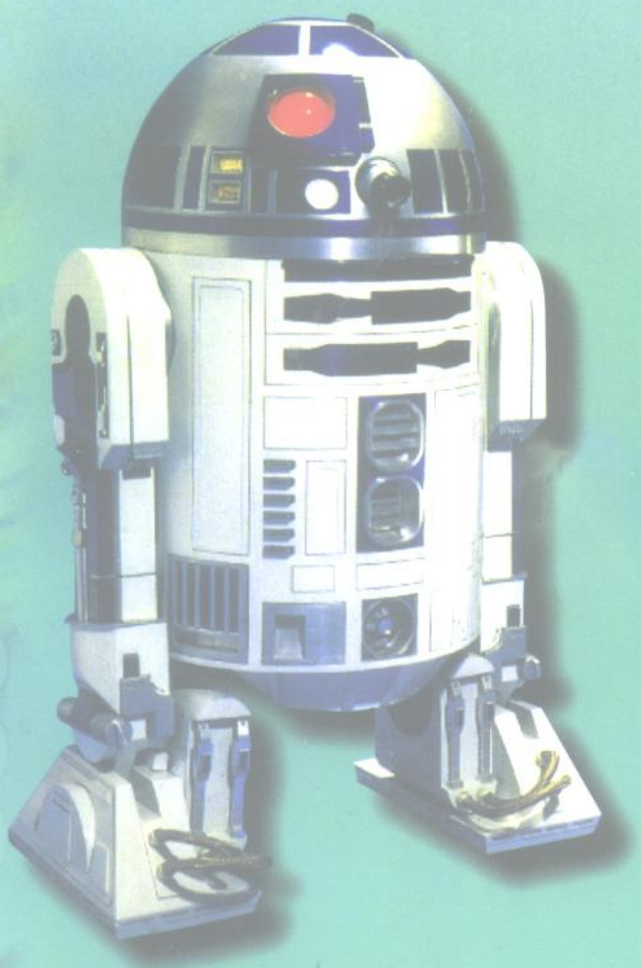
高性能16位微控制器 80C51XA

李 刚 编著



天津大学出版社

高性能16位微控制器80C51XA

高性能16位微控制器80C51XA

李　刚　编著

天津大学出版社

内 容 提 要

微控制器 80C51XA 是由 PHILIPS 公司在 80C51 的基础上发展起来的。它不仅在 80C51 的基础上进一步提高了控制能力，而且在数据处理等方面得到大大地增强。本书系统地介绍了 80C51XA 的硬件结构，存储器的组织结构，系统时钟与定时器，中断系统，外部总线、I/O 口与片内外围设备，通用串行接口与指令系统以及寻址方式与指令系统。书后附录部分有 80C51XA 指令速查表。

本书对从事单片机应用，特别是研制开发高速工业控制系统、高档智能仪器和机器人等对微处理器/微控制器要求较高的工程技术人员肯定会有所帮助；对于愿意学习和从事单片机应用的大学生和研究生，本书则可帮助他们在较高的水平上起步。

高性能 16 位微控制器 80C51XA

李 刚 编著

*

天津大学出版社出版
（天津大学内）
邮编：300072
天津宝坻第二印刷厂印刷
新华书店天津发行所发行

*

开本：787 × 1092 毫米 1/16 印张：11 3/4 字数：294 千
1998 年 4 月第 1 版 1998 年 5 月第 1 次印刷
印数：1—5000
ISBN 7-5618-1066-0

TP·115 定价：13.00 元

前　言

由微控制器 80C51XA 这个名字使人自然地想到单片微机 80C51——在我国应用最广的单片微机。80C51 系列单片微机以其面市早、控制能力强和性能好等特点而最先在中国推广应用。或许有的读者会认为 80C51XA 就是 80C51 系列单片微机的一个新品种。这种想法既对也不对。80C51XA 的确是 80C51 的最正宗的生产厂家——PHILIPS 公司在 80C51 的基础上发展起来的，80C51XA 在硬件和软件上均与 80C51 兼容，但这只是为了 80C51 的用户能很容易地提高已成熟产品的性能和档次，同时也使得 80C51XA 能继承 80C51 控制能力强的优点。重要的是，80C51XA 不仅在 80C51 的基础上进一步地提高了控制能力，而且在数据处理等方面得到大大地增强。其实，80C51XA 名字中的 XA 是 eXtended Architecture 的缩写，就是增强的意思。从以下几个方面可以了解到 XA 的确切含义：

①真正的 16 位静态 CPU。

②采用寄存器－寄存器结构。寄存器组中的任一寄存器都可以像累加器那样完成各种算术和逻辑运算。这就避免 80C51 只能在累加器中进行运算的“瓶颈”限制。寄存器组中的任一寄存器都可以作为数据指针，而不像 80C51 那样只有一个数据指针，这也避免了单一数据指针时片内外数据交换的“瓶颈”限制。80C51XA 的寻址方式不仅比 80C51 多，也比 80C51 强。由此可见，80C51XA 在数据处理和传输方面得到大大增强。

③80C51XA 的地址线高达 24 位。由于 80C51XA 也是采用普林斯顿结构，因而它的程序和数据的空间都可达 16 M 字(节)，这足以满足绝大多数应用场合的要求。

④80C51XA 具有硬件支持实时多任务的特点。对于高速工业控制和测量系统，具有实时多任务系统是不可缺少的。80C51XA 在这方面比 80C51 有了大幅度的增强。

⑤80C51XA 具有丰富的中断源和大大增强了的中断机制。80C51XA 有四大类中断源，即事件中断、例外中断、软件中断和陷阱中断，总共有 32 个中断矢量。

⑥80C51XA 的时钟频率可达 40 MHz，还有“猝发”方式取指，等等。其执行指令的速度要比 80C51 快几倍。考虑到 80C51XA 的字长和寻址能力，80C51XA 的实际数据处理能力要比 80C51 高出上百倍。

⑦虽然 80C51XA 串行口的基本原理和工作模式与 80C51 基本相同，但 80C51XA 有一套硬件支持的地址自动识别机制，再加上 80C51XA 串行口的发送和接收寄存器都是双缓冲结构的，因而 80C51XA 在多机通信方面也比 80C51XA 强了很多。

⑧80C51XA 的 I/O 口都是可编程的。80C51XA 的 I/O 口既可对其输出方式进行编程，也可对其时序特性进行编程。这就使得 80C51XA 很容易与各种速度的外围接口芯片和设备接口。80C51XA 的外部数据总线也可以编程为 8 位宽度或 16 位宽度。不难看出，80C51XA 的接口能力也比 80C51 增强很多很多。

……

如果要继续列举下去，80C51XA 还有很多值得一叙的特点，如 CPU 内核加片内外围设备的结构、颇具特色的“看门狗”定时器和“喂狗”指令、两种降低功耗的方式，等等。正因为

80C51XA有这么多的特点和功能，使得本书作者兴奋不已，急急忙忙将它介绍给大家，希望能与大家分享。急忙之中就容易出差错，加上作者的水平有限，本书中的错误在所难免。作者诚恳地欢迎各位的批评指正。但不管怎样，作者坚信本书对从事单片机应用，特别是研制开发高档智能仪器和机器人等对微处理器/微控制器要求较高的工程技术人员会有所帮助。此外，对于愿意学习和从事单片微机应用的大学生和研究生，本书则可以帮助他们在较高的水平上起步。

李　刚

1997年10月于天大校园

目　录

第一章　80C51XA 的硬件结构 …… (1)
1.1　概述 …… (1)
1.2　80C51XA 引脚的详细说明 …… (3)
1.3　CPU 组织 …… (5)
1.3.1　程序状态字 …… (5)
1.3.2　系统特征寄存器 …… (8)
1.3.3　复位 …… (9)
第二章　存储器的组织结构 …… (12)
2.1　寄存器文件 …… (12)
2.2　数据寄存器 …… (14)
2.3　程序存储器 …… (16)
2.4　专用寄存器 …… (17)
第三章　系统时钟与定时器 …… (19)
3.1　振荡器 …… (19)
3.2　功耗控制 …… (19)
3.3　定时/计数器 …… (20)
3.4　看门狗定时器 …… (24)
第四章　中断系统 …… (27)
4.1　堆栈 …… (27)
4.2　中断 …… (30)
4.3　堆栈帧 …… (37)
4.4　中断矢量表 …… (37)
4.5　跟踪模式调试 …… (39)
第五章　外部总线、I/O 口与片内外围设备 …… (41)
5.1　外部总线信号 …… (41)
5.2　外部总线的设置 …… (42)
5.3　总线定时与时序 …… (44)
5.3.1　访问外部程序存储器的定时和时序 …… (44)
5.3.2　访问外部数据存储器的定时和时序 …… (45)
5.3.3　外部总线信号定时的设置 …… (47)
5.4　复位配置 …… (49)
5.5　I/O 口 …… (50)
5.5.1　I/O 口操作 …… (50)
5.5.2　端口输出的配置 …… (50)

5.5.3 端口的复位状态和初始化 …… (52)
5.5.4 I/O端口与外围器件的共享 …… (52)
5.6 外围总线 …… (53)
5.6.1 已实现和可能的增强功能 …… (53)
5.6.2 读-改-写锁定 …… (53)
5.6.3 XA片内的外围设备 …… (53)
第六章 通用串行接口 …… (57)
6.1 通用串行接口的工作模式 …… (57)
6.2 串行口控制和状态寄存器 …… (58)
6.3 串行口中断 …… (59)
6.4 多机通信 …… (60)
第七章 寻址方式与指令系统 …… (63)
7.1 寻址方式 …… (63)
7.2 寻址方式的说明 …… (63)
7.3 相对分支和跳转 …… (67)
7.4 数据类型 …… (68)
7.5 指令详解 …… (68)
附录 …… (171)

第一章　80C51XA 的硬件结构

1.1　概　　述

80C51XA 有一种通用寄存器－寄存器结构，这种结构使高速的微控制器具有较好的性能价格比。兼顾与 80C51 的兼容性，使原来使用 80C51 的系统可方便地升级而得到更好的性能或更大的存储量。作为一种功能强大的通用 16 位微控制器，80C51XA 增加了对多任务操作系统和高级语言（如 C）的支持，同时还保留了 80C51 中很有特色的位操作功能。

目前，PHILIPS 公司提供的 80C51XA 系列 16 位 CMOS 单片微控制器主要有三种系列，即 XA－G1、XA－G2 和 XA－G3。这三种单片微控制器都具有相同的内核。除了片内配置的程序存储器大小不同（XA－G1 为 8kbyte，XA－G2 为 16kbyte 和 XA－G3 为 32kbyte）外，它们的内部结构和引脚配置都基本上相同。所以本书只介绍 XA－G1 的硬件结构，但对这些结构的介绍同样适用于 XA－G2 和 XA－G3，实际上对未来 XA 系列其他型号的产品也同样适用。

图 1－1 是 80C51XA CPU 内核的结构图。图 1－2 是 80C51XA 16 位 CMOS 单片微控制器的结构框图。图 1－3 是 80C51XA 的逻辑符号图。图 1－4(a)、(b)是 80C51XA G1 的引脚配置图。表 1－1 给出了两种封装形式的引脚功能。

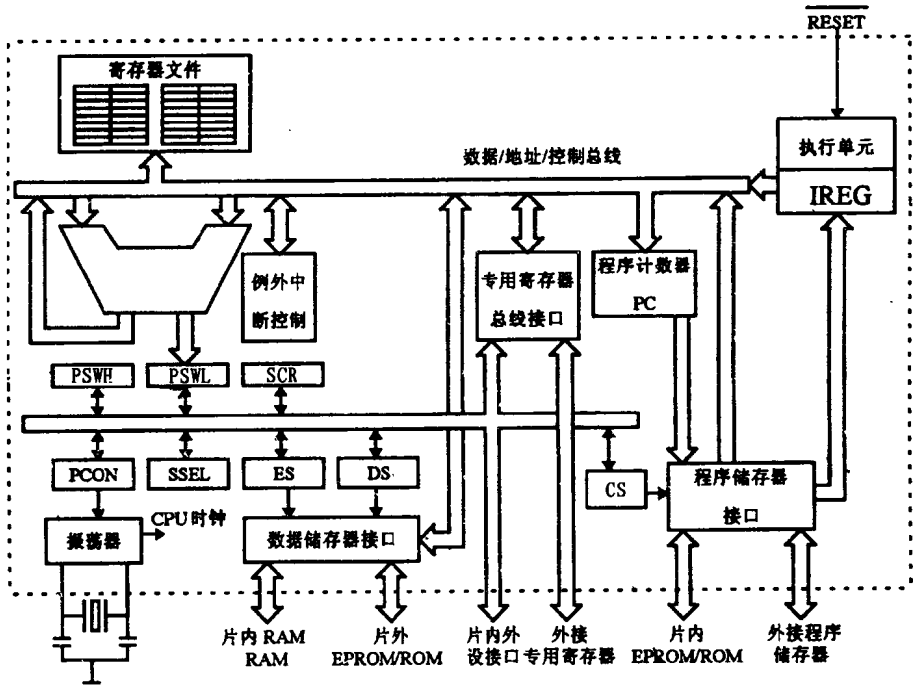


图 1－1　80C51XA CPU 内核的结构

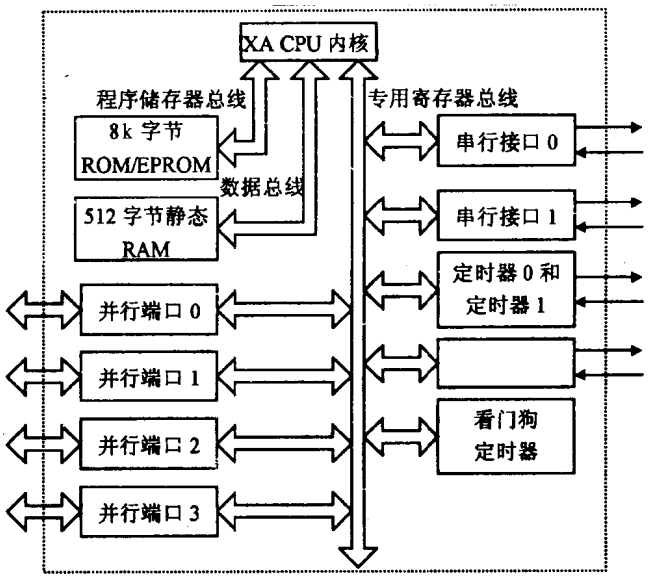


图 1－2　80C51XA 16 位 CMOS 单片微控制器的结构框图

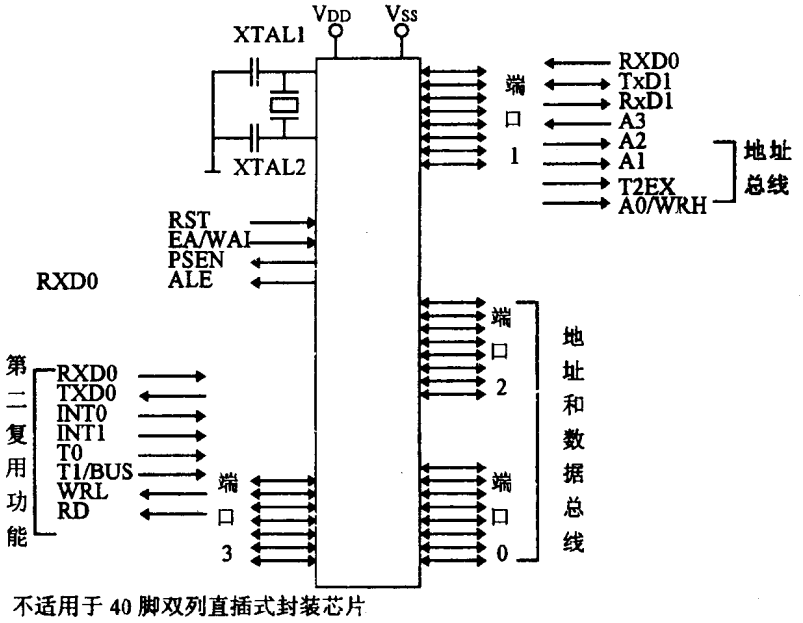
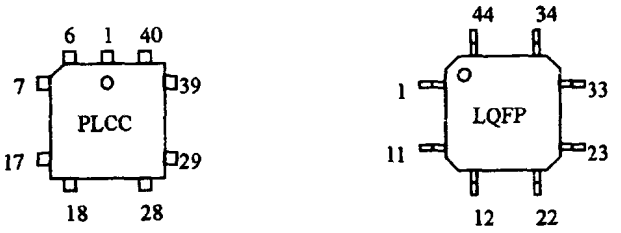


图 1－3　80C51XA 的逻辑符号图



(a) (b)

图 1－4 80C51XA G1 的引脚配置图

表 1－1 80C51XA G1 的引脚

引脚	功 能	引脚	功 能	引脚	功 能	引脚	功 能
1	V_{SS}	23	V_{DD}	1	$P_{1.5}/TxD_1$	23	$P_{2.5}/A_{17}D_{13}$
2	$P_{1.0}/A_0/\overline{WRH}$	24	$P_{2.0}/A_{12}D_8$	2	$P_{1.6}/T2$	24	$P_{2.6}/A_{18}D_{14}$
3	$P_{1.1}/A_1$	25	$P_{2.1}/A_{13}D_9$	3	$P_{1.7}/T2EX$	25	$P_{2.7}/A_{19}D_{15}$
4	$P_{1.2}/A_2$	26	$P_{2.2}/A_{14}D_{10}$	4	$\overline{RST}$	26	$\overline{PSEN}$
5	$P_{1.3}/A_3$	27	$P_{2.3}/A_{15}D_{11}$	5	$P_{3.0}/RxD0$	27	$ALE/\overline{PROG}$
6	$P_{1.4}/RxD1$	28	$P_{2.4}/A_{16}D_{12}$	6	NC	28	NC
7	$P_{1.5}/TxD1$	29	$P_{2.5}/A_{17}D_{13}$	7	$P_{3.1}/TxD0$	29	$\overline{EA}/VPP/WAIT$
8	$P_{1.6}/T2$	30	$P_{2.6}/A_{18}D_{14}$	8	$P_{3.2}/\overline{INT0}$	30	$P_{0.7}/A_{11}D_7$
9	$P_{1.7}/T2EX$	31	$P_{2.7}/A_{19}D_{15}$	9	$P_{3.3}/\overline{INT1}$	31	$P_{0.6}/A_{10}D_6$
10	$\overline{TST}$	32	$\overline{PSEN}$	10	$P_{3.4}/T0$	32	$P_{0.5}/A_9D_5$
11	$P_{3.0}/RxD0$	33	$ALE/\overline{PROG}$	11	$P_{3.5}/T1/BUSW$	33	$P_{0.4}/A_8D_4$
12	NC	34	NC	12	$P_{3.6}/\overline{WRL}$	34	$P_{0.3}/A_7D_3$
13	$P_{3.1}/TxD0$	35	$\overline{EA}/VPP/WAIT$	13	$P_{3.7}/\overline{RD}$	35	$P_{0.2}/A_6D_2$
14	$P_{3.2}/\overline{INT0}$	36	$P_{0.7}/A_{11}D_7$	14	XTAL2	36	$P_{0.1}/A_5D_1$
15	$P_{3.3}/\overline{INT1}$	37	$P_{0.6}/A_{10}D_6$	15	XTAL1	37	$P_{0.0}/A_4D_0$
16	$P_{3.4}/T0$	38	$P_{0.5}/A_9D_5$	16	V_{SS}	38	V_{DD}
17	$P_{3.5}/T1/BUSW$	39	$P_{0.4}/A_8D_4$	17	V_{DD}	39	V_{SS}
18	$P_{3.6}/\overline{WRL}$	40	$P_{0.3}/A_7D_3$	18	$P_{2.0}/A_{12}D_8$	40	$P_{1.0}/A_0/\overline{WRH}$
19	$P_{3.7}/\overline{RD}$	41	$P_{0.2}/A_6D_2$	19	$P_{2.1}/A_{13}D_9$	41	$P_{1.1}/A_1$
20	XTAL2	42	$P_{0.1}/A_5D_1$	20	$P_{2.2}/A_{14}D_{10}$	42	$P_{1.2}/A_2$
21	XTAL1	43	$P_{0.0}/A_4D_0$	21	$P_{2.3}/A_{15}D_{11}$	43	$P_{1.3}/A_3$
22	V_{SS}	44	V_{DD}	22	$P_{2.4}/A_{16}D_{12}$	44	$P_{1.4}/RxD1$

1.2 80C51XA 引脚的详细说明

表 1－2 列出 80C51XA G1 的引脚详细说明。

表 1－2　80C51XA G1 的引脚详细说明

符号	引脚序号		类型	名　称　和　功　能
	LCC	LQFP		
V_{SS}	1、22	16	I	GND，地，0V 参考点
V_{DD}	23、44	17	I	主电源输入端，提供正常工作、待机和掉电三种工作模式时的电源
$P_{0.0}$～$P_{0.7}$	43－36	37－30	I/O	P_0 口是一个 8 位 I/O 口。它可以由用户设置输出类型。复位时会自动对 P_0 口锁存器写入 1，使得 P_0 口设置为准双向工作模式。P_0 口也可以设置为输入或输出方式，并且每一根口线都可独立地设置。详细可参见有关 I/O 口设置和口线的直流特性部分的内容。作为外部程序/数据总线时，P_0 口分时多路转换数据/指令的低位字节；作为第 4～11 位地址线在对片内 EPROM 编程时，程序代码由 P_0 口写入；校验时则由 P_0 口读出
$P_{1.0}$～$P_{1.7}$	2－9 1－3	40－44	I/O	P_1 口是一个用户可编程的 8 位 I/O 口，在复位时设置为准双向工作模式。P_0 口也可以设置为输入或输出方式，并且每一根口线都可独立地设置。详见 I/O 口设置和口线的直流特性部分的内容
				P_1 口还有校验专用功能如下：
	2	40	O	A_0/$\overline{WRH}$是在外部总线设置为 8 位宽度时，它是第 0 位地址线，在外部数据总线设置为 16 位宽度时，它是高位字节数据写入控制信号线
	3	41	O	A_1 是外部地址总线第 1 位；
	4	42	O	A_2 是外部地址总线第 2 位；
	5	43	O	A_3 是外部地址总线第 3 位；
	6	44	I	RxD1($P_{1.4}$)是串行接口 1 的接收端；
	7	1	O	TxD1($P_{1.5}$)是串行接口 1 的发送端；
	8	2	I	T_2($P_{1.6}$)是定时/计数器 2 的计数输入/时钟输出端；
	9	3	I	T_2EX($P_{1.7}$)是定时/计数器 2 的重装/捕捉/直接控制输入线
$P_{2.0}$～$P_{2.7}$	24－31	18－25	I/O	P_2 口是一个用户可编程的 8 位 I/O 口，在复位时，设置为准双向工作模式。P_2 口也可以设置为输入或输出方式，并且每一根口线都可独立地设置。详细内容参见有关 I/O 口设置和口线的直流特性部分的内容。当外部数据总线设定为 16 位宽度时，P_2 口分时多路转换为高位字节的数据/指令线和地址线第 12 至 19 位。当外部数据总线设定为 8 位宽度时，P_2 口中作为地址线的根数由用户设定。在对片内程序存储器进行编程校验时，P_2 口接收低位字节的地址
$P_{3.0}$～$P_{3.7}$	11， 13－19	5， 7－13	I/O	P_3 口是一个用户可编程的 8 位 I/O 口，在复位时设置为准双向工作模式。P_3 口也可以设置为输入或输出方式，并且每一根口线都可以独立地设置。欲详细了解可参见有关 I/O 口设置和口线的直流特性部分的内容。在对片内程序存储器进行编程校验时，P_3 口接收高位字节的地址。P_3 口还具有专用功能如下：
	11	5	I	RxD0($P_{3.0}$)是串行口 0 的输入端；
	13	7	O	TxD0($P_{3.1}$)是串行口 0 的输出端；
	14	8	I	$\overline{INT0}$($P_{3.2}$)是外部中断 0 的输入端；
	15	9	I	$\overline{INT1}$($P_{3.3}$)是外部中断 0 的输入端；
	16	10	I/O	T_0($P_{3.4}$)是定时器 0 的外部输入端，或定时器 0 的溢出输出端；
	17	11	I/O	T_1/BUSW($P_{3.5}$)是定时器 1 的外部输入端，或定时器 1 的溢出输出端，在复位时端口锁存器将锁存该端口的输入状态，以确定外部数据总线的宽度(BUSW)；
	18	12	O	$\overline{WRL}$($P_{3.6}$)是外部数据存储器低位字节写控制信号输出端；
				$\overline{RD}$($P_{3.7}$)是外部数据存储器读控制信号输出端

续表

符号	引脚序号		类型	名称和功能
	LCC	LQFP		
$\overline{RST}$	10	4	I	复位端,低电平有效。在该端出现低电平时,将复位整个微控制器,使各个I/O端口和片上外设复位到缺省状态,CPU执行由复位向量指向的程序存储器地址起始的程序。参见有关复位部分的内容
ALE /$\overline{PROG}$	33	27	I/O	地址锁存使能端/编程脉冲输入端。在总线分时多路转换输出地址信号时,该端输出一个高电平到外部地址锁存电路的ALE端以锁存地址信号。这个ALE仅在除了一个总线周期需要时出现。在对片内EPROM编程时,这个端口作为编程脉冲输入端
$\overline{PSEN}$	32	26	O	程序存储器读出使能端。在微控制器访问外部程序存储器时,$\overline{PSEN}$输出一个低电平使外部程序存储器输出有效。这个信号仅在取指时有效
$\overline{EA}$ /WAIT /V_{PP}	35	29	I	外部取指/等待/编程电源端,对$\overline{EA}$端所施加的电平决定了微控制器是从内部还是外部程序存储器读取指令。在复位时,端口锁存器将锁存$\overline{EA}$引脚的输入信号。当锁存的是低电平(0)时,微控制器将从外部程序存储器读取指令。当锁存的是高电平(1)时,微控制器将从内部程序存储器读取指令。如果微控制器的程序计数器超出了内部程序存储器的范围,微控制器将自动地转向外部程序存储器读取指令。在复位结束后,该端口在正常工作期间是作为等待信号输入端,在访问外部存储器时,对WAIT端施加高电平将使CPU一直等待到WAIT端变为低电平为止。WAIT端口使得微控制器与一些低速器件的接口变得容易。在对片内EPROM编程时,该端口作为编程电压输入端
XTAL1	21	15	I	系统时钟振荡器中反相放大器的输入端。在使用内部时钟时,外接晶体或陶瓷片以构成片内振荡器。在使用外部时钟时,该端口作为输入端输入外部时钟信号
XTAL2	20	14	O	系统时钟振荡器中反相放大器的输出端。在使用内部时钟时,外接晶体或陶瓷片构成片内振荡器。在使用外部时钟时,该端口开路悬空

1.3 CPU组织

80C51XA内核中的振荡器提供基本的系统时钟。定时和控制逻辑由外部复位信号初始化。一旦初始化后这个逻辑就为内部和外部的程序和数据存储器访问提供定时。这个逻辑还管理程序计数器的加载和通过程序存储器接口把指令取到指令寄存器中。定时和控制逻辑还经过数据存储器接口管理数据的传送,还管理算术逻辑单元(ALU)完成算术和逻辑运算。ALU把状态信息储存到程序状态字(PSW)的低位字节(PSWL)中。在线寄存器文件保存当前堆栈(SP)中的值。程序状态字(PSW)的高位字节(PSWH)用于:①选择优先级较高的系统模式还是较有限制的用户模式;②选择用于单步调试的跟踪模式;③选择当前寄存器组;④记录当前所执行程序的优先级。设定系统特征寄存器(SCR)可以选择80C51XA本身的工作模式或与80C51兼容的模式。页面选择寄存器(SSL)则可选择代码页面寄存器(CS)、数据页面寄存器(DS)和附加页面寄存器(ES)。80C51XA的结构支持片内和片外的RAM、ROM/EPROM和SFR的接口。

1.3.1 程序状态字

程序状态字(PSW)是管理CPU运行模式和记录CPU工作状态的双字节专用寄存器。PSW

的低位字节(PSWL)保存 CPU 的状态标志,反映了每条指令的执行结果。不论是工作于用户模式还是系统模式,PSWL 都是可读写的。图 1-5 为 80C51XA 的程序状态字 PSW。

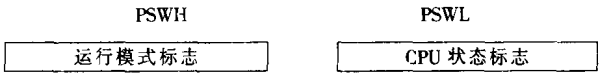


图 1-5 80C51XA 的程序状态字 PSW

PSW 的高位字节(PSWH)由程序设置,以确定 80C51XA 的工作模式和参数:系统/用户工作模式、跟踪模式、寄存器组的选择和任务的优先级等。PSWH 是可读的,但只有运行在系统模式时才可以写。图 1-6 为 CPU 状态标志寄存器 PSWL。

	207	206	205	204	203	202	201	200
420	C	AC	-	-	-	V	N	Z

图 1-6 CPU 状态标志寄存器 PSWL

PSWL 又叫作 CPU 状态标志。它包括进位位(C)标志、辅助进位位(AC)标志、溢出位(V)标志、负标志(N)和零标志(Z)。有些指令的执行会影响全部标志,有些只影响部分,还有一些指令不影响任何标志。一般来说,程序可读取这些标志以决定流向。在第六章将会详细地给出每条指令对这些标志的影响。

下面详细介绍 PSWL 中的每一标志位。

图 1-6 左端的"420"是 PSWL 在专用寄存器空间的地址,上方"207、206、…"为对应标志位的位地址。在后面涉及专用寄存器的介绍时,本书均以这种方式示意,而不再重复说明。

(1)C 进位位。一般来说,C 反映了算术和逻辑运算的结果。对算术运算指令,如 ADD、ADDC、CMP、CJNE、DA、SUB 和 SUBB 等,如果有进位,则 C 被置 1。对于一些逻辑运算指令,如 ASR、LSL、LSR、RLC 和 RRC 等,也可能间接地使得 C 被置 1。而乘法和除法指令,如 MUL8、MULU8、MULU16、DIV16、DIV32、DIVU8、DIVU16 和 DIVU32 等,则无条件地使 C 清 0。

(2)AC 辅助进位位。在执行 ADD、ADDC、CMP、SUB、SUBB 指令过程中,如果低半字节有向高半字节进位时则置 AC 为 1,否则清 0。这个标志用于十进制调整指令(DA)进行 BCD 码算术运算用。

(3)V 溢出标志。在执行 ADD、ADDC、CMP、NEG、SUB 和 SUBB 等指令对两个补码进行运算时若产生溢出,则置 V 为 1,否则清 V 为 0。所谓溢出,指出现如下的情况:

当执行加法指令时,若以 C_i 表示位 i 向位 i+1 有进位,则对两个字节的补码进行运算中,有

$$V = C_6 \oplus C_7$$

即当位 6 向位 7 有进位而位 7 不向 C 进位时,或者位 6 不向位 7 进位而位 7 向 C 进位时,溢出标志 V 置位,否则清 0。

在对两个字的补码进行运算时,类似地有

$$V = C_{14} \oplus C_{15}$$

在作减法时,如果以 C_i 表示位 i 向位 i+1 有借位,则在对两个字节的补码进行运算时有

$$V = C_6 \oplus C_7$$

而对两个字节的补码进行运算时有

$$V = C_{14} \oplus C_{15}$$

在执行除法指令 DIV16、DIV32、DIV8、DIVU16 和 DIVU32 时，如果运算的结果超出目的寄存器表存储的数字范围，或者被 0 除，都使标志 V 置 1，反之清 0。

而在执行除法指令 MUL16、MULU8 和 MULU16 时，如果运算结果超出源操作数的表示范围，也将使标志 V 置 1，反之清 0。此时的“溢出”仅仅是告诉程序，指令的运算结果比源操作数的类型要大，例如两个整数相乘而得到一个长整数就会使 V 置 1。

(4)N　负标志位。该标志反映了两个补码的运算结果，即符号位的内容。数据传送指令也会影响 N 标志。但 N 标志不受指令 PUSH 和 POP 的影响。

(5)Z　“0”标志。这个标志反映了算术运算和数据传输的结果。如果这两类操作的结果或数值为 0，则置 Z 标志为 1，反之清 0。该标志不受 PUSH、POP 和 XCH 等指令的影响。

在 PSWL 中标有“-”的位是为将来使用而保留的。最好不要对这些位进行操作以免造成不可预测的影响。这些位都被设置为 0，以免使将来新的 80C51XA 的型号受到影响。

PSWH 又称为运算方式标志寄存器。通过对 PSWH 的设置可以选择 80C51XA 的运算状态。除 RS1 和 RS0 外，其他标志只能在运行系统模式时改变。图 1-7 为运行方式标志寄存器 PSWH。PSWH 中各标志位的含义如下。

	20F	20E	20D	20C	20B	20A	209	208
401	SM	TM	RS1	RS0	IM3	IM2	IM1	IM0

图 1-7　运行方式标志寄存器 PSWH

(1)SM　系统模式位。当该位置 1 时，80C51XA 工作在系统模式。此时，程序可以访问所有的寄存器和存储器，运行任何一条指令。而当 SM 清 0 时，80C51XA 运行在用户模式，此时有许多操作受到限制，如对 PSWH 本身的许多位就不能进行操作。

(2)TM　跟踪方式位。当 TM 置位时，80C51XA 本身可提供排错调试的功能。而当 TM 清 0 时，就不能用这些功能。

(3)RS1 和 RS0　当前工作寄存器组设置位。选择当前工作寄存器组可以通过对这两位的设置而得到。被选择为当前工作寄存器组中的寄存器不论工作于何种模式都可被访问，反之，没有被选择作为当前工作寄存器组的其他 3 组寄存器，则无论如何都不能被访问。

(4)IM3 至 IM0　中断屏蔽位，用来标明目前正在执行的程序优先级的大小。事件中断控制系统自动比较正在执行程序的优先级和正在申请中断的优先级的大小，以决定是否响应中断。IM 的值为 0 时，表示正在执行程序的优先级最低，可以响应任何一个事件中断服务程序。而当 IM 值为 15 时，表示当前正在执行程序的优先级最高，不允许响应任何事件中断程序，但不能禁止响应例外中断和非屏蔽中断。

IM 的值只能在系统模式中写入，而用中断处理指令则可以读出，以决定软件中断的优先级。直接对 IM 写入一个新的值可导致优先级反转，即使当前正在执行程序的优先级低于以前申请中断的程序的优先级。某些 80C51XA 品种的软件中断机制可以避免优先级反转。详细情况可参见有关软件中断的内容。

PSW 中的两个字节 PSWH 和 PSWL 是作为两个专用寄存器来寻址的。当对 PSW 用一条指令进行操作时，可能会引起一些潜在的模糊性，这只能用一些显式指令来避免。以下例子可以说明这种情况：执行“MOV R0，#0081H”时，由于传送的是一个负的二进制补码数，所以 N 标志被置为 1。而执行“MOV PSWL，#81H”后会将 C 和 Z 标志置 1，但 N 标志被清 0，这实际上是显式地对 PSW 进行操作。如果执行“OR PSWH，#30h”，则 PSWL 的内容不受影响。

在 80C51XA 复位时，PSW 将自动装载位于地址为 0 的程序存储器中的复位矢量。PHILIPS 推荐对 PSW 中的 IM3 至 IM0 初始化为 1，这样 80C51XA 的初始化程序的优先级别最高。除了例外中断和不可屏蔽中断外，初始化程序不受其他中断程序的影响，以完成系统的初始化。在完成初始化以后，一般应将程序的优先级降为 0，这样可以让别的任务得到执行。PHILIPS 还推荐复位矢量设置为 1，以便程序开始运行时处于系统模式。

1.3.2 系统特征寄存器

由系统特征寄存器(SCR)可以设置 80C51XA 的全局运行模式。一般是系统开始运行时便将模式设置好，以后不再改变。图 1-8 为系统特征寄存器(SCR)，其中有 4 位是有定义的，含义如下。

图 1-8 系统特征寄存器(SCR)

(1)PZ 储存器模式设置位。如果 PZ = 0，80C51XA 工作在储存器大模式，使用全部的 24 位地址；而 PZ = 1 时，则工作在存储器小模式，仅使用 16 位地址，即在页面 0。在存储器小模式，由于堆栈和中断等占据一定的空间，程序和数据存储器实际可用的空间都小于 64kbyte。

(2)CM CPU 工作模式设置位。用于选择 80C51XA 固有的工作模式或与 80C51 兼容的模式。当 CM = 0 时，系统工作于 80C51XA 模式；当 CM = 1 时，系统工作与 80C51 兼容的模式。

(3)PT1 和 PT0 时钟分频数设置位。通过这两位的设置可以选择用振荡器时钟作为接口定时时钟源时的分频数，见表 1-3。

表 1-3 时钟分频数的设定

PT1	PT0	接口时钟
0	0	振荡器时钟/4
0	1	振荡器时钟/16
1	0	振荡器时钟/64
1	1	保 留

在 SCR 中有 4 位标有“-”的位是系统为将来的新型号而保留的，对这些位的操作要多加小心。下面就存储器的大模式和小模式做进一步说明。

当 CM = 0 和 PZ = 1 时，80C51XA 工作于存储器大模式，内核用 24 位地址管理 16Mbyte 的全空间，在某些特殊的型号里，也可能用少于 24 位的地址(也即管理较小的存储器空间)。不管怎样，只要工作在大模式，在执行调用指令和响应中断时，都将会把 24 位地址压入堆栈，在

返回时也同样弹出 24 位的返回地址。

在 80C51XA 工作于小模式时，内核用 16 位地址管理 64kbyte 的地址空间。这时，原来用作高 8 位的地址线可以作为其他用途。在执行调用指令和响应中断时，只将 16 位的地址压入堆栈，而在返回时也同样弹出 16 位的返回地址。在使用小模式时，可以节省一些堆栈空间和加速堆栈操作指令 PUSH 和 POP 等的操作。

不推荐在初始化后在大模式和小模式之间切换。原因是切换进入小模式的指令只能在页面 0 的程序存储器中。一旦工作在小模式中，指令代码的地址就成为 16 位，但由于 80C51XA 预取的指令在进入小模式以前已进入指令寄存器，执行该指令很容易引起混乱。再者，在模式切换前已被压入堆栈的地址在切换后就变为非法了。因而，在非要切换存储器模式时，绝对不允许在中断服务程序和子程序中进行。

1.3.3 复　位

所谓复位是在电源刚开启时，由硬件输入一个信号使得 80C51XA 完成一系列初始化操作。但也可以是由指令 RESET 或看门狗溢出使 80C51XA 完成一系列初始化操作。当 80C51XA 第一次上电时，在程序执行前由外部硬件触发的一系列复位动作。主要复位动作如下：

①外部硬件产生复位信号；

②内部产生一系列硬件复位动作；

③$\overline{\text{RST}}$引脚由低电平变高；

④确定外部总线宽度和存储器工作模式；

⑤产生复位例外中断；

⑥开始运行程序。

如果在上电时未能完成正常的复位操作，80C51XA 就可能完全不能工作。要使 80C51XA 正常复位，应该使其$\overline{\text{RST}}$引脚的电平在 V_{DD}上电正常后，至少保持 10μs 时间的低电平。这个时间足以让振荡器稳定地工作并让 CPU 检测复位的条件。同时，CPU 还开始一系列内部复位过程，$\overline{\text{RST}}$的引脚要保持在低电平有足够长的时间使 CPU 能完成这一系列内部复位的动作。CPU 完成这些动作需要 10μs 的时间。CPU 需要完成的这些操作是：

①多数的内核专用寄存器和接口专用寄存器清零，部分接口专用寄存器写入适宜的值；

②使 CS、DS 和 ES 清 0；

③使 SSEL 为 0，即所有的对外部存储器的寻址都经过 DS 进行；

④对寄存器文件中所有的寄存器都清 0；

⑤设置用户和系统堆栈指针(USP 和 SSP)为 0100H；

⑥清除 SCR 中的 PZ 位，即选择存储器大模式为缺省模式；

⑦清除 SCR 中的 CM 位，即工作在 80C51XA 的固有模式；

⑧清除 IE 中的 EA，即禁止所有的可屏蔽中断。

注意到内部复位过程并不初始化片内和片外的 RAM。但对 80C51XA 内核外的部件的影响随型号不同而不同，只能参考有关数据手册。但一般说来，复位会将所有的口线设置为输入方式。复位不影响的部件还有串行口缓冲寄存器、PCA 捕捉寄存器和看门狗加载寄存器等。在完成内部复位动作之后，80C51XA 处于静止状态，直到$\overline{\text{RST}}$变为高电平的瞬间，$\overline{\text{EA}}$和 BUSW

两引脚的电平被 CPU 采样以决定是使用片内程序存储器还是片外程序存储器和确定总线的宽度。如果$\overline{EA}$=0,则 CPU 将从片内程序存储器中取指,但 80C51XA 并没有智能去判断$\overline{EA}$引脚的信号与存储器硬件的关系。如果使$\overline{EA}$=1,但 80C51XA 无片内 EPROM/ROM,80C51XA 并不会自动转向在片外程序存储器中取指,其结果就是导致系统不工作。

如果 BUSW 为低电平,80C51XA 的外部总线接口将工作在 8 位宽度;如果 BUSW 为高电平,80C51XA 的外部总线接口将工作在 16 位宽度。此外,总线宽度也可以用软件设定总线特征寄存器(BCR)来改变总线的宽度。

在$\overline{RST}$释放后,80C51XA 某些型号的 BUSW 还有其他功能,视不同的型号以及封装而定,必要时可参考有关手册。

在$\overline{RST}$变为高电平之后,CPU 还会产生复位例外中断、初始化 PSW 和从位于程序存储器 0 地址处取回第一条指令,这条指令又叫"启动码"。下面是设置中断例外的例子:

```
code - seg                    ;确定程序页面
org 0h                        ;从 0 地址开始
reset - vector
dw initial - psw              ;定义一个字常数
dw startup - code             ;定义一个字常数
org 120h                      ;跳至矢量表以外
startup - code:
…                             ;在这儿开始放启动码
```

PSWL 的初值在复位时并不重要,但 PSWH 的初始化却需要重视。因为 PSWH 要影响许多系统的特征。下面是一个推荐的初始化 PSWH 的例子:

```
system __mode equ 8000h
max __priority equ 0F00h
initail __PSW equ system __mode + max __priority
```

通过把 PSW 中的 SW 置位,使 80C51XA 初始化在系统模式,这样 CPU 就能毫无限制地读取启动码。

把 RS1 和 RS0 都清 0,这样就选择了存储器组 0 为当前工作寄存器组。当然,选别的寄存器组为当前工作寄存器组也可以,这个选择并无什么特殊意义。

PHILIPS 公司推荐把启动码执行的优先级别取到最高,也就是使从 IM0 到 IM3 位都置 1,这样就可以最优先地执行启动码的指令。由于复位时硬件已将所有的中断关闭,只有非屏蔽中断,才有可能中断启动码的执行。一个较好的办法是在 80C51XA 完成启动过程前用硬件封锁非屏蔽中断。

PHILIPS 公司推荐启动码的第一条指令用于设置系统特征寄存器。由于堆栈指针的缺省值一般都不能正好符合实际需要,所以接着就应显式地设置堆栈指针。

起始码可以有散转和跳转指令。但在结束复位例外中断处理时,不能用 RETI 指令。这是由于 SP 可能被初始化,而中断时也没有返回地址。

下面简要地介绍复位过程与一些重要的子系统的相互作用。