

常用存储器芯片设计指南

现代通讯产品中，各种存储器的应用已经是越来越广泛，可以这么说，产品中包含的存储器的特性的好坏，直接关系到产品整体性能。因此，存储器芯片的设计，在通讯产品的设计中，也显得愈发重要。

目前在通讯产品中应用的存储器，主要有 FLASH、SSRAM、SDRAM、串行 PROM 等，由此延伸出去还有在接口电路中经常应用的 FIFO、双口 RAM 等，下面的内容就是这些常用存储器芯片的原理介绍和在产品中的设计指南。

FLASH 介绍

一、BOOT ROM 简介

我们在 CPU 最小系统中一般采用 AM29LV040B-90 // SST39VF040-90-4C-NH （代码：10300067，512kB，8 位总线宽度，PLCC32 封装，3.3V 供电）作为 BOOT ROM。

BOOT ROM 中存放的是系统自举程序，实现 CPU 系统的自举。当系统上电后，CPU 首先运行 BOOT ROM 中的程序，完成对 CPU 系统的初始化。

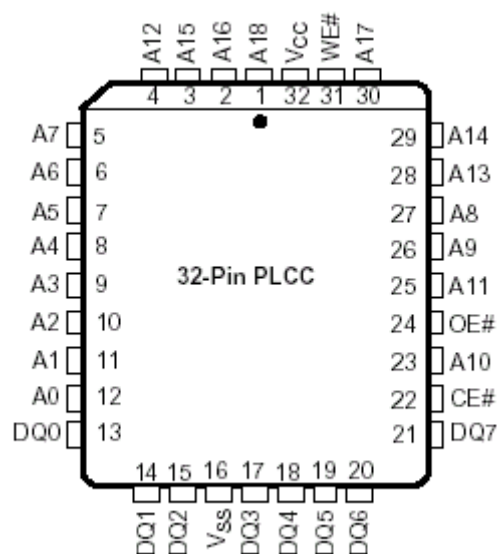


图 1 AM29LV040B-90 // SST39VF040-90-4C-NH 引脚图

该 FLASH 芯片可在线读写，但作为 BOOT ROM 时，我们一般用烧录机烧写入程序，不对其进行在线写。其读操作时序如图 2 所示。

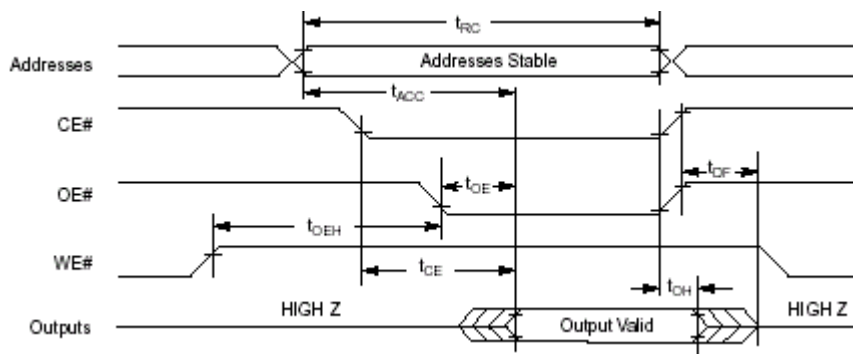


图 2 读操作时序

下面给出一个 MPC860 最小系统的应用例子。

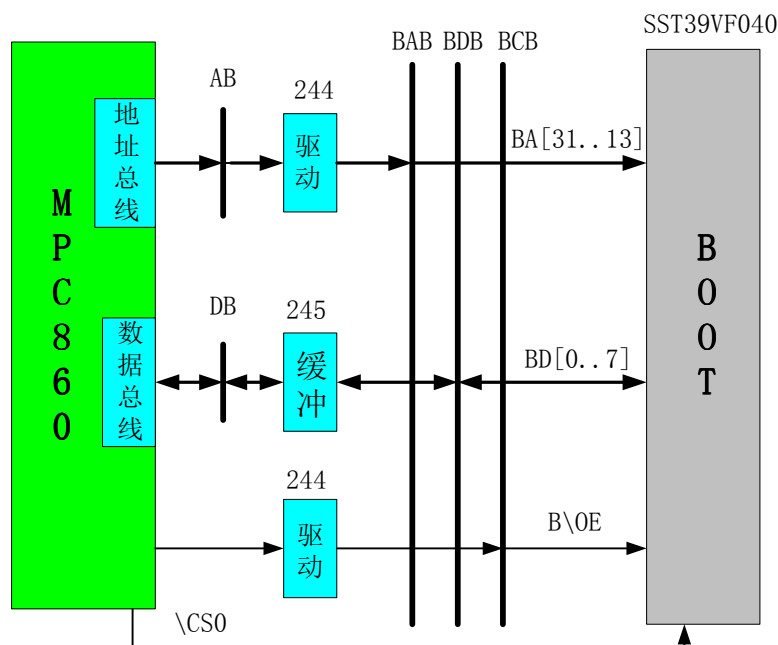


图 3 MPC860 BOOT 电路图

因为我们不需要在线写，所以为防止 BOOT FLASH 的程序被改写，一般将 /WE 信号接高电平。

MPC860 用 8 位数据口的方式访问 BOOT，经缓冲之后的数据线为 BD00-BD07。MPC860 地址线使用 A31-A13，经一级驱动与 BOOT 相连。使用 /CS0 片选端，地址范围 0x0800 0000~0x0807 FFFF，使用内部等待，等待周期为 8。

BOOT ROM 中存放的是系统自举程序，实现 MPC860 系统的自举。当系统上电后，MPC860 首先运行 BOOTROM 中的程序，该程序首先完成 MPC860 的初始化，然后根据参数，将 Flash ROM 中的应用程序复制到 SDRAM 空间中，然后将控制权移交给该应用程序运行；或准备应用程序加载，进入调试状态。

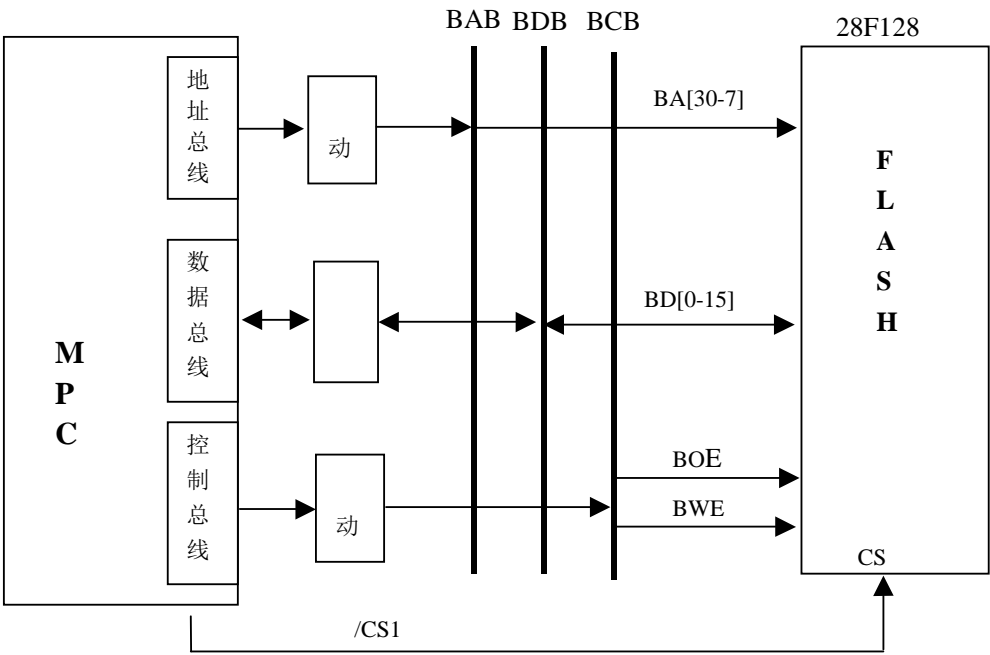
二：大容量 FLASH

由于 FLASH 具有在掉电情况下保持数据和容量大的特性，因此在公司的绝大多数运用中是用在 CPU 系统中存放系统的应用软件的，其运作过程如下：系统上电后，首先 BOOTROM 的片选被选中，它里面放的是 CPU 的初始化程序，这样 CPU 就起来了，接着 FLASH 条的版本程序被下载到内存条中，从而整个软件就在 CPU 系统中运行起来，这样一来可以提高系统运行速度，二来是方便版本的管理。因为一般来说系统支持远程加载和更新软件版本的功能，因此一般说来 FLASH 中必须开辟两个相同的区域，分别作为软件版本的保护和备份，这样一旦出现在系统动态加载软件失败时，能保证备用软件版本的正常启动，使系统不致崩溃。

典型介绍

生产 FLASH 的厂商很多，我们将以广为运用 INTEL 的 28F128J3A 芯片为例子来介绍，使大家对 FLASH 的操作有一个大致的了解，此芯片单片容量为 128MBIT,(16BIT X 8M)。由于此系列的 FLASH (28F128J3, 28F640J3, 28F320J3) 的引脚完全兼容，因此可以简单地通过器件替换提供 128MBIT,64MBIT,32MBIT 的容量。

28F128J3A 芯片的管脚简单，分数据总线，地址总线，控制线，电源，地这么几类；在使用的时候，只需接出使能，片选，写控制三根控制线即可，其他的控制线可以接固定电平，下图就是某 CPU 子卡的 FLASH 接法：



FLASH 28F128J3A 的操作是通过 CPU 分 来 现的，其 是通过数据总线，地址总线，控制线 现某个固定电平 ；下 就是 28F128J3A 有 对应的 的定 ：

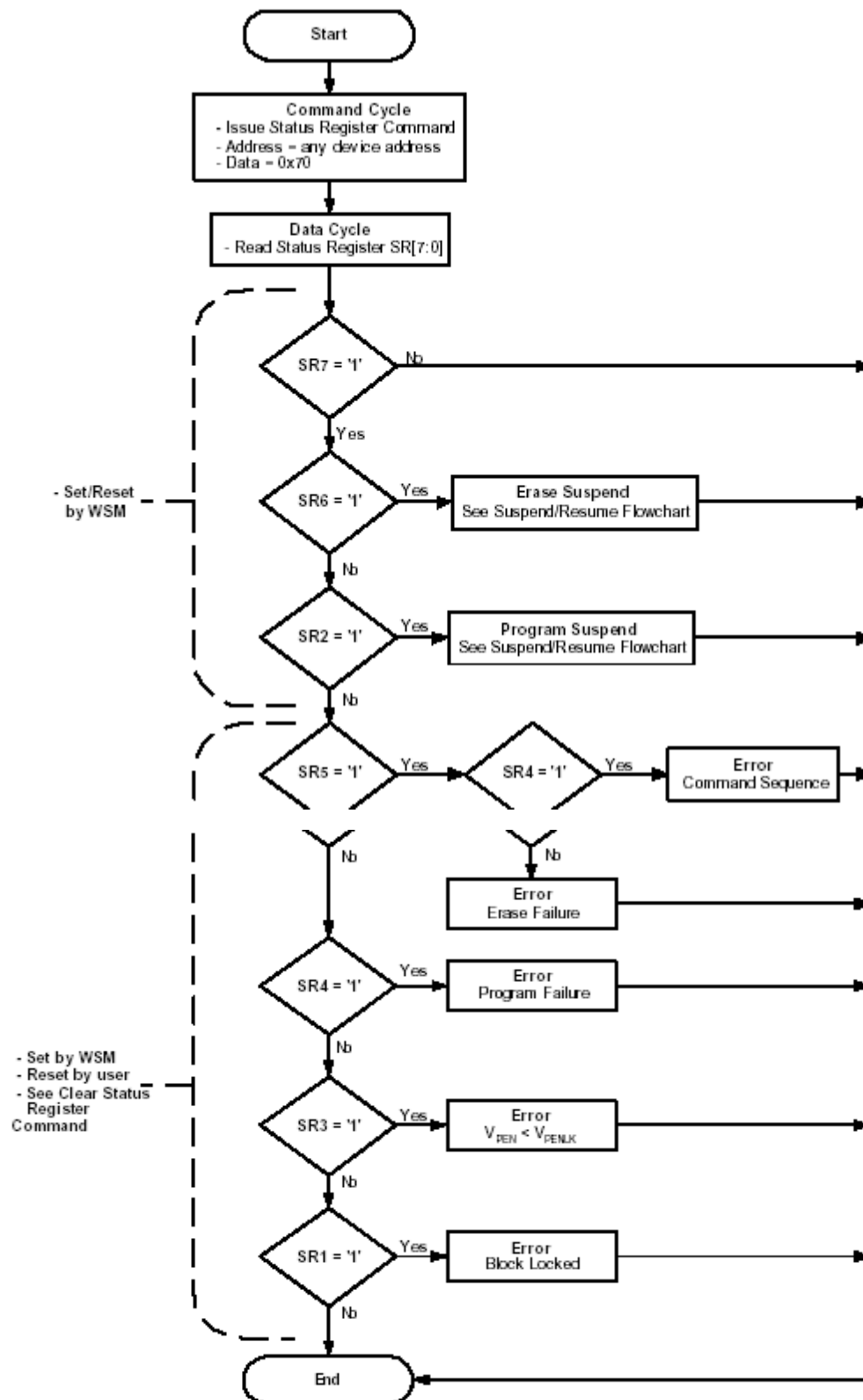
Command	Scalable or Basic Command Set ⁽²⁾	Bus Cycles Req'd.	First Bus Cycle			Second Bus Cycle			Notes
			Oper ⁽³⁾	Addr ⁽⁴⁾	Data ^(5,6)	Oper ⁽³⁾	Addr ⁽⁴⁾	Data ^(5,6)	
Read Array	SCS/BCS	1	Write	X	0xFF				1
Read Identifier Codes	SCS/BCS	≥ 2	Write	X	0X90	Read	IA	ID	1,7
Read Query	SCS	≥ 2	Write	X	0x98	Read	QA	QD	1

Read Status Register	SCS/BCS	2	Write	X	0x70	Read	X	SRD	1,8
Clear Status Register	SCS/BCS	1	Write	X	0x50				1
Write to Buffer	SCS/BCS	> 2	Write	BA	0xE8	Write	BA	N	1,9,10,11
Word/Byte Program	SCS/BCS	2	Write	X	0x40 or 0x10	Write	PA	PD	1,12,13
Block Erase	SCS/BCS	2	Write	BA	0x20	Write	BA	0xD0	1,11,12
Block Erase, Program Suspend	SCS/BCS	1	Write	X	0xB0				1,12,14
Block Erase, Program Resume	SCS/BCS	1	Write	X	0xD0				1,12
Configuration	SCS	2	Write	X	0xB8	Write	X	CC	1
Set Block Lock-Bit	SCS	2	Write	X	0x60	Write	BA	0x01	1
Clear Block Lock-Bits	SCS	2	Write	X	0x60	Write	X	0xD0	1,15
Protection Program		2	Write	X	0xC0	Write	PA	PD	1

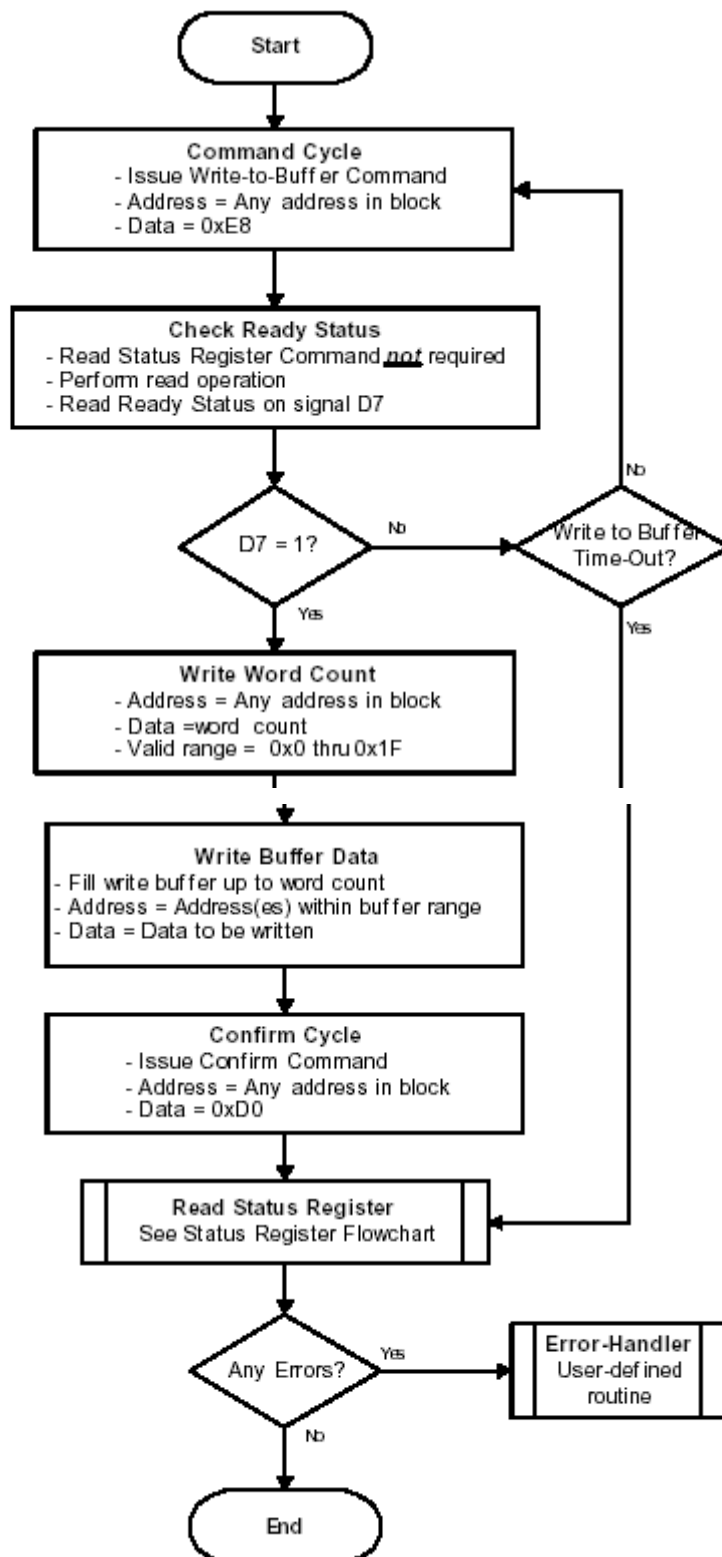
从上表可以看出28F128J3A大部分命令的实现分为两步实现：例如Block_Erase命令，而普通的读命令Read_Array只要一步。

对于28F128J3A的某些操作，是非常简单的，例如读操作：在芯片复位/上电后芯片默认为Read_Array模式，这样可以直接读取芯片的数据，但是如果在写FLASH或发布了其他命令后再想读取，那么就必须重新发布Read_Array命令才能读取；对于其他的操作，尤其是分两步完成的命令，其操作是通过几个不同命令组合而成，具有一定的流程，并且在过程中经常需要不断地从FLASH读取状态信息以进行下一步操作来保证操作的正确。

在这里举两个操作流程的例子，通过这个例子，可以大致了解FLASH的操作：第一个是Read_Status_Register命令，这个命令是用来读取芯片目前的状态，其他操作的流程中经常用到此命令以保证操作的正确，下图是进行Read_Status_Register命令的流图：



第二个例子是使用Write_to_Buffer命令过程，我们可以看到，在其过程中，我们可以看到需要发布Read_Status_Register命令检查芯片的状态信息以判断下一步：



FPGA 配置用 PROM 简介

这类可编程 ROM 是专为 FPGA 加载配置所用。其针对不同容量、型号、厂家的 FPGA，其存储大小、性能不尽相同；但其使用的电路、方法、时序基本相同。而且，基本上是只可一次性擦写。下面就以 XILINX 公司 XC17S00A 系列为例介绍一下。

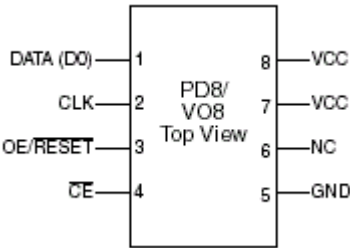


图 1 XC17S00A 引脚图

FPGA 的配置数据是事先烧入 PROM 中的，FPGA 采用的是主串行配置方式，上电后 FPGA 产生配置时钟给 PROM，PROM 则按照图 3 的时序对 FPGA 进行配置，当配置完成后 FPGA 的 DONE 信号就不使能 PROM。

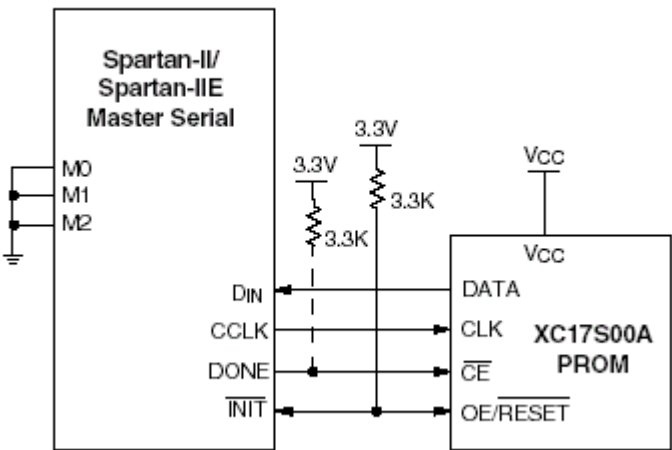


图 2 PROM 配置 FPGA 电路

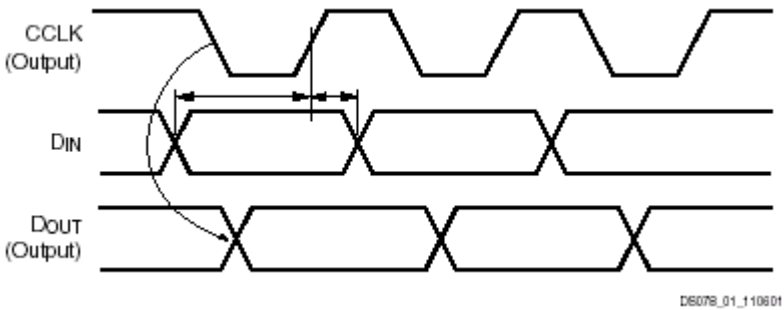


图 3 PROM 的配置时序

我们公司这类器件没有一个是通用的，我们不推荐使用 PROM 配置这种方式。最好用

CPU 进行 FPGA 配置，这样做增加了 FPGA 版本的灵活性，也大大降低了成本。

SSRAM 介绍

SSRAM 支持高性能 CPU、DSP、网络套片等多种应用场合。目前在公司多种产品中也有比较广泛的应用。

一、SSRAM 的选用

在选用 SSRAM 时，一般需要考虑以下几个方面：

- （1）根据设计需求、接口芯片的具体要求等确定 SSRAM 的芯片容量大小、数据宽度以及芯片速度等级；
- （2）根据以上信息确定 SSRAM 型号，并尽可能在公司通用件库中选型；
- （3）根据信号定义及信号时序等确定 SSRAM 及其接口芯片的具体的电路连接关系。

二、SSRAM 的电路设计

1、概述

一般来说，SSRAM 的信号主要包括以下几部分：数据线、地址线、时钟以及写使能、输出使能等控制信号等。在进行具体的电路设计时，一般只需将这些主要信号与其接口的特定芯片的对应信号直接对连即可，对于一些比较特殊的时序要求可以通过 EPLD 内做逻辑实现。对于 SSRAM 上的一些不用的输入信号进行适当的上、下拉处理。

2、应用实例

下面以 RNC 中的一块单板 ASC 上采用的 SSRAM 为例，来说明 SSRAM 的具体应用。

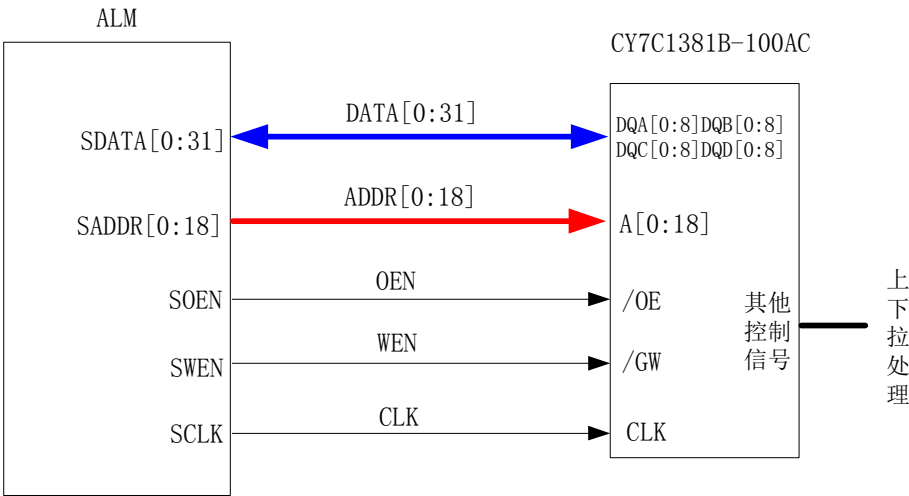
首先根据设计要求，ASC 单板上的 ATM 层 UNI/NNI 管理接口芯片 ALM 需要一片数据宽度为 32 位的 SSRAM 作为外部存储器。综合考虑 ALM 外部存储器所需的容量大小以及目前 SSRAM 的价格因素及芯片可采购性等各方面因素，可以选择公司通用件 CY7C1381B-100AC（代码 12300217）来实现。

CY7C1381B-100AC 数据宽度为 32 位，其信号线主要包括地址线、数据线、时钟和一些控制信号，其具体信号如下表所示。

信号类型	信号名	信号说明	处理方式
数据线	DQA[0: 8]	数据线	与其接口芯片直连
	DQB[0: 8]		
	DQC[0: 8]		
	DQD[0: 8]		
地址线	A[0: 18]	地址线	与其接口芯片直连
时钟线	CLK	时钟输入，用来捕获芯片的所有同步输入	与其接口芯片直连

主要 控制信号	/OE	输出使能	与其接口芯片直连
	/GW	全局写使能输入	与其接口芯片直连
	/CE1	芯片片选信号	下拉
	CE2		上拉
	/CE3		下拉
	/BWa, /BWb, /BWc, /BWd	字节写选择输入	上拉（因为使用/GW 信号，/BWa, /BWb, /BWc, /BWd, /BWE 将被忽略）
	/BWE	字节写使能	上拉（理由同上）
	/ADV	有效时，在 burst 访问模式时地址自动增加	上拉或下拉，可根据实际需要 要通过电阻选焊实现
	/ADSC	控制器地址选通	上拉或下拉，本应用中下拉
	/ADSP	处理器地址选通	上拉或下拉，本应用中上拉
	MODE	Burst 序列选择，接地时选择线性 burst 顺序，悬空或上拉时选择交织 burst	上拉或下拉，可根据实际需要 要通过电阻选焊，本应用中 选择下拉

分析 ALM 及所选 SSRAM 的具体接口信号可知，电路连接方面基本上只要将二者数据线、地址线、时钟以及几个主要控制信号直连，其他的信号做一些上下拉处理即可。如下图所示，给出了二者接口的电路连接框图。同时在表 1 中最后一列也给出了在该例中 SSRAM 各具体信号的相应处理方式。



SDRAM 应用

存储器是容量数据处理电路的重要组成部分。随着数据处理技术的进一步发展，对于存储器的容量和性能提出了越来越高的要求。同步动态随机存储器 SDRAM（Synchronous Dynamic Random Access Memory）因其容量大、读写速度快、支持突发式读写及相对低廉的价格而得到了广泛的应用。SDRAM 的控制比较复杂，其接口电路设计是关键。

1. SDRAM 的主要控制信号和基本命令

SDRAM 的主要控制信号为：

- CS：片选使能信号，低电平有效；
- RAS：行地址选通信号，低电平有效；
- CAS：列地址选通信号，低电平有效；
- WE：写使能信号，低电平有效。

SDRAM 的基本命令及主要控制信号见表 1。

表 1 SDRAM 基本操作及控制信号

命 令 名 称	CS	RAS	CAS	WE
命令禁止（NOP:Command inhibit）	H	X	X	X
空操作（NOP： No operation）	L	H	H	H
激活操作（ACT： Select bank and active row）	L	L	H	H
读操作（READ： Select bank and column,and start READ burst）	L	H	L	H
写操作（WRITE： Select bank and column,and start WRITE burst）	L	H	L	L
突发操作停止（BTR： Burst terminate）	L	H	H	L
预充电（PRE： Deactive row in bank or banks）	L	L	H	L
自动刷新或自我刷新（REF： Auto refresh or self refresh）	L	L	L	H
配置模式寄存器（LMR： Load mode register）	L	L	L	L

所有的操作控制信号、输入输出数据都与外部时钟同步。

一个完备的 SDRAM 接口很复杂。对于常规的 SDRAM 应用来说，处理的事件相对来说比较简单，因而可以简化设计而不影响性能。接口电路 SDRAM 的主要操作可以分为：初始化操作、读操作、写操作、自动刷新操作。

（1）初始化操作

SDRAM 上电一段时间后，经过初始化操作才可以进入正常工作过程。初始化主要完成预充电、自动刷新模式寄存器的配置。

（2）激活操作

SDRAM 在进行读写之前，必须将位于某一个 BANK 或所有 BANK 中的行（row）地址进行激活，之后才能进行对相应区域进行读写，激活操作中，地址线上出现的将是行地址和 BANK 选择地址。

（3）读写操作

读写操作就是对 SDRAM 进行数据的存取，在读写操作期间，地址线上出现的将是列地址（COLUMN）和 BANK 选择地址。读写操作可以进行单字节的操作，也可以进行 BURST 操作。

（4）刷新操作

动态存储器（Dynamic RAM）都存在刷新问题。这里主要采用自动刷新方式，每隔一段时间向 SDRAM 发一条刷新命令。

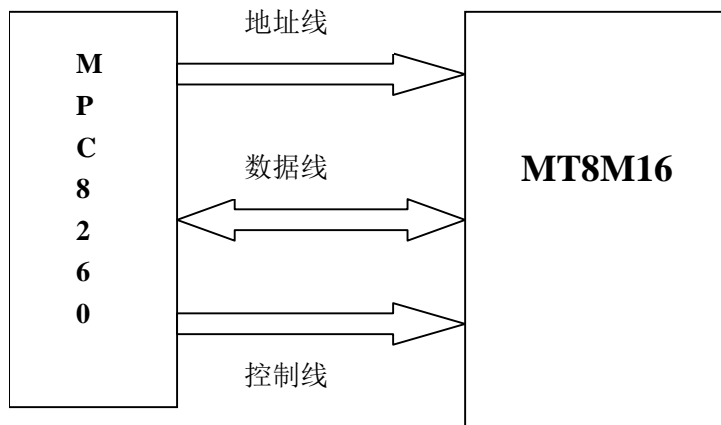
2. SDRAM 应用介绍

目前我们的产品中，在应用到微处理器时，基本上都要使用 SDRAM 作为处理器的主要内存，由于技术的不断发展，现在的 CPU 对 SDRAM 的要求也越来越高，主要体现在容量和速度上的提高，下面以美光公司的 8M16 为例，来介绍一下 SDRAM 在通讯系统中的具体应用。

MT8M16 是美光公司在 sdram 上的一个代表产品，主要特点如下：

- 1) 128MBIT 容量： 2M X 16 X 4 BANKS
- 2) 速度有 10ns 和 7ns 两种，最新的还有 5ns，支持最高的时钟频率可达 200MHZ。
- 3) 物理地址线 A0-A11 总共为 12 根，行列地址线复用，其中，行地址线为 A0-A11，列地址线为 A0-A9，因此总共的逻辑地址共为 22 根，数据线宽为 16bit，总容量为 8MX16bit=128Mbit。
- 4) 内部有 4 个 BANK，通过和控制器相应的输出控制线相连，即可实现整个区间的访问。

下图是用 MOTOROLA 的 MPC8260 和 MT8M16 相连接的示意图：



其中地址线和数据线可以按照常规的 CPU 接外设的方法来实现，要注意的是，对于 SDRAM 来说，行列地址线是复用的，控制线主要包括：

- 1) 片选 CS，选中 SDRAM，实际上通过对 cpu 寄存器的设置就可以通过该片选信号决定了该 SDRAM 在 CPU 的地址空间中的基址
- 2) 时钟信号。
- 3) 读写信号
- 4) 其他：包括数据输出屏蔽，时钟使能等信号，这些都可以由 cpu 相应的管脚来控制。

FIFO

FIFO(first in first out memory)是一种先进先出的存储器，广泛应用于接口电路中的数据缓冲，数据暂存，在现代通讯产品中，许多数据的处理都是要经过许多级处理器的树立阶段，在这些阶段的结合过程中，往往需要进行数据的缓冲，这时，我们可以充分利用 FIFO 进行设计，使得我们的数据处理流程能够更加合理和灵活。

FIFO 一般分为同步 FIFO 和异步 FIFO，同步 FIFO 的读写都是和读写时钟保持同步的，而异步 FIFO 的读写就没有时钟同步的概念了。

一般来说，FIFO 的主要信号有：

- ✓ 读写数据线：有 9 位的，也有 18 位的。
- ✓ 读写使能线：只有有效的时候才能进行读写。
- ✓ FIFO 满标志 (full) 和 FIFO 空标志 (empty)：当满 (full) 标志有效时，表明目前 FIFO 不能在进行写操作，只有经过若干读 FIFO 操作之后，full 标志无效时才可以进行写 FIFO 操作；当空 (empty) 标志有效时，表明目前 FIFO 不能进行读操作，只有经过若干写 FIFO 操作之后，才可以进行读 FIFO 操作。

目前 FIFO 芯片的主要性能主要体现在：数据宽度（9 位和 18 位），存储容量（从几十 K 到几 M 字节），存储速度（15ns，10ns，等），我们在具体的应用中，应该根据自己单板上的实际需求，合理的选择相应的芯片进行设计。