         EQU    P1.3        ; 写选通 (低电平有效)  
ALE        EQU    P1.2        ; 地址锁存信号 (低电平有效)  
CS         EQU    P1.1        ; SRAM 片选信号 (低电平有效)  
  
-----  
; 变量  
-----  
  
-----  
; 复位和中断向量  
-----  
; 复位向量  
  
        org    00h  
        ljmp   Main  
  
-----  
; 主程序代码  
-----  
  
        org 0B3h  
Main:
```

## AN006 — 外部 SRAM 与 C8051F000 接口

---

; 禁止看门狗定时器, (中断尚未被允许)。如果中断被允许, 我们应明确禁止中断,  
; 以保证下面的两条指令能在 4 个时钟周期内执行完。

```
mov    WDTCN, #0DEh
mov    WDTCN, #0ADh

; 设置交叉开关
Mov    XBR2, #40h      ; 弱上拉, 允许交叉开关
lcall  SRAM_Init      ; 初始化 SRAM

mov    R0, #0ffh
mov    DPH, #00h      ; 初始化 16 位地址
mov    DPL, #00h      ;
mov    a, R0          ; 装写入值

; Loop 将写一个值到 SRAM, 然后读出, 与写入值比较
loop:
lcall  SRAM_Write     ; 写入 SRAM
clr    a              ; 清除装入值
lcall  SRAM_Read      ; 读同一地址
cjne   a, 00h, error  ; 检查读到的值是否与写入值相同
inc    dptr           ; 下一地址
mov    a, DPH         ; 检查 DPTR, 判断是否结束 (dptr=0000h)
orl    a, DPL         ;
jz     bldone         ; 如果 DPTR 循环回到初始值, 说明我们已结束
; 了对第一个 64K 块的操作。

mov    a, R0          ; 重装写入值
jmp    loop           ; 再次进行写, 读和验证

bldone:
orl    P1, #00111010b ; 转到对存储块 2 操作
mov    R0, #0ffh
mov    DPH, #00h      ; 初始化 16 位地址
mov    DPL, #00h      ;
mov    a, R0          ; 装写入值

; Loop1 将写一个值到 SRAM, 然后读出, 与写入值比较
loop1:
lcall  SRAM_Write     ; 写入 SRAM
clr    a              ; 清除装入值
lcall  SRAM_Read      ; 读同一地址
cjne   a, 00h, error  ; 检查读到的值是否与写入值相同
inc    dptr           ; 下一地址
mov    a, DPH         ; 检查 DPTR, 判断是否结束 (dptr=0000h)
orl    a, DPL         ;
jz     b2done         ; 如果 DPTR 循环回到初始值, 说明我们已结束
; 了对第一个 64K 存储块的操作。

mov    a, R0          ; 重装写入值
jmp    loop1          ; 再次进行写, 读和验证
```

## AN006 — 外部 SRAM 与 C8051F000 接口

---

```
b2done:   jmp     $           ;
error:   jmp     $           ; 验证时发现错误

;-----
; SRAM_Init
;-----
; 该程序初始化 SRAM 接口逻辑。必须在任何 SRAM_Read 或 SRAM_Write 操作之前
; 被调用一次，可作为复位操作的一部分。该程序假定交叉开关已经被允许(XBR2.6 = '1')。
;

SRAM_Init:
    mov     DATACF, #00h       ; 允许端口 3(DATA)作为输入总线
    mov     DATA1, #0ffh
    mov     ADDR, #0ffh        ; 允许端口 2(ADDR)作为输出
    mov     ADDR, #0ffh        ; 驱动为高电平 ($ff)
    orl     PRT1CF, #00111110b ; 允许 P1.7..3 作为输出
    anl     P1, #11011011b     ; A16 = '0'; ALE = '0'   存储块 1
    orl     P1, #00011010b     ; RD, WR, CS = '1'
    ret

;-----
; SRAM_Read
;-----
; 该程序读外部 SRAM。从 DPTR 指向的地址读取数据存到 ACC。此处未处理存储块选择(操作 A16)
;

SRAM_Read:
    clr     CS                 ; 选择 SRAM
    mov     ADDR, DPH          ; 输出外部地址 A15..A8
    mov     DATACF, #0ffh    ; 允许 AD7..0 为输出
    mov     DATA1, DPL       ; 输出外部地址 A7..A0
    setb    ALE                ; 锁存该地址
    clr     ALE
    mov     DATACF, #00h     ; 允许 AD7..0 为输入
    mov     DATA1, #0ffh
    clr     RD                 ; 发出读选通
    mov     a, DATA1         ; 读数据(注: 对于这片 SRAM, 读操作建立时间
                                ; 为 45ns。在 SYSCLK = 20MHz 时, 该指令需
                                ; 两个时钟周期, 或 50ns * 2 = 100ns。
    setb    RD                 ; 使读选通无效
    setb    CS                 ; SRAM 片选无效
    ret

; 一个读操作共需:
; 30 字接, 34 个周期。

;-----
; SRAM_Write
;-----
```

---

## AN006 — 外部 SRAM 与 C8051F000 接口

---

```
; 该程序写一个字到外部 SRAM。将 ACC 中的字节写到 DPTR 指向的地址。
; 此处未处理存储块选择(操作 A16)。
;
SRAM_Write:
    clr    CS                ; 选择外部 SRAM
    mov    ADDR, DPH        ; 输出外部地址 A15..A8
    mov    DATACF, #0ffh    ; 允许 AD7..0 为输出
    mov    DATA1, DPL      ; 输出外部地址 A7..A0
    setb   ALE              ; 锁存该地址
    clr    ALE
    mov    DATA1, a        ; 将数据送到数据总线
    clr    WR               ; 启动写操作
    setb   WR               ; 写选通无效
                                ; 注: 在 sysclk = 20MHz 时, 这种方式将产生 100ns
                                ; 宽的写脉冲, 该 SRAM 所要求的最小值为 7ns。
    mov    DATACF, #00h    ; 允许 AD7..0 为输入
    mov    DATA1, #0ffh
    setb   CS               ; SRAM 片选无效
    ret
;-----
; 文件结束。

END
```