

C8051F326/7
全速 USB、16KB FLASH 微控制器
数 据 手 册

潘 琢 金 译

Rev 0.5 2006.01

版权所有

版 权 声 明

本手册中文版版权归译者和新华龙电子有限公司所有。研究和开发人员可以自由使用本手册。任何单位和个人未经版权所有者授权不得在任何形式的出版物中摘抄本手册内容。

原文中比较明显的错误已经在译文中更正。译者将在本手册英文版更新后及时更新中文版内容。译文中一定存在不少错误和不准确之处，望业界同仁不吝赐教，以便在新版本中更正。

译者联系方式：

沈阳航空工业学院 计算机学院 潘琢金

电话：024-86141552，13066535936

Email: panzhuojin@sina.com 或 panzhj@syiae.edu.cn

C8051F326/7 全速 USB、16KB FLASH 微控制器

USB 控制器

- 符合 USB 规范 2.0 版
- 全速(12Mbps)或低速(1.5Mbps)操作
- 集成时钟恢复电路；无需外部晶体
- 支持 3 个固定功能端点
- 256 字节 USB 缓存
- 集成收发器；无需外部电阻

在片调试

- 片内调试电路提供全速、非侵入式的在系统调试（不需仿真器）
- 支持断点、单步、观察/修改存储器和寄存器
- 比使用仿真芯片、目标仿真头和仿真插座的仿真系统有更优越的性能

电源电压输入：2.7V - 5.25V

- 使用片内电压调整器时，支持的电压范围为 3.6~5.25V

高速 8051 微控制器内核

- 流水线指令结构；70%的指令的执行时间为一个或两个系统时钟周期
- 速度可达 25MIPS（时钟频率为 25MHz 时）
- 扩展的中断系统

存储器

- 1536 B 内部数据 RAM(1K+256+256 USB FIFO)
- 16KB FLASH；可在系统编程，扇区大小为 512 字节

数字外设

- 15 个端口 I/O；均耐 5V 电压，大灌电流
- 增强型增强型 UART
- 2 个通用 16 位计数器/定时器

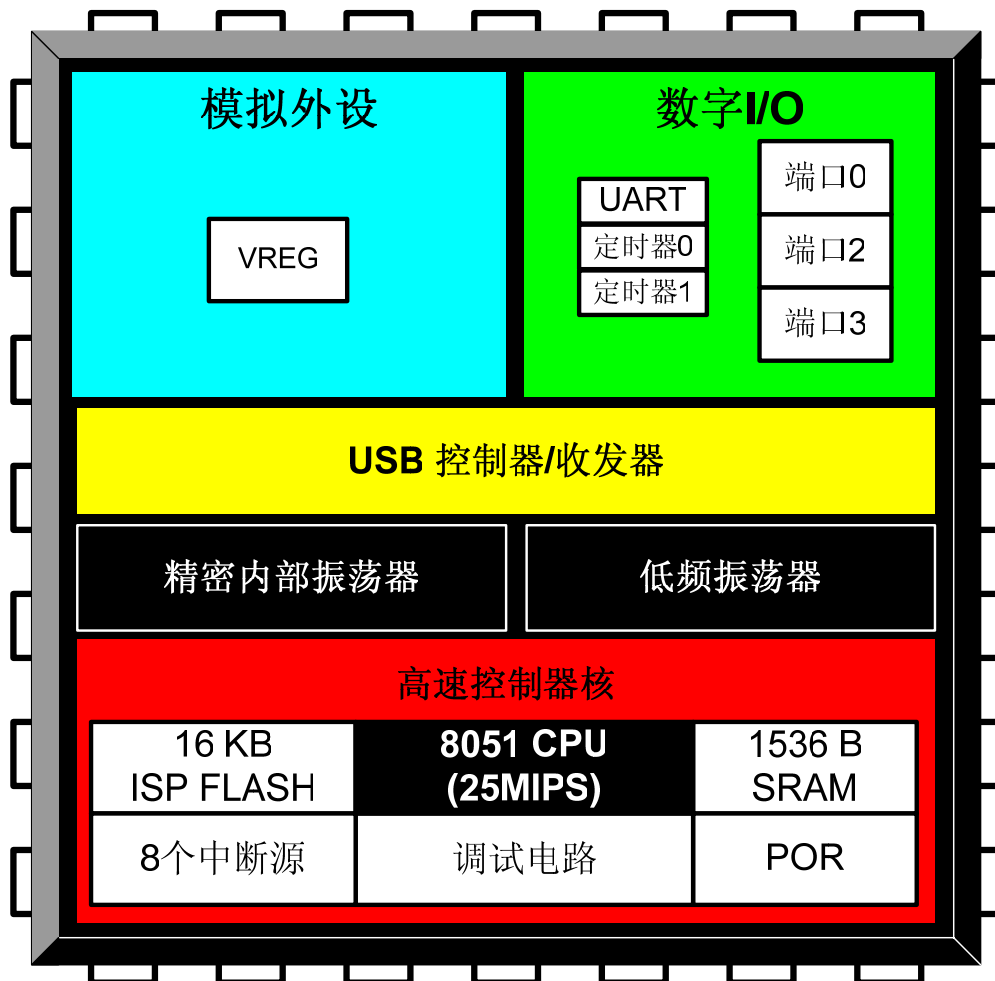
时钟源

- 内部振荡器：0.25%的精度(时钟恢复被使能时)。支持所有 USB 和 UART 方式
- 外部 CMOS 时钟
- 可在运行中切换时钟源，适用于节电方式

封装

- 28 脚 QFN

温度范围：-40°C - +85°C



C8051F326/7 全速 USB、16KB FLASH 微控制器

1. 系统概述	1
1.1 CIP-51™ 微控制器核	4
1.1.1 与 8051 完全兼容	4
1.1.2 速度提高	4
1.1.3 增加的功能	5
1.2 片内存储器	6
1.3 通用串行总线控制器	7
1.4 稳压器	7
1.5 片内调试电路	8
1.6 可编程数字 I/O	8
1.7 串行端口	9
2. 极限参数	10
3. 总体直流电气特性	11
4. 引脚和封装定义	12
5. 稳压器 (REG0)	18
5.1 稳压器方式选择	19
5.2 VBUS 检测	19
6. CIP-51 微控制器	23
6.1 指令集	24
6.1.1 指令和 CPU 时序	24
6.1.2 MOVX 指令和程序存储器	24
6.2 存储器组织	28
6.2.1 程序存储器	28
6.2.2 数据存储器	29
6.2.3 通用寄存器	29
6.2.4 位寻址空间	29
6.2.5 堆栈	29
6.2.6 特殊功能寄存器	30
6.2.7 寄存器说明	33
6.3 中断系统	36
6.3.1 MCU 中断源和中断向量	36
6.3.2 外部中断	36
6.3.3 中断优先级	37
6.3.4 中断响应时间	37

C8051F326/7 全速 USB、16KB FLASH 微控制器

6.3.5 中断寄存器说明.....	39
6.4 电源管理方式.....	43
6.4.1 空闲方式.....	43
6.4.2 停机方式.....	43
7. 复位源	45
7.1 上电复位.....	46
7.2 掉电复位和VDD监视器	47
7.3 外部复位.....	48
7.4 时钟丢失检测器复位.....	48
7.5 FLASH错误复位.....	48
7.6 软件复位.....	48
7.7 USB复位.....	48
8. FLASH存储器	51
8.1 FLASH存储器编程.....	51
8.1.1 FLASH锁定和关键码功能.....	51
8.1.2 FLASH擦除.....	51
8.1.3 FLASH写.....	52
8.2 非易失性数据存储	52
8.3 安全选项	53
9. 外部RAM.....	56
9.1 访问用户XRAM.....	56
9.2 访问USB FIFO	57
10. 振荡器	59
10.1 可编程内部振荡器	60
10.1.1 C8051F326/7 的内部振荡器编程.....	60
10.1.2 内部振荡器挂起方式.....	60
10.2 内部低频（L-F）振荡器.....	62
10.3 4倍时钟乘法器	63
10.4 系统时钟和USB时钟选择	64
10.4.1 系统时钟选择.....	64
10.4.2 USB时钟选择.....	64
11. 端口输入/输出.....	67
11.1 端口I/O初始化.....	69
11.2 通用端口I/O.....	70
12. 通用串行总线控制器（USB0）	74
12.1 端点地址.....	75

C8051F326/7 全速 USB、16KB FLASH 微控制器

12.2	USB收发器	75
12.3	USB寄存器访问	77
12.4	USB时钟配置	81
12.5	FIFO管理	82
12.5.1	FIFO分割方式	82
12.5.2	FIFO双缓冲	83
12.5.3	FIFO访问	83
12.6	功能地址	84
12.7	功能配置和控制	85
12.8	中断	88
12.9	串行接口引擎	91
12.10	端点 0	91
12.10.1	端点 0 SETUP 传输	91
12.10.2	端点 0 IN 传输	92
12.10.3	端点 0 OUT 传输	92
12.11	配置端点 1	94
12.12	控制端点 1 输入 (IN)	94
12.12.1	端点 1 输入 (IN) 中断或批量方式	94
12.12.2	端点 1 输入 (IN) 同步方式	95
12.13	控制端点 1 输出 (OUT)	98
12.13.1	端点 1 输出 (OUT) 中断或批量方式	98
12.13.2	端点 1 输出 (OUT) 同步方式	98
13.	UART0	102
13.1	波特率发生器	103
13.2	数据格式	104
13.3	配置和操作	106
13.3.1	数据发送	106
13.3.2	数据接收	106
13.3.3	多机通信	107
14.	定时器	111
14.1	定时器 0 和定时器 1 工作方式	111
14.1.1	方式 0 — 13 位计数器/定时器	111
14.1.2	方式 1 — 16 位计数器/定时器	113
14.1.3	方式 2 — 自动重载的 8 位计数器/定时器	113
14.1.4	方式 3 — 两个 8 位计数器/定时器 (仅定时器 0)	114
15.	C2 接口	119
15.1	C2 接口寄存器	119
15.2	C2 引脚共享	121

C8051F326/7 全速 USB、16KB FLASH 微控制器

1. 系统概述

C8051F326/7 器件是完全集成的混合信号片上系统型 MCU。下面列出了一些主要特性，有关某一产品的具体特性参见表 1.1。

- 高速、流水线结构的 8051 兼容的微控制器内核（可达 25MIPS）
- 全速、非侵入式的在系统调试接口（片内）
- 通用串行总线（USB）功能控制器，有 3 个固定功能的端点管道，集成收发器和 256 字节的 FIFO RAM
- 电源稳压器
- 高精度可编程的 12MHz 内部振荡器和 4 倍时钟乘法器
- 16KB 片内 FLASH 存储器
- 1536 字节片内 RAM（256B+1KB+256B USB FIFO）
- 增强型 UART，硬件实现的串行接口
- 2 个通用的 16 位定时器
- 片内上电复位、VDD 监视器和时钟丢失检测器
- 15 个端口 I/O（容许 5V 输入）

具有片内上电复位、VDD 监视器、电压调整器和时钟振荡器的 C8051F326/7 是真正能独立工作的片上系统。FLASH 存储器还具有在系统重新编程能力，可用于非易失性数据存储，并允许现场更新 8051 固件。用户软件对所有外设具有完全的控制，可以关断任何一个或所有外设以节省功耗。

片内 Silicon Labs 二线（C2）开发接口允许使用安装在最终应用系统上的产品 MCU 进行非侵入式（不占用片内资源）、全速、在系统调试。调试逻辑支持观察和修改存储器和寄存器，支持设置断点、单步、运行和停机命令。在使用 C2 进行调试时，所有的模拟和数字外设都可全功能运行。两个 C2 接口引脚可以与用户功能共享，使在系统调试功能不占用封装引脚。

每种器件都可在工业温度范围（-45℃到+85℃）内用 2.7V-5.25V 的电压工作。当电压大于 3.6V 时，必须使用内部电压调整器。对于 USB 通信，电源电压最小值为 3.0V。端口 I/O 和 /RST 引脚都容许 5V 的输入信号电压。C8051F326/7 采用两种引出脚不同的 28 脚 QFN 封装。符合 RoHS 规范的器件型号带有后缀-GQ。C8051F326 的端口 I/O 由一个单独的 I/O 电源引脚供电，可以与低电压逻辑接口。

表 1.1 产品选择指南

	MIPS（峰值）	FLASH 存储器	RAM	校准的内部振荡器	USB	电源电压调整器	UART	定时器(16 位)	数字端口 I/O	独立的 I/O 电源	封装
C8051F326	25	16K	1536	√	√	√	√	2	15	√	QFN-28
C8051F326-GQ	25	16K	1536	√	√	√	√	2	15	√	QFN-28
C8051F327	25	16K	1536	√	√	√	√	2	15	—	QFN-28
C8051F327-GQ	25	16K	1536	√	√	√	√	2	15	—	QFN-28

C8051F326/7 全速 USB、16KB FLASH 微控制器

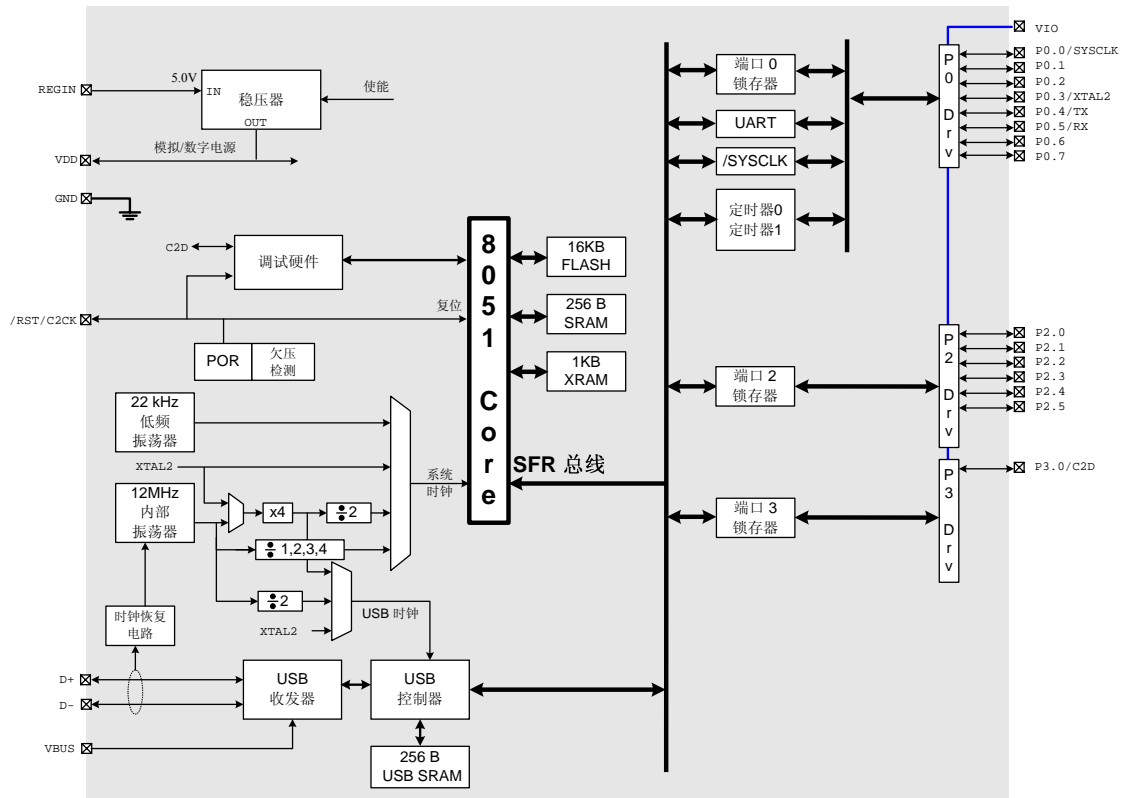


图 1.1 C8051F326 原理框图

C8051F326/7 全速 USB、16KB FLASH 微控制器

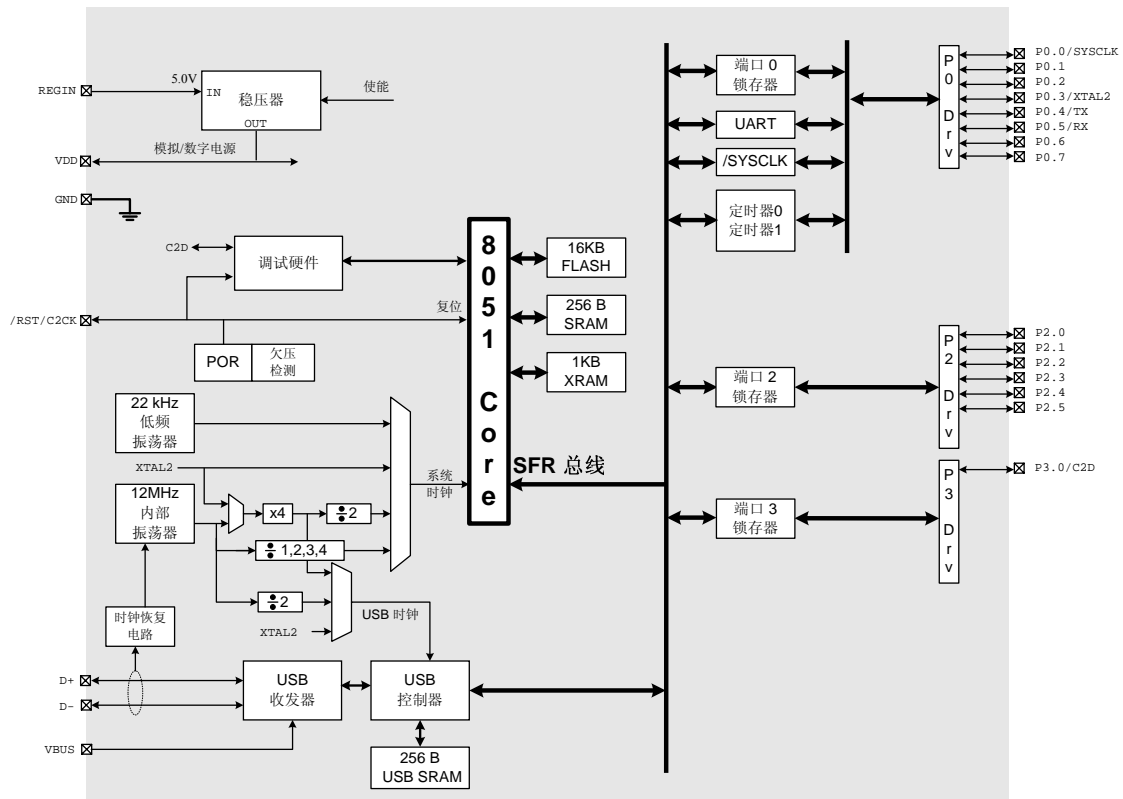


图 1.2 C8051F327 原理框图

C8051F326/7 全速 USB、16KB FLASH 微控制器

1.1 CIP-51™ 微控制器核

1.1.1 与 8051 完全兼容

C8051F326/7 系列器件使用Silicon Labs的专利CIP-51 微控制器内核。CIP-51 与MCS-51™ 指令集完全兼容，可以使用标准 803x/805x的汇编器和编译器进行软件开发。CIP-51 内核具有**标准 8052**的所有外设部件，包括 2 个 16 位计数器/定时器、一个具有增强波特率配置的全双工UART、1536 字节内部RAM、128 字节特殊功能寄存器（SFR）地址空间及 15 个I/O引脚。

1.1.2 速度提高

CIP-51 采用流水线结构，与标准的 8051 结构相比指令执行速度有很大的提高。在一个标准的 8051 中，除 MUL 和 DIV 以外所有指令都需要 12 或 24 个系统时钟周期，最大系统时钟频率为 12-24MHz。而对于 CIP-51 内核，70%的指令的执行时间为 1 或 2 个系统时钟周期，只有 4 条指令的执行时间大于 4 个系统时钟周期。

CIP-51 共有 111 条指令。下表列出了指令条数与执行时所需的系统时钟周期数的关系。

执行周期数	1	2	2/3	3	3/4	4	4/5	5	8
指令数	26	50	5	16	7	3	1	2	1

CIP-51 工作在最大系统时钟频率 25MHz 时，它的峰值速度达到 25MIPS。图 1.3 给出了几种 8 位微控制器内核工作在最大系统时钟时峰值速度的比较关系。

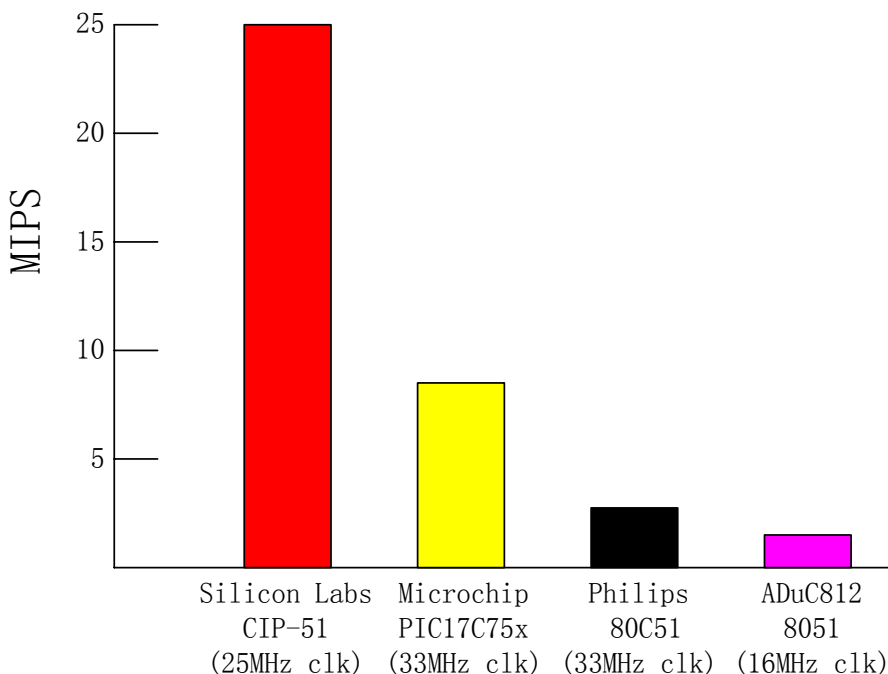


图 1.3 MCU 峰值执行速度比较

C8051F326/7 全速 USB、16KB FLASH 微控制器

1.1.3 增加的功能

C8051F326/7 系列 MCU 在 CIP-51 内核和外设方面有几项关键性的改进，提高了整体性能，更易于在最终应用中使用。

扩展的中断系统向 CIP-51 提供 8 个中断源。一个中断驱动的系统需要较少的 MCU 干预，因而有更高的执行效率。在设计一个多任务实时系统时，这些增加的中断源是非常有用的。

C8051F326/7 有多达 7 个复位源：上电复位电路（POR）、片内 VDD 监视器（当电源电压低于 V_{RST} 时强制复位）、USB 控制器（USB 总线复位或 VBUS 状态变化）、时钟丢失检测器、软件强制复位、外部复位输入引脚和 FLASH 读/写错误保护电路复位。除了 POR、复位输入引脚及 FLASH 操作错误这三个复位源之外，其他复位源都可以被软件禁止。

C8051F326/7 器件的内部振荡器在出厂时已经被校准为 $12\text{MHz} \pm 1.5\%$ ，该振荡器的周期可以由用户以大约 0.25% 的增量编程。片内还有一个低频振荡器，便于低功耗操作。时钟恢复电路允许内部振荡器与 4 倍时钟乘法器配合，提供全速方式 USB 时钟源。内部振荡器还被用作低速方式下的 USB 时钟源。外部 CMOS 时钟也可以与 4 倍时钟乘法器配合使用。系统时钟可以被配置为使用内部振荡器、外部时钟、低频振荡器或时钟乘法器输出二分频。如果需要，可以在 CPU 运行时切换系统振荡源。外部时钟和内部低频振荡器在低功耗系统中是非常有用的，它允许 MCU 从一个低频率（节电）时钟源运行，当需要时再周期性地切换到高速的内部振荡器。

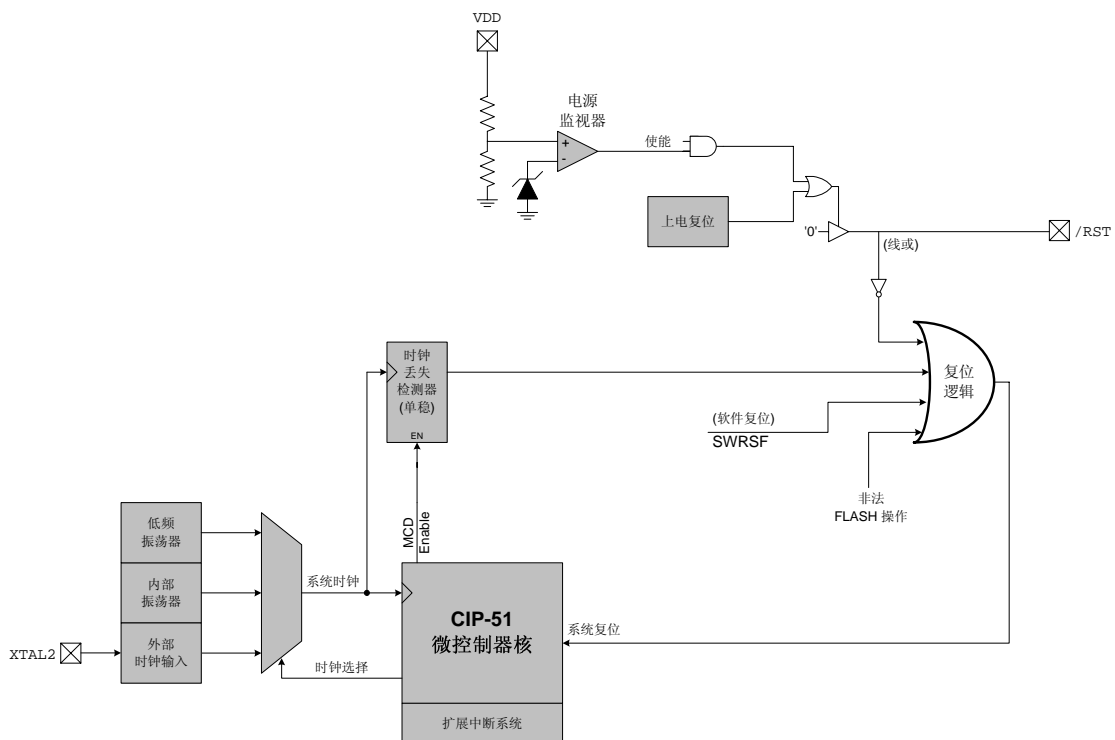


图 1.4 片内时钟和复位电路

C8051F326/7 全速 USB、16KB FLASH 微控制器

1.2 片内存储器

CIP-51 有标准 8051 的程序和数据地址配置。它包括 256 字节的数据 RAM，其中高 128 字节为双映射。用间接寻址访问通用 RAM 的高 128 字节，用直接寻址访问 128 字节的 SFR 地址空间。数据 RAM 的低 128 字节可用直接或间接寻址方式访问。前 32 个字节为 4 个通用寄存器区，接下来的 16 字节既可以按字节寻址也可以按位寻址。

程序存储器包含 16KB 的 FLASH。该存储器以 512 字节为一个扇区，可以在系统编程，且不需特别的编程电压。图 1.5 给出了 MCU 系统的存储器结构。

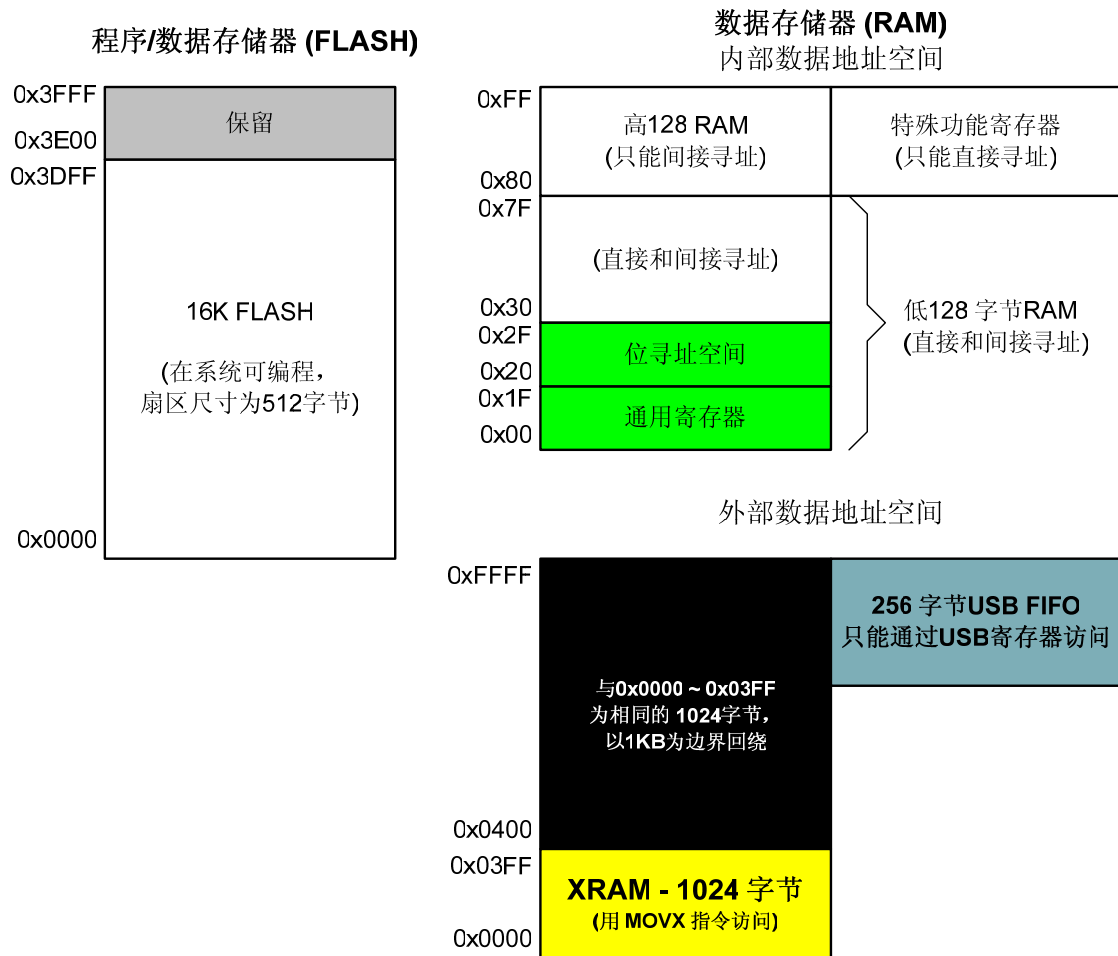


图 1.5 片内存储器组织

C8051F326/7 全速 USB、16KB FLASH 微控制器

1.3 通用串行总线控制器

通用串行总线控制器（USB0）符合USB2.0规范，集成了收发器和端点FIFO RAM，支持全速或低速操作。共有3个端点管道：一个双向控制端点（端点0）和一个数据端点（端点1），数据端点具有一个IN管道和一个OUT管道。

XRAM中有256B的存储块被专门用作USB FIFO空间。该FIFO空间被分配给端点0和端点1。端点0占64字节，端点1的IN管道和OUT管道分别占64字节和128字节。

USB0可以工作在全速或低速方式。片内4倍时钟乘法器和时钟恢复电路允许使用内部高精度振荡器作为USB时钟源，实现全速和低速通信。外部时钟源也可以与4倍时钟乘法器配合使用来产生USB时钟。

USB收发器符合USB2.0规范，并包含内部匹配和上拉电阻。上拉电阻可以被用软件使能/禁止，可以根据软件选择的速度设置（全速或低速）出现在D+或D-引脚。

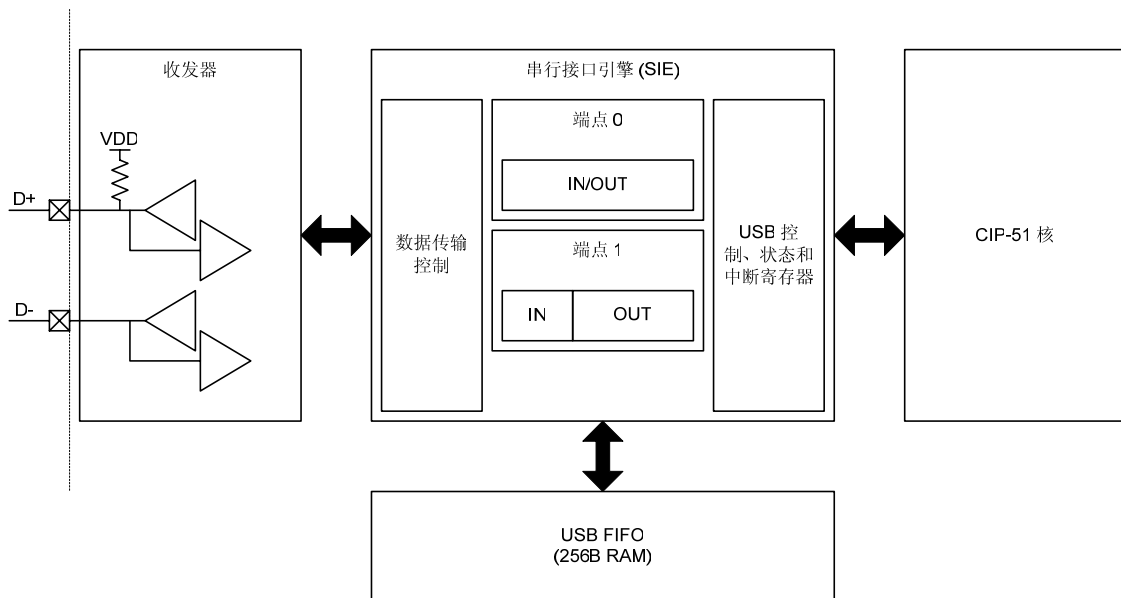


图1.6 USB控制器原理框图

1.4 稳压器

C8051F326/7器件内部包含一个电压调整器（REG0）。当被使能时，REG0输出连到VDD引脚，并可为外部器件提供电源。REG0可以被软件使能/禁止。

C8051F326/7 全速 USB、16KB FLASH 微控制器

1.5 片内调试电路

C8051F326/7器件具有片内Silicon Labs 2线（C2）接口调试电路，支持使用安装在最终应用系统中的产品器件进行非侵入式、全速的在系统调试。

Silicon Labs的调试系统支持观察和修改存储器和寄存器，支持断点和单步执行。不需要额外的目标RAM、程序存储器、定时器或通信通道。在调试时所有的模拟和数字外设都正常工作。当MCU单步执行或遇到断点而停止运行时，所有的外设（USB除外）都停止运行，以保持与指令执行同步。

开发套件C8051F326DK具有开发应用代码和对C8051F326/7 MCU进行在系统调试所需要的全部硬件和软件。开发套件中包括Windows开发环境、一个连接C2端口的串行适配器和一个目标应用板。套件中还包括所有必须的通信电缆及一个墙装电源。对于开发和调试嵌入式应用来说，Silicon Laboratories的调试环境比采用标准MCU仿真器要优越得多。标准的MCU仿真器要使用在板仿真芯片和目标电缆，还需要在应用板上有MCU的插座。Silicon Laboratories的调试环境既便于使用又能保持高精度模拟外设的性能。

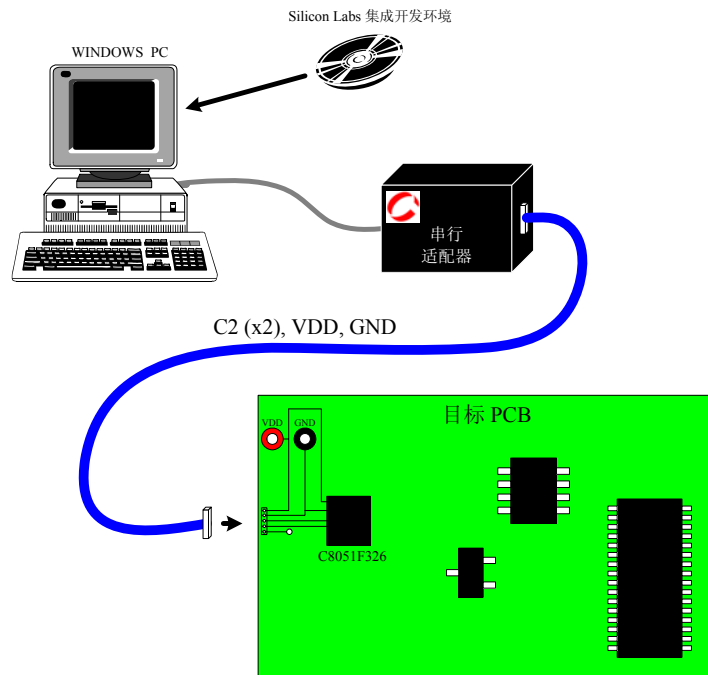


图1.7 开发/在系统调试示意图

1.6 可编程数字 I/O

C8051F326/7器件有15个I/O引脚（一个8位口、一个6位口和一个1位口）。C8051F326/7端口的工作情况与标准8051相似，但有一些改进。每个端口引脚都可以被配置为数字输入或输出引脚。被选择作为数字输出的引脚还可以被配置为推挽或漏极开路输出。在标准8051中固定的“弱上拉”可以被总体禁止，提供了节能能力。

C8051F326/7 全速 USB、16KB FLASH 微控制器

1.7 串行端口

C8051F326/7系列MCU包含一个具有增强型波特率配置的全双工UART。该串行接口完全用硬件实现，能向CIP-51产生中断，因此需要很少的CPU干预。

C8051F326/7 全速 USB、16KB FLASH 微控制器

2. 极限参数

表 2.1 极限参数*

参 数	条 件	最小值	典型值	最大值	单位
环境温度（通电情况下）		-55		125	℃
储存温度		-65		150	℃
任何端口I/O引脚或/RST相对GND的电压		-0.3		5.8	V
VDD或VIO相对GND的电压		-0.3		4.2	V
通过VDD、VIO和GND的最大总电流				500	mA
/RST或任何端口引脚的最大输出灌电流				100	mA

*注：超过这些列出的“极限参数”可能导致器件永久性损坏。长时间在最大允许值或超过最大允许值的条件下工作可能影响器件的可靠性。

C8051F326/7 全速 USB、16KB FLASH 微控制器

3. 总体直流电气特性

表 3.1 总体直流电气特性

-40℃到+85℃，25MHz 系统时钟（除非特别说明）。

参 数	条 件	最小值	典型值	最大值	单 位
I/O电源电压(VIO) ^{1,2}		1.8	3.3	3.6	V
内核电源电压(VDD) ³		2.7	3.3	3.6	V
内核电源电流（CPU活动）	VDD=3.3V, CLK=24MHz	—	11	—	mA
	VDD=3.3V, CLK= 3MHz	—	1.9	—	mA
	VDD=3.3V, CLK=32kHz	—	20	—	μA
内核电源电流（CPU和USB处于活动状态，全速或低速）	VDD=3.3V, CLK=24MHz	—	TBD	—	mA
	VDD=3.3V, CLK= 6MHz	—	TBD	—	mA
内核电源电流（CPU不活动，即不访问FLASH）	VDD=3.3V, CLK=24MHz	—	4.4	—	mA
	VDD=3.3V, CLK= 3MHz	—	0.83	—	mA
	VDD=3.3V, CLK=32kHz	—	13	—	μA
数字电源电流（挂起方式或停机方式）	振荡器停止运行	—	<0.1	—	μA
RAM数据保持电源电压		—	1.5	—	V
SYSCLK（系统时钟） ⁴		0	—	25	MHz
TSYSH(SYSCLK高电平时间)		18	—	—	ns
TSYSL(SYSCLK低电平时间)		18	—	—	ns
额定工作温度范围		-40	—	+85	℃
注： 1. I/O 电源电压（VIO）必须小于或等于内核电源电压（VDD）。 2. 对于 C8051F327，VIO 在内部连接到 VDD。 3. USB 需要最小 3.0V 的内核电源电压。 4. 为能使用调试功能，SYSCLK 至少应为 32kHz。					

C8051F326/7 全速 USB、16KB FLASH 微控制器

4. 引脚和封装定义

表 4.1 C8051F326/7 引脚定义

引脚名称	引脚号		引脚类型	说 明
	F326	F327		
VDD	6	6	电源输入 电源输出	2.7V – 3.6V 内核电源电压输入 3.3V 稳压器输出。见第 5 章。
VIO	5	—	电源输入	I/O 电源电压输入。对于 F326，该引脚的电压必须小于或等于内核电源电压（VDD）；对于 F327，该引脚在内部连接到 VDD。
GND	2	3		地。
/RST	9	9	数字 I/O	器件复位。内部上电复位或 VDD 监视器的漏极开路输出。一个外部源可以通过将该引脚驱动为低电平（至少 15μs）来启动一次系统复位。见第 7 章。
C2CK			数字 I/O	C2 调试接口的时钟信号。
P3.0	10	10	数字 I/O	端口 P3.0
C2D			数字 I/O	C2 调试接口的双向数据信号。
REGIN	7	7	电源输入	稳压器的 5V 输入。该引脚是片内稳压器的输入。
VBUS	8	8	数字输入	VBUS 检测输入。该引脚应被连接到 USB 网络的 VBUS 信号。该引脚上出现 5V 信号表示有 USB 连接。
D+	3	4	数字 I/O	USB 的 D+。
D-	4	5	数字 I/O	USB 的 D-。
P0.0	1	2	数字 I/O	端口 P0.0
P0.1	28	1	数字 I/O	端口 P0.1
P0.2	27	28	数字 I/O	端口 P0.2
P0.3	26	27	数字 I/O	端口 P0.3
XTAL2			数字输入	外部时钟输入。见第 10 章。
P0.4	25	26	数字 I/O	端口 P0.4
P0.5	24	25	数字 I/O	端口 P0.5
P0.6	23	24	数字 I/O	端口 P0.6
P0.7	22	23	数字 I/O	端口 P0.7
P2.0	19	19	数字 I/O	端口 P2.0
P2.1	18	18	数字 I/O	端口 P2.1
P2.2	12	12	数字 I/O	端口 P2.2
P2.3	11	11	数字 I/O	端口 P2.3
P2.4	17	17	数字 I/O	端口 P2.4
P2.5	16	16	数字 I/O	端口 P2.5
F326 的 13、14、15、20 和 21 脚为无连接引脚。 F327 的 13、14、15、20、21 和 22 脚为无连接引脚。				

C8051F326/7 全速 USB、16KB FLASH 微控制器

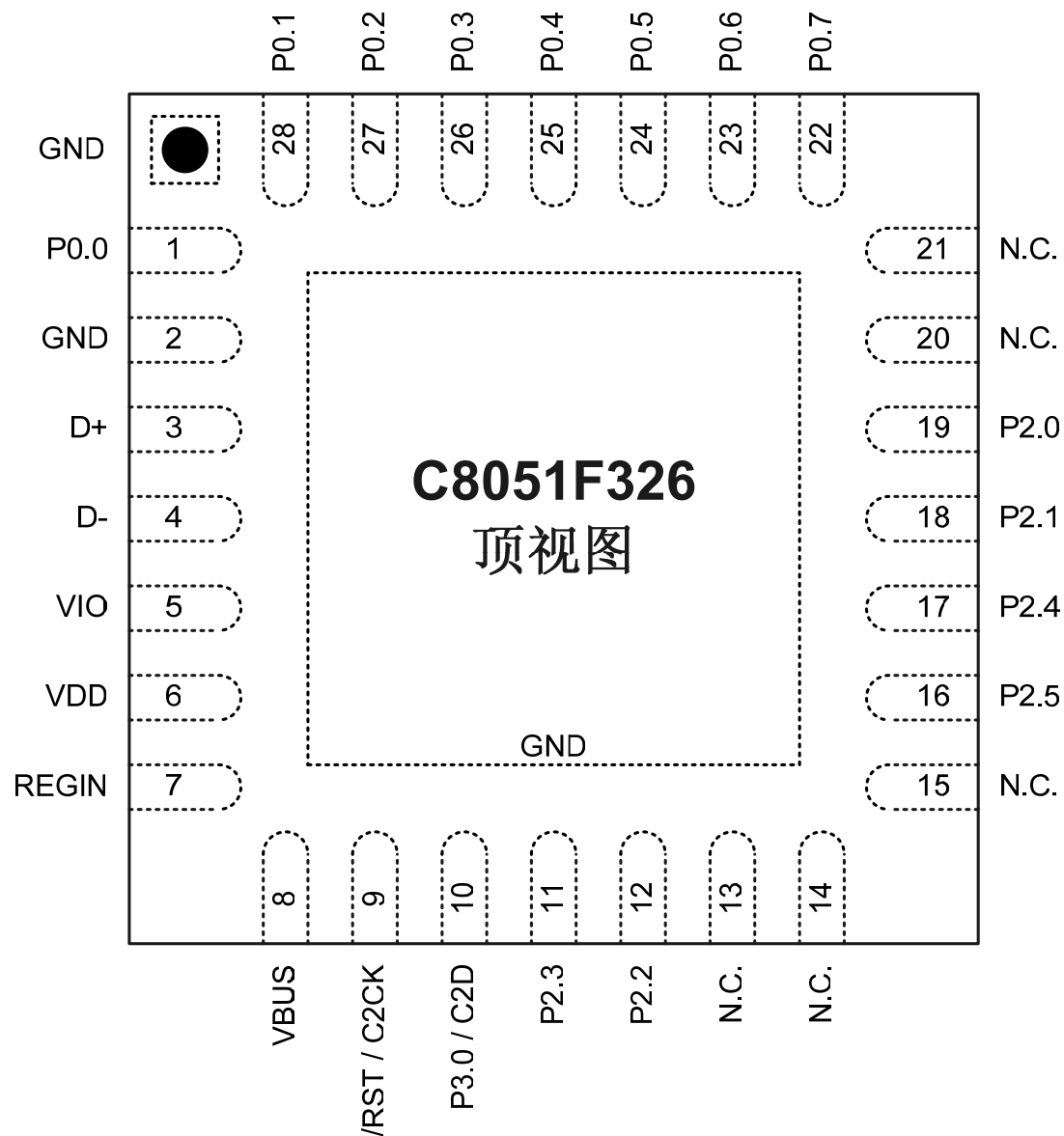


图 4.1 C8051F326 QFN-28 引脚图（顶视图）

C8051F326/7 全速 USB、16KB FLASH 微控制器

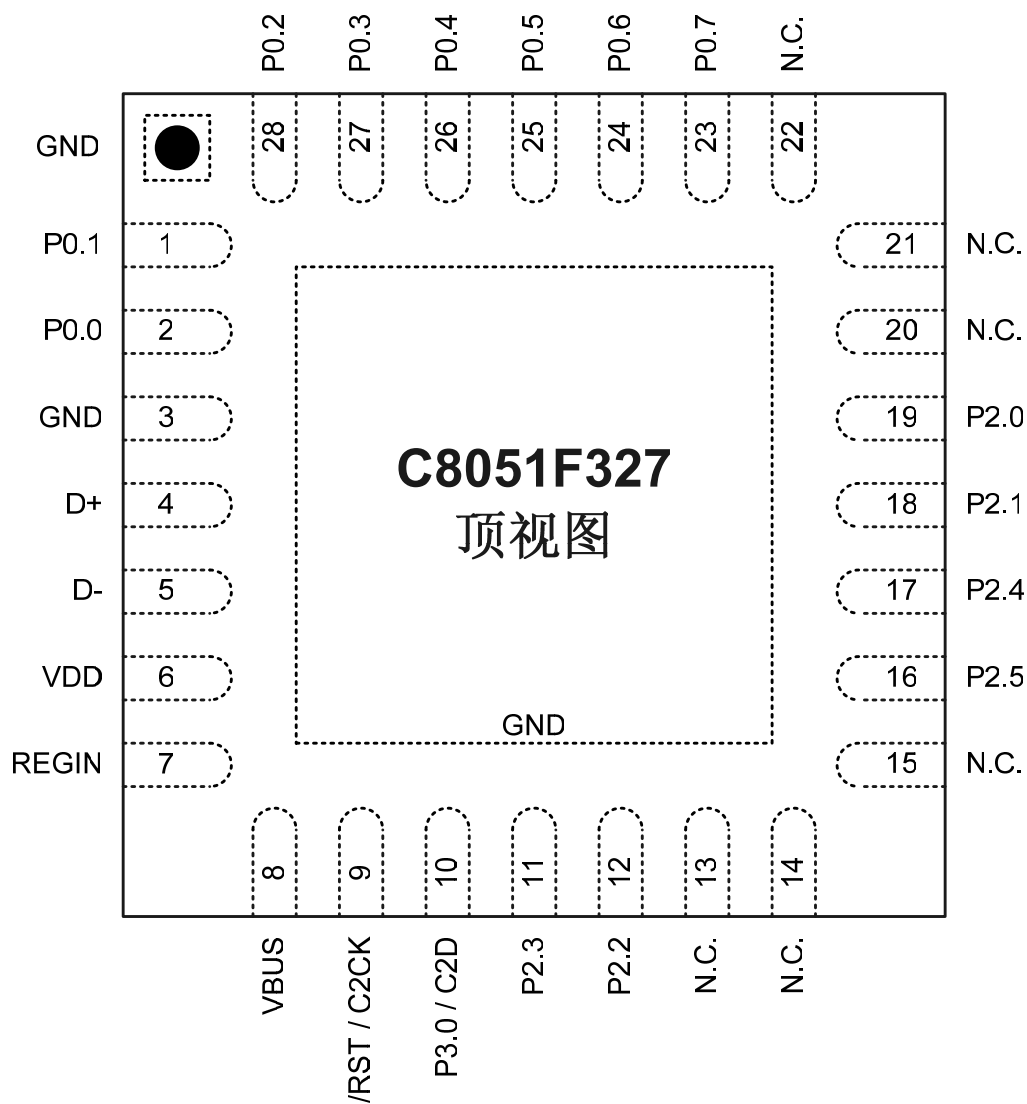


图 4.1 C8051F327 QFN-28 引脚图（顶视图）

C8051F326/7 全速 USB、16KB FLASH 微控制器

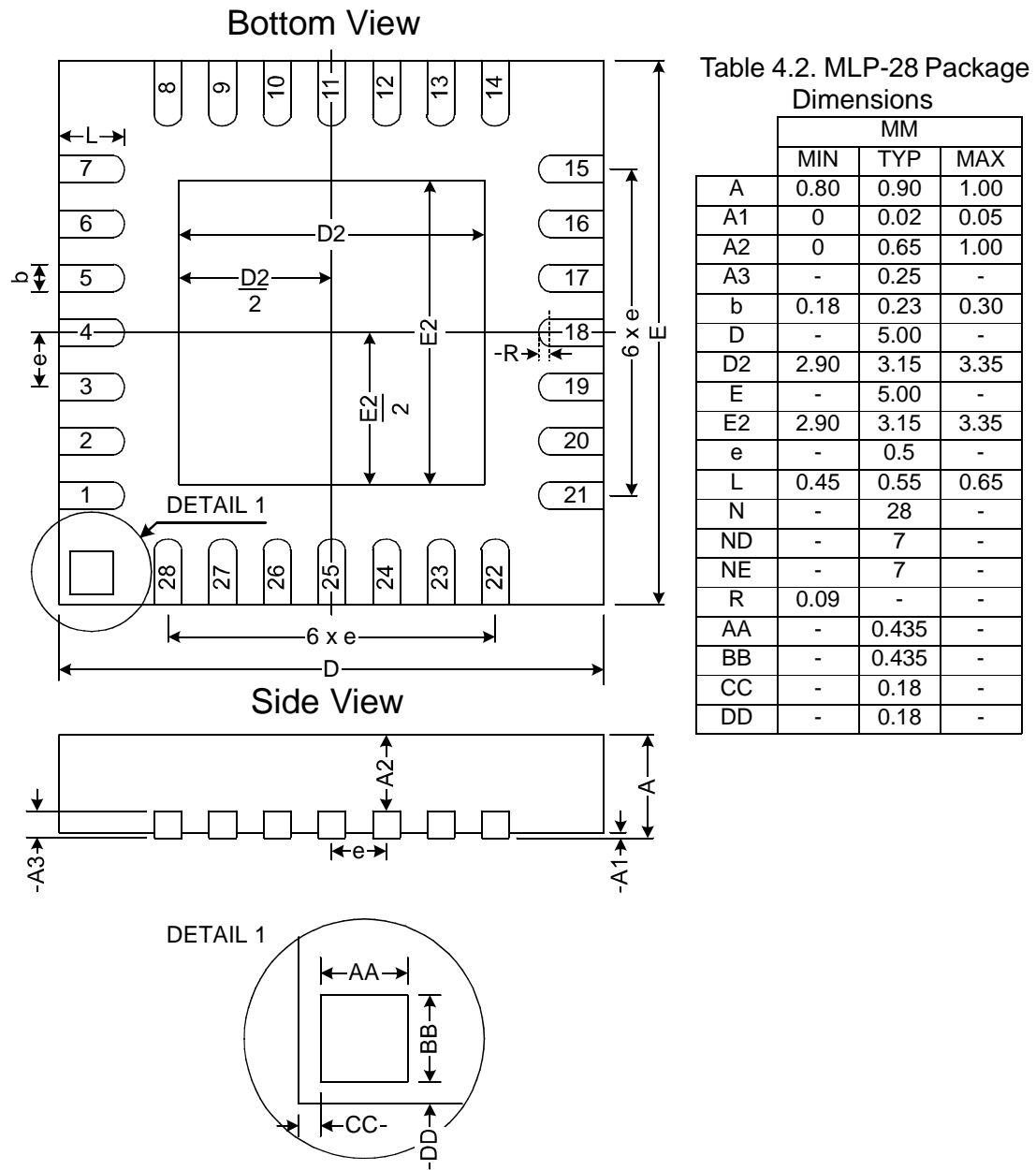


图 4.3 QFN-28 封装图

C8051F326/7 全速 USB、16KB FLASH 微控制器

Top View

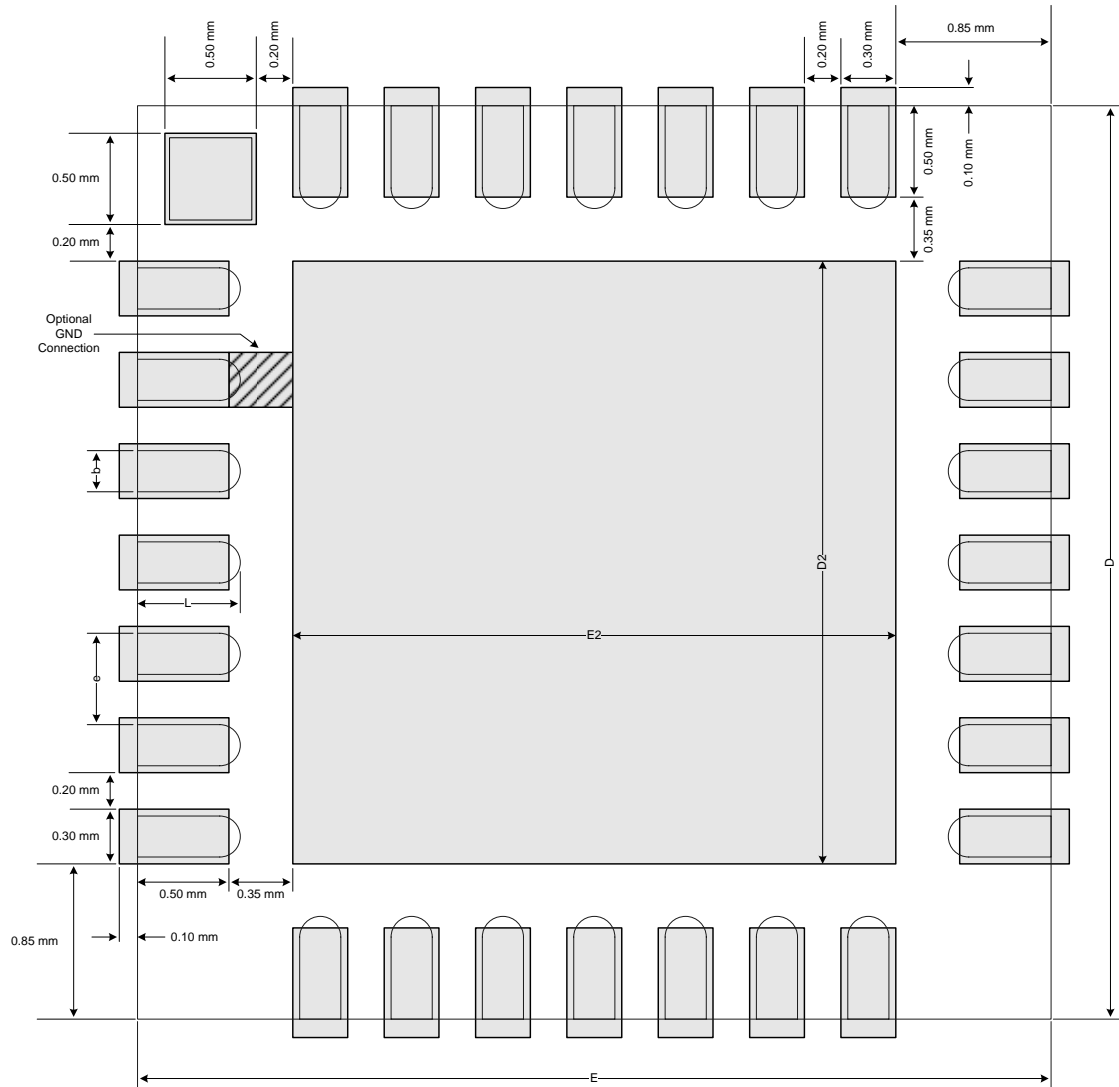


图 4.4 典型的 C8051F326 QFN-28 焊盘图

C8051F326/7 全速 USB、16KB FLASH 微控制器

Top View

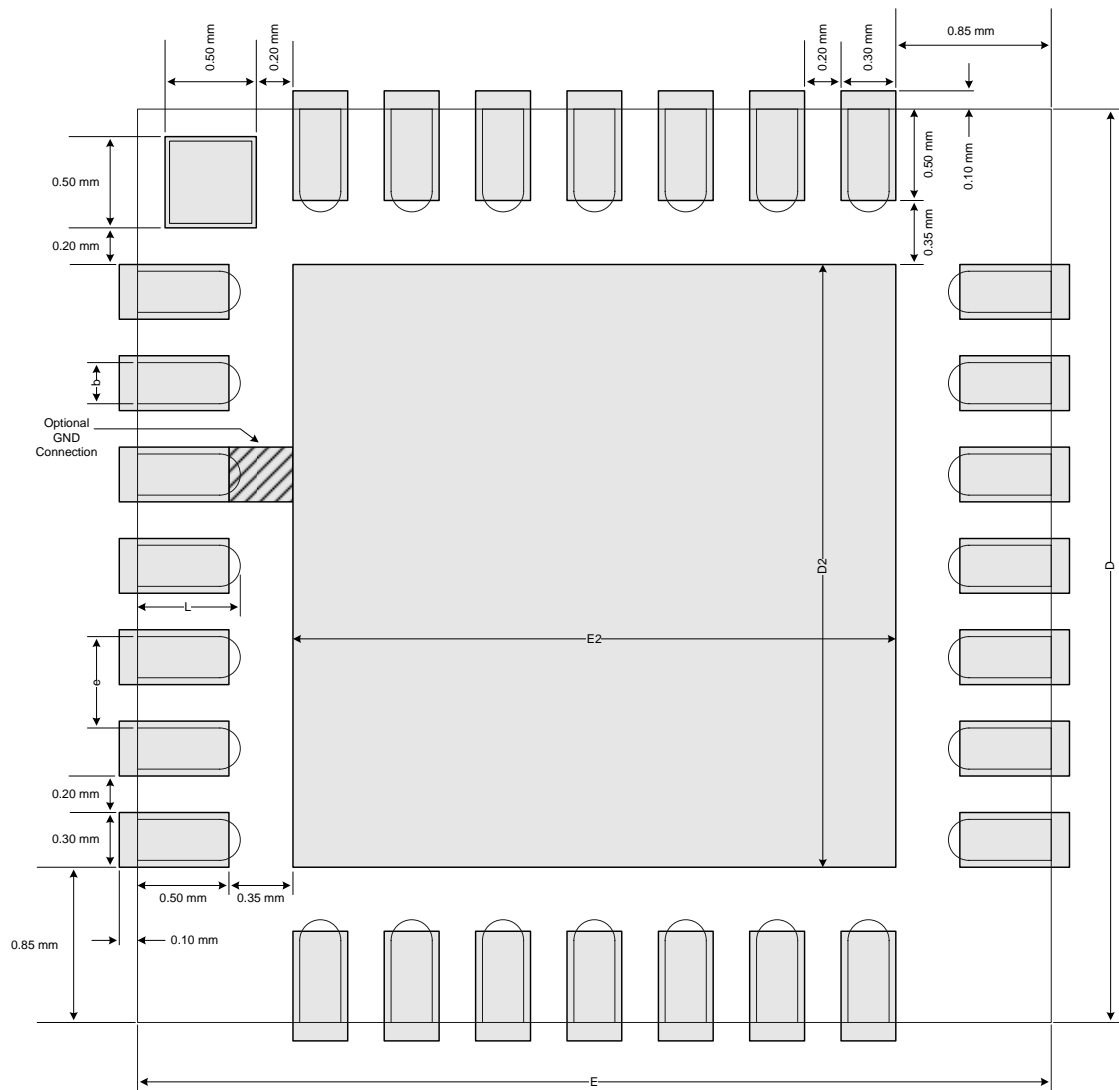


图 4.5 典型的 C8051F327 QFN-28 焊盘图

Top View

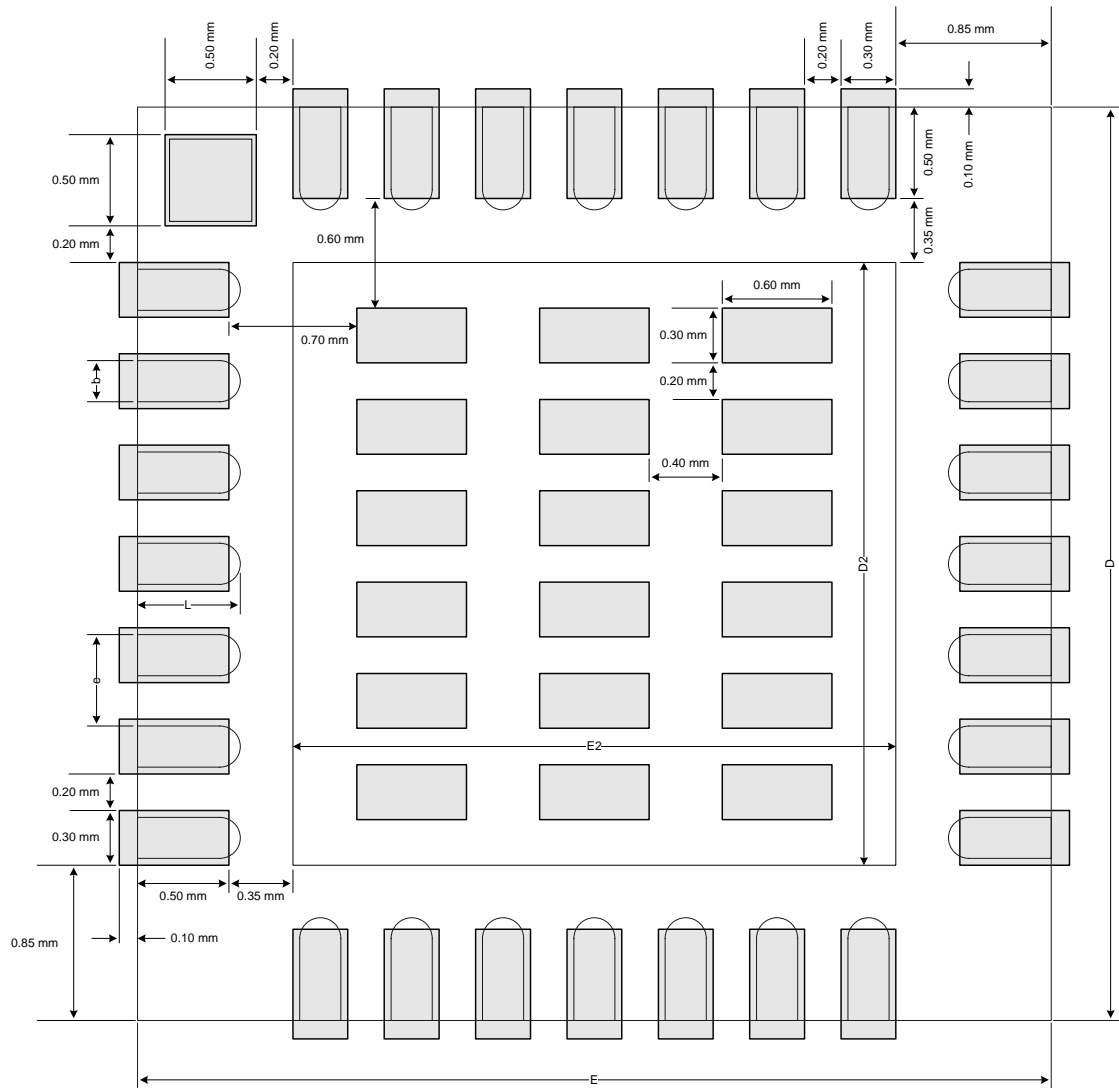


图 4.6 典型的 QFN-28 锡膏层

5. 稳压器（REG0）

C8051F326/7器件内部包含一个稳压器（REG0）。当被使能时，REG0输出连到VDD引脚，并可为外部器件提供电源。可以用软件通过寄存器REG0CN中的REGEN位使能/禁止REG0。REG0的电气特性见表5.1。

稳压器在复位时被使能。当器件用 3V 电源网络工作在自供电方式时，稳压器可以被禁止，以节省功耗。注意：如果稳压器的输入电压大于内核电源电压，则稳压器不应被禁止；否则可能造成器件的永久性损坏。

C8051F326/7 全速 USB、16KB FLASH 微控制器

当在 USB 网络中使用 C8051F326/7 器件时，VBUS 信号必须被连到 VBUS 引脚。VBUS 信号只应在器件工作于总线供电方式时被连接到 REGIN 引脚。REG0 的配置选项示于图 5.1 ~ 图 5.4 中。

5.1 稳压器方式选择

REG0 提供低功耗工作方式，在器件处于挂起方式时使用。在低功耗方式，REG0 输出电压保持正常，但 REG0 的动态性能（响应时间）降低。表 5.1 给出了正常方式和低功耗方式下的电源电流。REG0 方式选择由寄存器 REG0CN 中的 REGMOD 位控制。

5.2 VBUS 检测

当使用 USB 功能控制器时，VBUS 信号应被连接到 VBUS 引脚。VBSTAT 位（寄存器 REG0CN）指示 VBUS 信号的当前逻辑电平。当 VBUS 信号与寄存器 REG0CN 中的 VBPOL 位所选择的极性一致时，会产生 VBUS 中断（如果被允许）。VBUS 中断是电平敏感的，没有相应的中断请求标志。只要 VBUS 信号与 VBPOL 选择的极性一致，VBUS 中断就有效。表 5.1 给出了 VBUS 输入信号的参数。

注意：在 USB0 被选择为复位源的情况下，当 VBUS 信号与 VBPOL 位选择的极性匹配时会导致产生系统复位。有关选择 USB 为复位源的详细信息，见“7. 复位源”

表 5.1 稳压器电气特性

VDD=3.0V，-40℃到+85℃（除非特别说明）

参 数	条 件	最小值	典型值	最大值	单 位
输入电压范围		2.7	—	5.25	V
输出电压	输出电流 = 1 ~ 100 mA	3.0	3.3	3.6	V
VBUS 检测输入门限		1.0	1.8	4.0	V
偏置电流	正常方式（REGMOD = 0）	—	75	TBD	μA
	低功耗方式（REGMOD = 1）	—	41	TBD	
电压降落（V _{DO} ）*	IDD = 1 mA		1		mV/mA
	IDD = 100 mA		100		
*注：最小输入电压为 2.7V或V _{DD} +V _{DO} （最大负载）中的较大者。					

C8051F326/7 全速 USB、16KB FLASH 微控制器

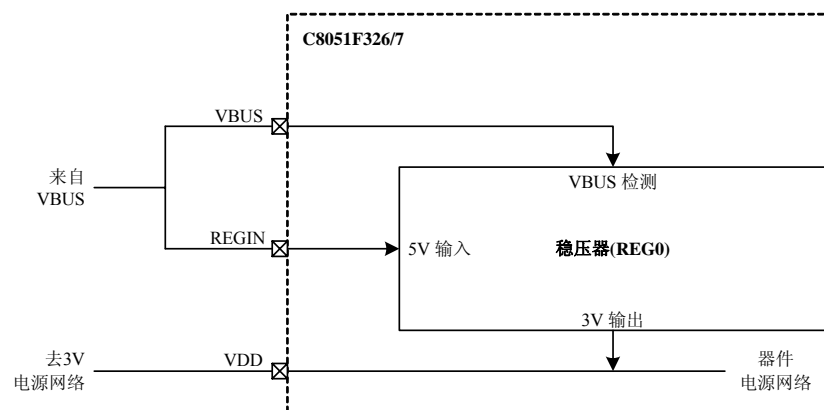


图 5.1 REG0 配置: USB 总线供电

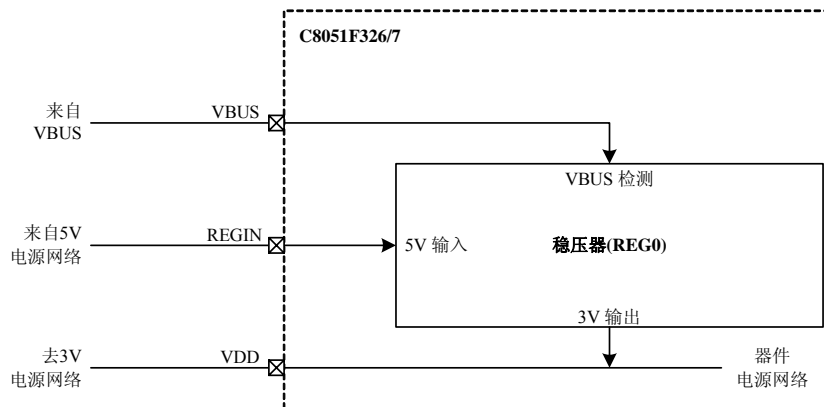


图 5.2 REG0 配置: USB 自供电

C8051F326/7 全速 USB、16KB FLASH 微控制器

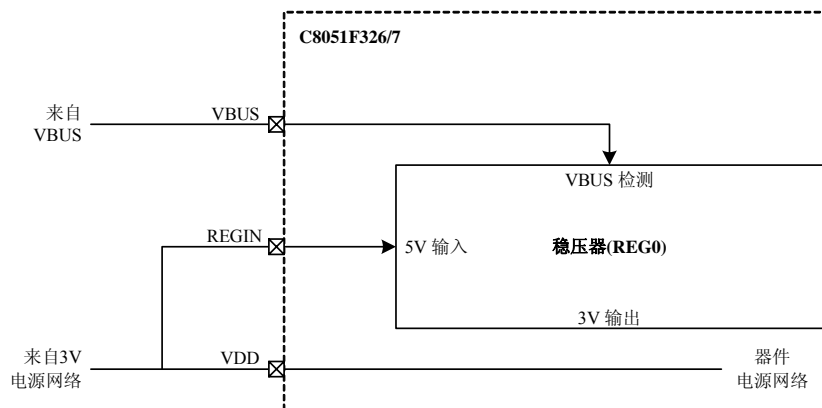


图 5.3 REG0 配置: USB 自供电, 稳压器禁止

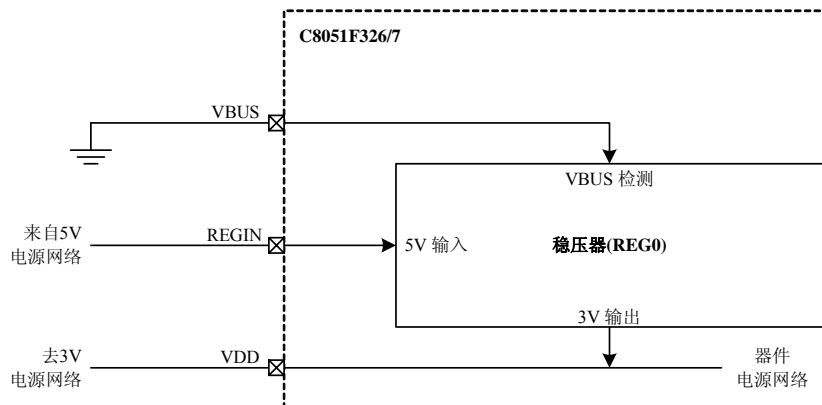


图 5.4 REG0 配置: 无 USB 连接

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 5.1 REG0CN: 稳压器控制寄存器

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	复位值
REGDIS	VBSTAT	VBPOL	REGMOD	保留	保留	保留	保留	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xC9
位 7: REGDIS: 稳压器禁止位								
0: 稳压器使能。								
1: 稳压器禁止。。								
位 6: VBSTAT: VBUS 信号状态位								
0: 未检测到 VBUS 信号（器件未连接到 USB 网络）。								
1: 检测到 VBUS 信号（器件连接到 USB 网络）。								
位 5: VBPOL: VBUS 中断极性选择位								
该位选择 VBUS 中断极性。								
0: 当 VBUS 为低电平时 VBUS 中断有效。								
1: 当 VBUS 高低电平时 VBUS 中断有效。								
位 4: REGMOD: 稳压器方式选择位								
该位选择稳压器工作方式。当 REGMOD 被置 1 时，稳压器工作在低功耗（挂起）方式。								
0: USB0 稳压器工作在正常方式。								
1: USB0 稳压器工作在低功耗方式。								
位 3-0: 保留。读 = 0000b。必须写 0000b。								

6. CIP-51 微控制器

MCU系统控制器的内核是CIP-51 微控制器。CIP-51 与MCS-51TM指令集完全兼容，可以使用标准 803x/805x的汇编器和编译器进行软件开发。该系列MCU具有标准 8051 的所有外设部件，包括 2 个 16 位的计数器/定时器（详见第 14 章）、一个增强型全双工UART（详见第 13 章）、256 字节内部RAM、128 字节特殊功能寄存器（SFR）地址空间及 15 个端口I/O（详见第 11 章）。CIP-51 还包含片内调试硬件（详见第 15 章），与USB和其它数字子系统直接接口，在一个集成电路内提供了完全的解决方案。

CIP-51 微控制器内核除了具有标准 8051 的组织结构和外设以外，另有增加的定制外设和功能，大大增强了它的处理能力（见图 6.1 的原理框图）。CIP-51 具有下列特性：

- 与 MCS-51 指令集完全兼容
- 在 25MHz 时钟时最大速度为 25MIPS
- 0 到 25MHz 的时钟频率
- 256 字节内部 RAM
- 15 个端口 I/O
- 扩展的中断处理系统
- 复位输入
- 电源管理方式
- 片内调试逻辑
- 程序和数据存储器安全

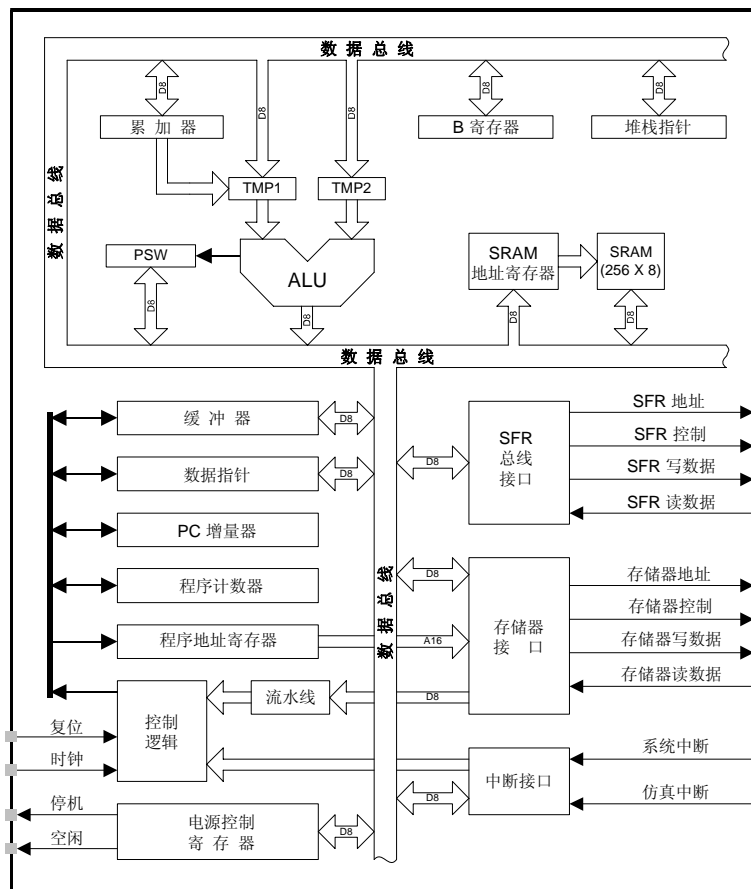


图 6.1 CIP-51 原理框图

C8051F326/7 全速 USB、16KB FLASH 微控制器

性能

CIP-51采用流水线结构，与标准的8051结构相比指令执行速度有很大的提高。在一个标准的8051中，除MUL和DIV以外所有指令都需要12或24个系统时钟周期，并且通常最大系统时钟频率为12 MHz。而对于CIP-51内核，70%的指令的执行时间为1或2个系统时钟周期，没有执行时间超过8个系统时钟周期的指令。

CIP-51 工作在最大系统时钟频率 25MHz 时，它的峰值速度达到 25MIPS。CIP-51 共有 111 条指令。下表列出了指令条数与执行时所需的系统时钟周期数的关系。

执行周期数	1	2	2/3	3	3/4	4	4/5	5	8
指令数	26	50	5	16	7	3	1	2	1

编程和调试支持

对 FLASH 程序存储器的在系统编程和与片内调试支持逻辑的通信是通过 Silicon Labs 2 线开发接口（C2）实现的。注意，可以在用户软件中使用 MOVC 和 MOVX 指令对可再编程的 FLASH 读和写，每次读或写一个字节。这一特性允许将程序存储器用于非易失性数据存储以及在软件控制下更新代码。

片内调试逻辑支持全速的在系统调试，允许设置硬件断点，支持开始、停止和单步执行（包括中断服务程序）命令，支持检查程序调用堆栈及读/写寄存器和存储器。这种片内调试方法完全是非侵入式的，不需要额外的目标 RAM、堆栈、定时器或其它片内资源。有关 C2 接口的详细信息见“15. C2 接口”。

CIP-51 有 Silicon Labs 和第三方供应商的开发工具支持。Silicon Labs 提供一个集成开发环境（IDE），包括编辑器、宏汇编器、调试器和编程器。IDE 的调试器和编程器与 CIP-51 之间通过 C2 接口，提供快速和有效的在系统编程和调试。也有第三方的宏汇编器和 C 编译器可用。

6.1 指令集

CIP-51 系统控制器的指令集与标准MCS-51TM指令集完全兼容，可以使用标准 8051 的开发工具开发CIP-51 的软件。所有的CIP-51 指令在二进制码和功能上与同类的MCS-51TM产品完全等价，包括操作码、寻址方式和对PSW标志的影响，但是指令时序与标准 8051 不同。

6.1.1 指令和 CPU 时序

在很多的 8051 产品中，机器周期和时钟周期是不同的，机器周期的长度在 2 到 12 个时钟周期之间。但是 CIP-51 只基于时钟周期，所有指令时序都以时钟周期计算。

由于 CIP-51 采用了流水线结构，大多数指令执行所需的时钟周期数与指令的字节数一致。条件转移指令在不发生转移时的执行周期数比发生转移时少一个。表 6.1 给出了 CIP-51 指令一览表，包括每条指令的助记符、字节数和时钟周期数。

6.1.2 MOVX 指令和程序存储器

MOVC 指令通常用于访问外部数据存储器（注：C8051F326/7 不支持外部数据或程序存储器）。在 CIP-51 中，MOVC 指令可用于访问片内 XRAM 或片内 FLASH 程序存储器。这一特性为 CIP-51 提供了由用户程序更新程序代码和将程序存储器空间用于非易失性数据存储的机制，详见“8. FLASH 存储器”。

C8051F326/7 全速 USB、16KB FLASH 微控制器

表 6.1 CIP-51 指令集

助记符	功能说明	字节数	时钟周期数
算术操作类指令			
ADD A,Rn	寄存器加到累加器	1	1
ADD A,direct	直接寻址字节加到累加器	2	2
ADD A,@Ri	间址 RAM 内容加到累加器	1	2
ADD A,#data	立即数加到累加器	2	2
ADDC A,Rn	寄存器加到累加器(带进位)	1	1
ADDC A,direct	直接寻址字节加到累加器(带进位)	2	2
ADDC A,@Ri	间址 RAM 加到累加器(带进位)	1	2
ADDC A,#data	立即数加到累加器(带进位)	2	2
SUBB A,Rn	累加器减去寄存器(带借位)	1	1
SUBB A,direct	累加器减去直接寻址字节(带借位)	2	2
SUBB A,@Ri	累加器减去间址 RAM(带借位)	1	2
SUBB A,#data	累加器减去立即数(带借位)	2	2
INC A	累加器加 1	1	1
INC Rn	寄存器加 1	1	1
INC direct	直接寻址字节加 1	2	2
INC @Ri	间址 RAM 加 1	1	2
DEC A	累加器减 1	1	1
DEC Rn	寄存器减 1	1	1
DEC direct	直接寻址字节减 1	2	2
DEC @Ri	间址 RAM 减 1	1	2
INC DPTR	数据地址加 1	1	1
MUL AB	累加器与寄存器 B 相乘	1	4
DIV AB	累加器除以寄存器 B	1	8
DA A	累加器十进制调整	1	1
逻辑操作类指令			
ANL A,Rn	寄存器“与”到累加器	1	1
ANL A,direct	直接寻址字节“与”到累加器	2	2
ANL A,@Ri	间址 RAM “与”到累加器	1	2
ANL A,#data	立即数“与”到累加器	2	2
ANL direct,A	累加器“与”到直接寻址字节	2	2
ANL direct,#data	立即数“与”到直接寻址字节	3	3
ORL A,Rn	寄存器“或”到累加器	1	1
ORL A,direct	直接寻址字节“或”到累加器	2	2
ORL A,@Ri	间址 RAM “或”到累加器	1	2
ORL A,#data	立即数“或”到累加器	2	2
ORL direct,A	累加器“或”到直接寻址字节	2	2
ORL direct,#data	立即数“或”到直接寻址字节	3	3
XRL A,Rn	寄存器“异或”到累加器	1	1
XRL A,direct	直接寻址字节“异或”到累加器	2	2
XRL A,@Ri	间址 RAM “异或”到累加器	1	2
XRL A,#data	立即数“异或”到累加器	2	2
XRL direct,A	累加器“异或”到直接寻址字节	2	2
XRL direct,#data	立即数“异或”到直接寻址字节	3	3
CLR A	累加器清零	1	1
CPL A	累加器求反	1	1
RL A	累加器循环左移	1	1
RLC A	带进位的累加器循环左移	1	1

C8051F326/7 全速 USB、16KB FLASH 微控制器

助记符	功能说明	字节数	时钟周期数
RR A	累加器循环右移	1	1
RRC A	带进位的累加器循环右移	1	1
SWAP A	累加器内高低半字节交换	1	1
数据传送类指令			
MOV A,Rn	寄存器传送到累加器	1	1
MOV A,direct	直接寻址字节传送到累加器	2	2
MOV A,@Ri	间址 RAM 传送到累加器	1	2
MOV A,#data	立即数传送到累加器	2	2
MOV Rn,A	累加器传送到寄存器	1	1
MOV Rn,direct	直接寻址字节传送到寄存器	2	2
MOV Rn,#data	立即数传送到寄存器	2	2
MOV direct,A	累加器传送到直接寻址字节	2	2
MOV direct,Rn	寄存器传送到直接寻址字节	2	2
MOV direct,direct	直接寻址字节传送到直接寻址字节	3	3
MOV direct,@Ri	间址 RAM 传送到直接寻址字节	2	2
MOV direct,#data	立即数传送到直接寻址字节	3	3
MOV @Ri,A	累加器传送到间址 RAM	1	2
MOV @Ri,direct	直接寻址字节传送到间址 RAM	2	2
MOV @Ri,#data	立即数传送到间址 RAM	2	2
MOV DPTR,#data16	16 位常数装入 DPTR	3	3
MOVC A,@A+DPTR	相对于 DPTR 的代码字节传送到累加器	1	3
MOVC A,@A+PC	相对于 PC 的代码字节传送到累加器	1	3
MOVX A,@Ri	外部 RAM(8 位地址)传送到累加器	1	3
MOVX @Ri,A	累加器传到外部 RAM (8 位地址)	1	3
MOVX A,@DPTR	外部 RAM(16 位地址)传送到累加器	1	3
MOVX @DPTR,A	累加器传到外部 RAM (16 位地址)	1	3
PUSH direct	直接寻址字节压入栈顶	2	2
POP direct	栈顶数据弹出到直接寻址字节	2	2
XCH A,Rn	寄存器和累加器交换	1	1
XCH A,direct	直接寻址字节与累加器交换	2	2
XCH A,@Ri	间址 RAM 与累加器交换	1	2
XCHD A,@Ri	间址 RAM 和累加器交换低半字节	1	2
位操作类指令			
CLR C	清进位位	1	1
CLR bit	清直接寻址位	2	2
SETB C	进位位置 1	1	1
SETB bit	直接寻址位置位	2	2
CPL C	进位位取反	1	1
CPL bit	直接寻址位取反	2	2
ANL C,bit	直接寻址位“与”到进位位	2	2
ANL C,/bit	直接寻址位的反码“与”到进位位	2	2
ORL C,bit	直接寻址位“或”到进位位	2	2
ORL C,/bit	直接寻址位的反码“或”到进位位	2	2
MOV C,bit	直接寻址位传送到进位位	2	2
MOV bit,C	进位位传送到直接寻址位	2	2
JC rel	若进位位为 1 则跳转	2	2/3
JNC rel	若进位位为零则跳转	2	2/3
JB bit,rel	若直接寻址位为 1 则跳转	3	3/4
JNB bit,rel	若直接寻址位为零则跳转	3	3/4
JBC bit,rel	若直接寻址位为 1 则跳转，并清除该位	3	3/4

C8051F326/7 全速 USB、16KB FLASH 微控制器

助记符	功能说明	字节数	时钟周期数
控制转移类指令			
ACALL addr11	绝对调用子程序	2	3
LCALL addr16	长调用子程序	3	4
RET	从子程序返回	1	5
RETI	从中断返回	1	5
AJMP addr11	绝对转移	2	3
LJMP addr16	长转移	3	4
SJMP rel	短转移（相对地址）	2	3
JMP @A+DPTR	相对 DPTR 的间接转移	1	3
JZ rel	累加器为 0 则转移	2	2/3
JNZ rel	累加器为非 0 则转移	2	2/3
CJNE A,direct,rel	比较直接寻址字节与累加器，不相等则转移	3	3/4
CJNE A,#data,rel	比较立即数与累加器，不相等则转移	3	3/4
CJNE Rn,#data,rel	比较立即数与寄存器，不相等则转移	3	3/4
CJNE @Ri,#data,rel	比较立即数与间接寻址 RAM，不相等则转移	3	4/5
DJNZ Rn,rel	寄存器减 1，不为零则转移	2	2/3
DJNZ direct,rel	直接寻址字节减 1，不为零则转移	3	3/4
NOP	空操作	1	1

寄存器、操作数和寻址方式说明：

Rn – 当前选择的寄存器区的寄存器 R0-R7。

@Ri – 通过寄存器 R0-R1 间接寻址的数据 RAM 地址。

rel – 相对于下一条指令第一个字节的 8 位有符号（2 的补码）偏移量。SJMP 和所有条件转移指令使用。

direct – 8 位内部数据存储器地址。可以是直接访问数据 RAM 地址（0x00-0x7F）或一个 SFR 地址（0x80-0xFF）。

#data – 8 位立即数

#data16 – 16 位立即数

bit – 数据 RAM 或 SFR 中的直接寻址位

addr11 – ACALL 或 AJMP 使用的 11 位目的地址。目的地址必须与下一条指令第一个字节处于同一个 2K 字节的程序存储器页。

addr16 – LCALL 或 LJMP 使用的 16 位目的地址。目的地址可以是 8K 程序存储器空间内的任何位置。

有一个未使用的操作码（0xA5），它执行与 NOP 指令相同的功能。

6.2 存储器组织

CIP-51 系统控制器的存储器组织与标准 8051 的存储器组织类似。有两个独立的存储器空间：程序存储器和数据存储器。程序和数据存储器共享同一个地址空间，但用不同的指令类型访问。CIP-51 的存储器组织如图 6.2 所示。

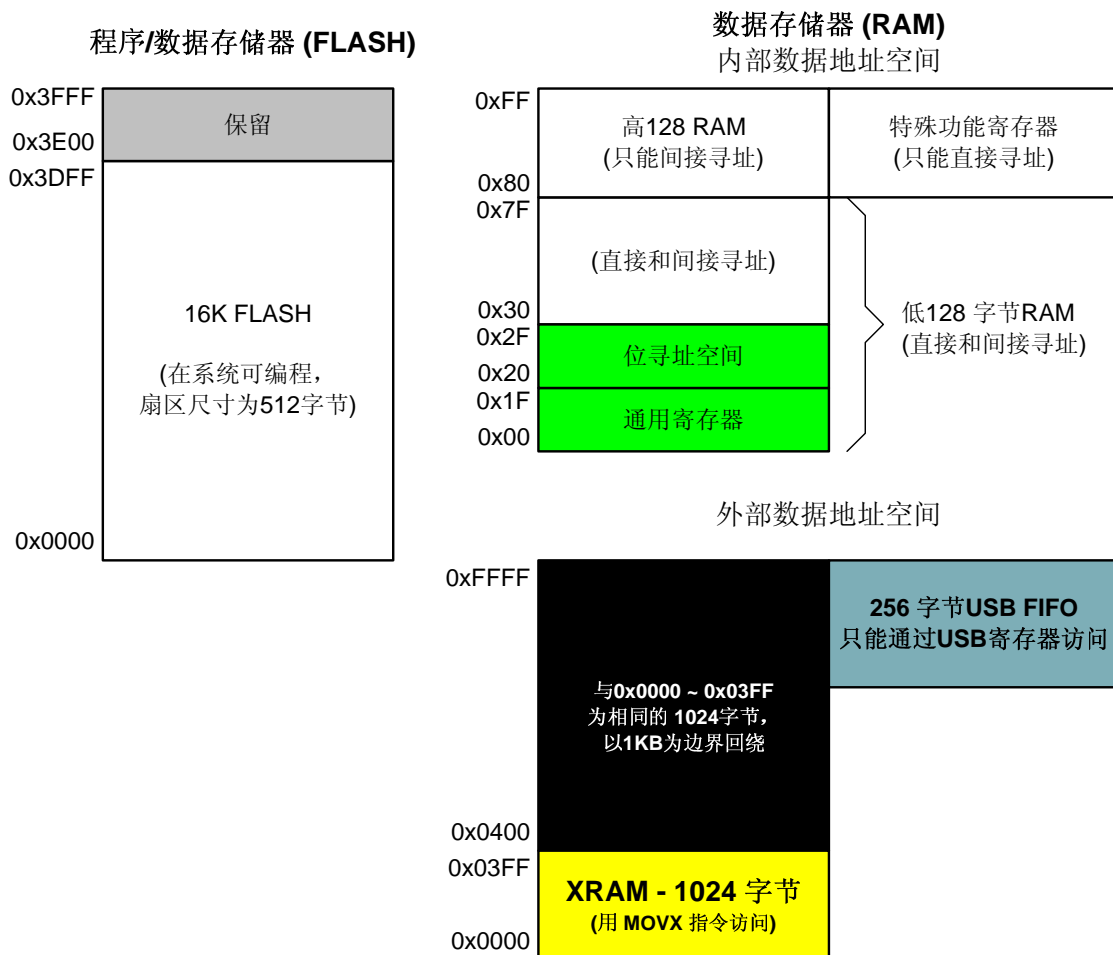


图 6.2 程序存储器结构图

6.2.1 程序存储器

CIP-51 有 64KB 的程序存储器空间。C8051F326/7 在这个程序存储器空间中实现了 16KB 的可在系统编程的 FLASH 存储器，组织在一个连续的存储块内 (0x0000 – 0x3FFF)。注意：0x3DFF 以上的地址被保留。

程序存储器通常被认为是只读的。但是 CIP-51 可以通过设置程序存储写允许位 (PSCTL.0) 用 MOVX 指令对程序存储器写入。这一特性为 CIP-51 提供了更新程序代码和将程序存储器空间用于非易失性数据存储的机制。更详细的信息见“8. FLASH 存储器”。

6.2.2 数据存储器

CIP-51 的数据存储器空间中有 256 字节的内部 RAM, 位于地址 0x00 到 0xFF 的地址空间。数据存储器中的低 128 字节用于通用寄存器和临时存储器。可以用直接或间接寻址方式访问数据存储器中的低 128 字节。从 0x00 到 0x1F 为 4 个通用寄存器区, 每个区有 8 个 8 位寄存器。接下来的 16 字节, 从地址 0x20 到 0x2F, 既可以按字节寻址又可以作为 128 个位地址用直接位寻址方式访问。

数据存储器中的高 128 字节只能用间接寻址访问。该存储区与特殊功能寄存器 (SFR) 占据相同的地址空间, 但物理上与 SFR 空间是分开的。当寻址高于 0x7F 的地址时, 指令所用的寻址方式决定了 CPU 是访问数据存储器的高 128 字节还是访问 SFR。使用直接寻址方式的指令将访问 SFR 空间, 间接寻址高于 0x7F 地址的指令将访问数据存储器的高 128 字节。图 6.2 给出了 CIP-51 数据存储器组织的示意图。

6.2.3 通用寄存器

数据存储器的低 32 字节, 从地址 0x00 到 0x1F, 可以作为 4 个通用寄存器区访问。每个区有 8 个 8 位寄存器, 称为 R0 - R7。在某一时刻只能选择一个寄存器区。程序状态字中的 RS0 (PSW.3) 和 RS1 (PSW.4) 位用于选择当前的寄存器区 (见图 6.4 中关于 PSW 的说明)。这允许在进入子程序或中断服务程序时进行快速现场切换。间接寻址方式使用 R0 和 R1 作为间接寻址寄存器。

6.2.4 位寻址空间

除了直接访问按字节组织的数据存储器外, 从 0x20 到 0x2F 的 16 个数据存储器单元还可以作为 128 个独立寻址位访问。每个位有一个位地址, 从 0x00 到 0x7F。位于地址 0x20 的数据字节的位 0 具有位地址 0x00, 位于 0x20 的数据字节的位 7 具有位地址 0x07。位于 0x2F 的数据字节的位 7 具有位地址 0x7F。由所用指令的类型来区分是位寻址还是字节寻址。

MCS-51TM 汇编语言允许用 XX.B 的形式替代位地址, XX 为字节地址, B 为寻址位在字节中的位置。例如, 指令:

```
MOV    C,  22h.3
```

将 0x13 中的布尔值 (字节地址 0x22 中的位 3) 传送到进位标志。

6.2.5 堆栈

程序的堆栈可以位于 256 字节数据存储器中的任何位置。堆栈区域用堆栈指针 (SP, 0x81) SFR 指定。SP 指向最后使用的位置。下一个压入堆栈的数据将被存放在 SP+1, 然后 SP 加 1。复位后堆栈指针被初始化为地址 0x07, 因此第一个被压入堆栈的数据将被存放在地址 0x08, 这也是寄存器区 1 的第一个寄存器 (R0)。如果使用不止一个寄存器区, SP 应被初始化为数据存储器中不用于数据存储的位置。堆栈深度最大可达 256 字节。

C8051F326/7 全速 USB、16KB FLASH 微控制器

6.2.6 特殊功能寄存器

从 0x80 到 0xFF 的直接寻址存储器空间为特殊功能寄存器（SFR）。SFR 提供对 CIP-51 的资源和外设的控制及 CIP-51 与这些资源和外设之间的数据交换。CIP-51 具有标准 8051 中的全部 SFR，还增加了一些用于配置和访问专有子系统的 SFR。这就允许在保证与 MCS-51™ 指令集兼容的前提下增加新的功能。表 6.2 列出了 CIP-51 系统控制器中的全部 SFR。

任何时刻用直接寻址方式访问 0x80~0xFF 的存储器空间将访问特殊功能寄存器（SFR）。地址以 0x0 或 0x8 结尾的 SFR（例如 P0、TCON、SCON0、IE 等）既可以按字节寻址也可以按位寻址，所有其它 SFR 只能按字节寻址。SFR 空间中未使用的地址保留为将来使用，访问这些地址会产生不确定的结果，应予避免。有关每个寄存器的详细说明请参见本数据表的相关部分（表 6.3 中已标明）。

表 6.2 特殊功能寄存器（SFR）存储器映象

F8							VDM0CN
F0	B					EIP1	EIP2
E8							RSTSRC
E0	ACC		GPIOCN	OSCLCN		EIE1	EIE2
D8							
D0	PSW						USB0XCN
C8		REG0CN					
C0							
B8	IP	CLKMUL					
B0	P3		OSCICN	OSCICL		FLSCL	FLKEY
A8	IE	CLKSEL	EMI0CN				
A0	P2			P0MDOUT		P2MDOUT	P3MDOUT
98	SCON0	SBUF0	SMOD0				
90	P1	SBCON0		SBRLLO	SBRLHO	USB0ADR	USB0DAT
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON
80	P0	SP	DPL	DPH			PCON
	0(8) 可位寻址	1(9)	2(A)	3(B)	4(C)	5(D)	6(E) 7(F)

C8051F326/7 全速 USB、16KB FLASH 微控制器

表 6.3 特殊功能寄存器

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

寄存器	地址	说 明	页码
ACC	0xE0	累加器	
B	0xF0	B 寄存器	
CKCON	0x8E	时钟控制寄存器	
CLKMUL	0xB9	时钟乘法器寄存器	
CLKSEL	0xA9	时钟选择寄存器	
DPH	0x83	数据指针高字节	
DPL	0x82	数据指针低字节	
EIE1	0xE6	扩展中断允许寄存器 1	
EIE2	0xE7	扩展中断允许寄存器 2	
EIP1	0xF6	扩展中断优先级寄存器 1	
EIP2	0xF7	扩展中断优先级寄存器 2	
EMI0CN	0xAA	外部存储器接口控制寄存器	
FLKEY	0xB7	FLASH 锁定和关键码寄存器	
FLSCL	0xB6	FLASH 存储器时序预分频器	
GPIOCN	0xE2	全局 I/O 控制寄存器	
IE	0xA8	中断允许寄存器	
IP	0xB8	中断优先级寄存器	
OSICL	0xB3	内部振荡器校准寄存器	
OSICN	0xB2	内部振荡器控制寄存器	
OSCLCN	0xE3	低频内部振荡器控制寄存器	
P0	0x80	端口 0 锁存器	
P0MDOUT	0xA4	端口 0 输出方式配置寄存器	
P2	0xA0	端口 2 锁存器	
P2MDOUT	0xA6	端口 2 输出方式配置寄存器	
P3	0xB0	端口 3 锁存器	
P3MDOUT	0xA7	端口 3 输出方式配置寄存器	
PCON	0x87	电源控制寄存器	
PSCTL	0x8F	程序存储读/写控制寄存器	
PSW	0xD0	程序状态字	
RSTSRC	0xEF	复位源寄存器	
SBUF0	0x99	UART0 数据缓冲器	
SBCON0	0x91	波特率发生器 0 控制寄存器	
SBRLH0	0x94	波特率发生器 0 重载值高字节寄存器	
SBRLLO	0x93	波特率发生器 0 重载值低字节寄存器	
SCON0	0x98	UART0 控制寄存器	
SMOD0	0x9A	UART0 方式寄存器	

C8051F326/7 全速 USB、16KB FLASH 微控制器

表 6.3 特殊功能寄存器（续）

SFR 以字母顺序排列，所有未定义的 SFR 位置保留。

寄存器	地址	说 明	页码
SP	0x81	堆栈指针	
TCON	0x88	计数器/定时器控制寄存器	
TH0	0x8C	计数器/定时器 0 高字节	
TH1	0x8D	计数器/定时器 1 高字节	
TL0	0x8A	计数器/定时器 0 低字节	
TL1	0x8B	计数器/定时器 1 低字节	
TMOD	0x89	计数器/定时器方式寄存器	
USB0ADR	0x96	USB0 间接地址寄存器	
USB0DAT	0x97	USB0 数据寄存器	
USB0XCN	0xD7	USB0 收发器控制寄存器	
VDM0CN	0xFF	VDD 监视器控制寄存器	

C8051F326/7 全速 USB、16KB FLASH 微控制器

6.2.7 寄存器说明

下面对与 CIP-51 系统控制器操作有关的 SFR 加以说明。保留位不应被置为逻辑‘1’。将来的产品版本可能会使用这些位实现新功能，在这种情况下各位的复位值将是逻辑‘0’以选择缺省状态。有关其它 SFR 的详细说明见本数据表中与它们对应的系统功能相关的章节。

SFR 定义 6.1 DPL: 数据指针低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x82
位 7-0: DPL: 数据指针低字节								
DPL 为 16 位数据指针 (DPTR) 的低字节。DPTR 用于访问间接寻址的存储器。								

SFR 定义 6.2 DPH: 数据指针高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x83
位 7-0: DPH: 数据指针高字节								
DPH 为 16 位数据指针 (DPTR) 的高字节。DPTR 用于访问间接寻址的存储器。								

SFR 定义 6.3 SP: 堆栈指针

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000111
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x81
位 7-0: SP: 堆栈指针								
堆栈指针保持栈顶位置。在每次执行 PUSH 操作前，堆栈指针加 1。SP 寄存器复位后的默认值为 0x07。								

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 6.4 PSW：程序状态字

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CY	AC	F0	RS1	RS0	OV	F1	PARITY	00000000
位7	位6	位5	位4	位3	位2	位1	位0 (可位寻址)	SFR地址: 0xD0

位 7: CY：进位标志。
当最后一次算术操作产生进位（加法）或借位（减法）时，该位置 1。其它算术操作将其清 0。

位 6: AC：辅助进位标志。
当最后一次算术操作向高半字节有进位（加法）或借位（减法）时，该位置 1。其它算术操作将其清 0。

位 5: F0：用户标志 0。
这是一个可位寻址、受软件控制的通用标志位。

位 4-3: RS1-RS0：寄存器区选择。
该两位在寄存器访问时用于选择寄存器区。

RS1	RS0	寄存器区	地址
0	0	0	0x00-0x07
0	1	1	0x08-0x0F
1	0	2	0x10-0x17
1	1	3	0x18-0x1F

位 2: OV：溢出标志。
该位在下列情况下被置 1：

- ADD、ADDC 或 SUBB 指令引起符号位变化溢出。
- MUL 指令引起溢出（结果大于 255）。
- DIV 指令的除数为 0。

ADD、ADDC、SUBB、MUL 和 DIV 指令的其它情况使该位清 0。

位 1: F1：用户标志 1。
这是一个可位寻址、受软件控制的通用标志位。

位 0: PARITY：奇偶标志。
若累加器中 8 个位的和为奇数时该位置 1，为偶数时清 0。

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 6.5 ACC: 累加器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: (可位寻址) 0xE0
位 7-0: ACC: 累加器 该寄存器为算术操作的累加器。								

SFR 定义 6.6 B: B 寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: (可位寻址) 0xF0
位 7-0: B: B 寄存器 该寄存器为某些算术操作的第二累加器。								

6.3 中断系统

CIP-51 包含一个扩展的中断系统，支持 8 个中断源，每个中断源有两个优先级。中断源在片内外设与外部输入引脚之间的分配随器件的不同而变化。除 USB0 外，每个中断源都在一个 SFR 中有一个或多个中断标志。USB0 中断标志位于 USB 寄存器中。有关 USB 中断的更详细信息见 12.8 节。当一个外设或外部源满足有效的中断条件时，相应的中断标志被置为逻辑‘1’。

如果一个中断源被允许，则在中断标志被置位时将产生一个中断。一旦当前指令执行完，CPU 产生一个 LCALL 到预定地址，开始执行中断服务程序（ISR）。每个 ISR 必须以 RETI 指令结束，使程序回到中断前执行的那条指令的下一条指令。如果中断未被允许，中断标志将被硬件忽略，程序继续正常执行。中断标志置‘1’与否不受中断允许/禁止状态的影响。

每个中断源都可以用一个 SFR（IE – EIE2）中的相关中断允许位来允许或禁止，但是必须首先将 EA 位（IE.7）置‘1’，以保证每个单独的中断允许位有效。不管每个中断允许位的设置如何，清‘0’EA 位将禁止所有中断。

注意：任何使 EA 位清 0 的指令的后面都应跟随一条具有两个或更多操作码字节的指令。例如：

```
// 用 C 语言编程
EA = 0; // 清‘0’EA 位
EA = 0; // 跟随一条两字节操作码的指令

; 用汇编语言编程
CLR    EA    ; 清‘0’EA 位
CLR    EA    ; 跟随一条两字节操作码的指令
```

如果一个中断在“CLR EA”指令（或任何使 EA 清‘0’的指令）的执行阶段有效，并且该指令的后面是一条单周期的指令，则该中断可能被响应，但在中断服务程序中读 EA 位将返回‘0’值。如果“CLR EA”指令后面是一条多周期的指令，则中断不会被响应。

某些中断标志在 CPU 进入 ISR 时被自动清除，但大多数中断标志不是由硬件清除的，必须在 ISR 返回前用软件清除。如果一个中断标志在 CPU 执行完中断返回（RETI）指令后仍然保持置位状态，则会立即产生一个新的中断请求，CPU 将在执行完下一条指令后再次进入该 ISR。

6.3.1 MCU 中断源和中断向量

MCU 支持 8 个中断源。软件可以通过将任何一个中断标志设置为逻辑‘1’来模拟一个中断。如果中断标志被允许，系统将产生一个中断请求，CPU 将转向与该中断标志对应的 ISR 地址。表 6.4 给出了 MCU 中断源、对应的向量地址、优先级和控制位一览表。关于外设有效中断条件和中断标志位工作状态方面的详细信息，请见与特定外设相关的章节。

6.3.2 外部中断

/INT0 低电平有效外部中断被映射到 P0.0（或 P0.2，当 TMOD.3 为 1 时），并可被配置为边沿触发或电平触发。IT0 位（TCON.0，见图 14.1）用于选择电平或边沿触发。当全局端口

C8051F326/7 全速 USB、16KB FLASH 微控制器

I/O 输入被使能时，/INT0 会监视输入引脚的电压，一旦检测到下降沿或低电平信号，CPU 就会转去执行/INT0 中断服务程序。

/INT1 中断源为两种事件提供中断，取决于 GATE1 (TMOD.7) 的逻辑电平。当 GATE1 为逻辑 1 时，每两个低频内部振荡器周期产生一次中断，这就允许 CPU 以 40KHz 的速率执行/INT1 中断服务程序。如果 GATE1 为逻辑 0，则在内部振荡器从挂起状态恢复时会产生中断。

/INT0 和/INT1 的中断标志在复位时被置 1。如果使用/INT0 或/INT1 中断，则应在中断被使能之前清除响应的中断标志位，以避免产生一次意外中断。/INT0 和/INT1 中断标志位在 TCON 寄存器中。

6.3.3 中断优先级

每个中断源都可以被独立地编程为两个优先级中的一个：低优先级或高优先级。一个低优先级的中断服务程序可以被高优先级的中断所中断，但高优先级的中断不能被中断。每个中断在 SFR (IP 或 EIP1、EIP2) 中都有一个配置其优先级的中断优先级设置位，缺省值为低优先级。如果两个中断同时发生，具有高优先级的中断先得到服务。如果这两个中断的优先级相同，则由固定的优先级顺序（见表 6.4）决定哪一个中断先得到服务。

6.3.4 中断响应时间

中断响应时间取决于中断发生时 CPU 的状态。中断系统在每个系统时钟周期对中断标志采样并对优先级译码。最快的响应时间为 5 个系统时钟周期：一个周期用于检测中断，4 个周期完成对 ISR 的长调用 (LCALL)。如果中断标志有效时 CPU 正在执行 RETI 指令，则需要再执行一条指令才能进入中断服务程序。因此，最长的中断响应时间（没有其它中断正被服务或新中断具有较高优先级）发生在 CPU 正在执行 RETI 指令，而下一条指令是 DIV 的情况。在这种情况下，响应时间为 18 个系统时钟周期：1 个时钟周期检测中断，5 个时钟周期执行 RETI，8 个时钟周期完成 DIV 指令，4 个时钟周期执行对 ISR 的长调用 (LCALL)。如果 CPU 正在执行一个具有相同或更高优先级的中断的 ISR，则新中断要等到当前 ISR 执行完（包括 RETI 和下一条指令）才能得到服务。

在 FLASH 写/擦除操作期间 CPU 暂停执行指令。对于在 CPU 暂停执行指令期间发生的中断，中断服务响应时间将延长。这种情况下的中断延迟时间由标准中断服务响应过程（如前所述）和 CPU 暂停执行指令的时间决定。

C8051F326/7 全速 USB、16KB FLASH 微控制器

表 6.4 中断一览表

中断源	中断向量	优先级	中断标志	位寻址	硬件清除	中断允许	优先级控制
复位	0x0000	最高	无	N/A	N/A	始终允许	总是最高
外部中断 0 (/INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
定时器 0 溢出	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
外部中断 1 (/INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
定时器 1 溢出	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y	N	ES0 (IE.4)	PS0 (IP.4)
USB0	0x0043	8	特殊*	N	N	EUSB0 (EIE1.1)	PUSB0 (EIP1.1)
VBUS 电平	0x007B	15	N/A	N/A	N/A	EVBUS (EIE2.0)	PVBUS (EIP2.0)
*注：有关 USB 中断的详细信息见 12.8 节。							

6.3.5 中断寄存器说明

下面介绍用于允许中断源和设置中断优先级的特殊功能寄存器。关于外设有效中断条件和中断标志位工作状态方面的详细信息，请见与特定片内外设相关的章节。

SFR 定义 6.7 IE：中断允许寄存器

R/W	R	R	R/W	R/W	R/W	R/W	R/W	复位值
EA	—	—	ES0	ET1	EX1	ET0	EX0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xA8 (可位寻址)
<p>位 7: EA: 允许所有中断。 该位允许 / 禁止所有中断。它超越所有的单个中断屏蔽设置。 0: 禁止所有中断源。 1: 开放中断。每个中断由它对应的中断屏蔽设置决定。</p> <p>位 6-5: 未用。读 = 00b, 写 = 忽略。</p> <p>位 4: ES0: UART0 中断允许位。 该位设置 UART0 的中断屏蔽。 0: 禁止 UART0 中断。 1: 允许 UART0 中断。</p> <p>位 3: ET1: 定时器 1 中断允许位。 该位用于设置定时器 1 的中断屏蔽。 0: 禁止定时器 1 中断。 1: 允许 TF1 标志位的中断请求。</p> <p>位 2: EX1: 外部中断 1 允许位。 该位用于设置外部中断 1 的中断屏蔽。 0: 禁止外部中断 1。 1: 允许/INT1 引脚的中断请求</p> <p>位 1: ET0: 定时器 0 中断允许位。 该位用于设置定时器 0 的中断屏蔽。 0: 禁止定时器 0 中断。 1: 允许 TF0 标志位的中断请求。</p> <p>位 0: EX0: 外部中断 0 允许位。 该位用于设置外部中断 0 的中断屏蔽。 0: 禁止外部中断 0。 1: 允许/INT0 引脚的中断请求</p>								

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 6.8 IP: 中断优先级寄存器

R	R	R	R/W	R/W	R/W	R/W	R/W	复位值
—	—	—	PS0	PT1	PX1	PT0	PX0	10000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址:
							(可位寻址)	0xB8
位 7-5: 未用。读 = 100b, 写 = 忽略。								
位 4: PS0: UART0 中断优先级控制。 该位设置 UART0 中断的优先级。 0: UART0 为低优先级。 1: UART1 为高优先级。								
位 3: PT1: 定时器 1 中断优先级控制 该位设置定时器 1 中断的优先级。 0: 定时器 1 为低优先级。 1: 定时器 1 为高优先级。								
位 2: PX1: 外部中断 1 优先级控制 该位设置外部中断 1 的优先级。 0: 外部中断 1 为低优先级。 1: 外部中断 1 为高优先级。								
位 1: PT0: 定时器 0 中断优先级控制 该位设置定时器 0 中断的优先级。 0: 定时器 0 为低优先级。 1: 定时器 0 为高优先级。								
位 0: PX0: 外部中断 0 优先级控制 该位设置外部中断 0 的优先级。 0: 外部中断 0 为低优先级。 1: 外部中断 0 为高优先级。								

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 6.9 EIE1: 扩展中断允许 1

R	R	R	R	R	R	R/W	R	复位值
—	—	—	—	—	—	EUSB0	—	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xE6
位 7-2: 未用。读 = 000000b, 写 = 忽略。								
位 1: EUSB0: USB0 中断允许位 该位设置 USB0 的中断屏蔽。 0: 禁止 USB0 中断。 1: 允许 USB0 的中断请求。								
位 0: 未用。读 = 0b, 写 = 忽略。								

SFR 定义 6.10 EIP1: 扩展中断优先级 1

R	R	R	R	R	R	R/W	R	复位值
—	—	—	—	—	—	PUSB0	—	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xF6
位 7-2: 未用。读 = 000000b, 写 = 忽略。								
位 1: PUSB0: USB0 中断优先级控制 该位设置 USB0 中断的优先级。 0: USB0 中断为低优先级。 1: USB0 中断为高优先级。								
位 0: 未用。读 = 0b, 写 = 忽略。								

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 6.11 EIE2: 扩展中断允许 2

R	R	R	R	R	R	R	R/W	复位值
—	—	—	—	—	—	—	EVBUS	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xE7
位 7-1: 未使用。读 = 0000000b, 写 = 忽略。								
位 0: EVBUS: VBUS 电平中断允许位								
该位设置 VBUS 的中断屏蔽。								
0: 禁止所有 VBUS 中断。								
1: 允许 VBUS 电平检测电路的中断请求。								

SFR 定义 6.12 EIP2: 扩展中断优先级 2

R	R	R	R	R	R	R	R/W	复位值
—	—	—	—	—	—	—	PVBUS	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xF7
位 7-1: 未使用。读 = 0000000b, 写 = 忽略。								
位 0: PVBUS: VBUS 电平中断优先级控制								
该位设置 VBUS 中断的优先级。								
0: VBUS 中断为低优先级。								
1: VBUS 中断为高优先级。								

6.4 电源管理方式

CIP-51 有两种可软件编程的电源管理方式：空闲和停机。在空闲方式，CPU 停止运行，而外设和时钟处于活动状态。在停机方式，CPU 停止运行，所有的中断都处于非活动状态，内部振荡器被停止（模拟外设保持在所选择的状态；外部振荡器不受影响）。由于在空闲方式下时钟仍然运行，所以功耗与进入空闲方式之前的系统时钟频率和处于活动状态的外设数目有关。停机方式消耗最少的功率。图 6.13 对用于控制 CIP-51 电源管理方式的电源控制寄存器作出了说明。

虽然 CIP-51 具有空闲和停机方式（与任何标准 8051 结构一样），但通过管理系统时钟和每个外设，可以使整个 MCU 的功耗最小。每个模拟外设在不用时都可以被禁止，使其进入低功耗方式。像定时器、串行总线这样的数字外设在不使用时消耗很少的功率。关闭振荡器可以大大降低消耗功率，但需要复位来重新启动 MCU。

内部振荡器可以被置于挂起方式（见“10. 振荡器”）。在挂起方式，内部振荡器停止运行，直到检测到一个非空闲 USB 事件或 VBUS 输入信号与 VBPOL 位（位于寄存器 REG0CN）所选择的极性匹配。

6.4.1 空闲方式

将空闲方式选择位（PCON.0）置 1 导致 CIP-51 停止 CPU 运行并进入空闲方式，在执行完对该位置 1 的指令后 MCU 立即进入空闲方式。所有内部寄存器和存储器都保持原来的数据不变。所有模拟和数字外设空闲方式期间都可以保持活动状态。

有被允许的中断发生或复位有效将结束空闲方式。当有一个被允许的中断发生时，空闲方式选择位（PCON.0）被清 0，CPU 将继续工作。该中断将得到服务，中断返回（RETI）后将开始执行设置空闲方式选择位的那条指令的下一条指令。如果空闲方式因一个内部或外部复位而结束，则 CIP-51 进行正常的复位过程并从地址 0x0000 开始执行程序。

6.4.2 停机方式

将停机方式选择位（PCON.1）置 1 导致 CIP-51 进入停机方式，在执行完对该位置 1 的指令后 MCU 立即进入停机方式。在停机方式，内部振荡器、CPU 和所有的数字外设都停止工作，但外部振荡器电路的状态不受影响。在进入停机方式之前，每个模拟外设（包括外部振荡器电路）都可以被单独关断。只有内部或外部复位能结束停机方式。复位时，CIP-51 进行正常的复位过程并从地址 0x0000 开始执行程序。

如果被使能，时钟丢失检测器将产生一个内部复位，从而结束停机方式。如果想要使 CPU 的休眠时间长于 100 微秒的 MCD 超时时间，则应禁止时钟丢失检测器。

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 6.13 PCON: 电源控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
GF5	GF4	GF3	GF2	GF1	GF0	STOP	IDLE	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x87
<p>位 7-2: GF5-GF0: 通用标志位 5-0。 这些位是用作软件控制的通用标志位。</p> <p>位 1: STOP: 停机方式选择。 将该位置 ‘1’ 使 CIP-51 进入停机方式。该位读出值总是为 0。 1: 进入掉电方式（内部振荡器停止运行）</p> <p>位 0: IDLE: 空闲方式选择。 将该位置 ‘1’ 使 CIP-51 进入空闲方式。该位读出值总是为 0。 1: CPU 进入空闲方式。（关闭供给 CPU 的时钟信号，但定时器、中断、串口和模拟外设保持活动状态。）</p>								

7. 复位源

复位电路允许很容易地将控制器置于一个预定的缺省状态。在进入复位状态时，将发生以下过程：

- CIP-51 停止程序执行
- 特殊功能寄存器（SFR）被初始化为所定义的复位值
- 外部端口引脚被置于一个已知状态
- 中断和定时器被禁止。

所有的 SFR 都被初始化为预定值，SFR 中各位的复位值在 SFR 的详细说明中定义。在复位期间内部数据存储器的内容不发生改变，复位前存储的数据保持不变。但由于堆栈指针 SFR 被复位，堆栈实际上已丢失，尽管堆栈中的数据未发生变化。

端口 I/O 锁存器的复位值为 0xFF（全部为逻辑‘1’），处于漏极开路方式。在复位期间和复位之后弱上拉被使能。对于 VDD 监视器和上电复位，/RST 引脚被驱动为低电平，直到器件退出复位状态。

在退出复位状态时，程序计数器（PC）被复位，MCU 使用内部振荡器作为默认的系统时钟。有关选择和配置系统时钟源的详细说明见“10. 振荡器”。程序从地址 0x0000 开始执行。

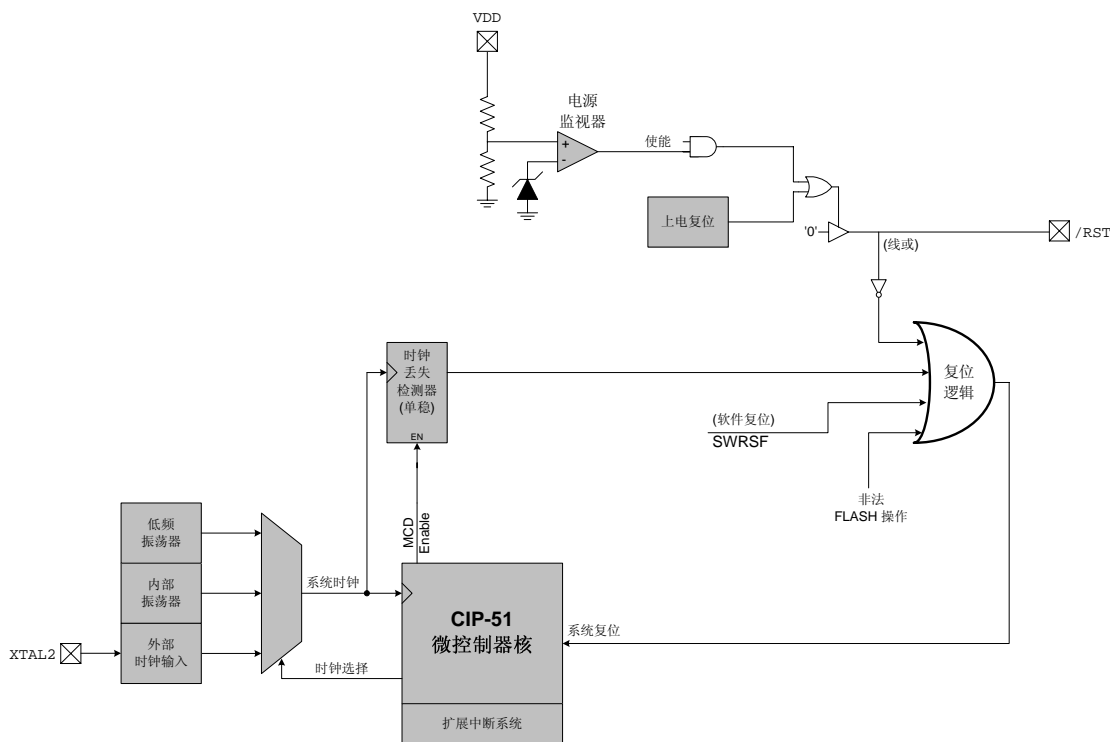


图 7.1 复位源框图

在上电期间，器件保持在复位状态，/RST引脚被驱动到低电平，直到VDD上升到超过 V_{RST} 电平。从复位开始到退出复位状态要经过一个上电复位延时（ $T_{PORDelay}$ ）；该延时通常小于0.3ms。图 7.2 给出了上电复位和VDD监视器复位时序。

软件可以通过向寄存器 RSTSRC 中的 PINRSF 位写 ‘1’ 来强制产生一次上电复位。



7.2 掉电复位和 VDD 监视器

当发生掉电或因电源波动导致VDD降到 V_{RST} 以下时，电源监视器将/RST引脚驱动为低电平并使CIP-51 保持复位状态（见图 7.2）。当VDD又回到高于 V_{RST} 的电平时，CIP-51 将退出复位状态。注意，尽管内部数据存储器的内容可能没有因掉电复位而发生改变，但无法确定VDD是否降到了数据保持所要求的最低电平以下。如果PORSF标志的读出值为 1，则内部RAM的数据可能不再有效。在上电复位后VDD监视器被使能，但它的状态（使能/禁止）不受任何其它复位源的影响。例如，在VDD监视器被使能后执行一次软件复位，复位后VDD监视器仍然为使能状态。

注意：在选择 VDD 监视器作为复位源之前，必须使能 VDD 监视器。在 VDD 监视器被使能或稳定之前选其为复位源可能导致系统复位。将 VDD 监视器配置为复位源的步骤如下：

1. 使能 VDD 监视器（VDM0CN.7 = 1）；
2. 等待 VDD 监视器稳定（见表 7.1 的 VDD 监视器启动时间）；
3. 选择 VDD 监视器作为复位源（RSTSR.1 = 1）。

图 7.2 给出了 VDD 监视器的时序。表 7.1 给出了 VDD 监视器的电气特性。

SFR 定义 7.1 VDM0CN: VDD 监视器控制寄存器

R/W	R	R	R	R	R	R	R	复位值
VDMEN	VDDSTAT	保留	保留	保留	保留	保留	保留	可变
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xFF
<div> <div>位 7:</div> <div> VDMEN: VDD 监视器使能位。 该位控制 VDD 监视器电源的通断。VDD 监视器在被选择为复位源之前不能产生系统复位。在被选择为复位源之前，VDD 监视器必须稳定。在 VDD 监视器稳定之前选其为复位源可能导致系统复位。VDD 监视器的最小启动时间见表 7.1。在上电复位后 VDD 监视器被使能。 0: 禁止 VDD 监视器。 1: 使能 VDD 监视器。 </div> </div> <div> <div>位 6:</div> <div> VDDSTAT: VDD 状态。 该位指示当前电源状态（VDD 监视器输出）。 0: VDD 等于或低于 VDD 监视器阈值。 1: VDD 高于 VDD 监视器阈值 </div> </div> <div> <div>位 5-0:</div> <div>保留。读 = 可变，写 = 忽略。</div> </div>								

7.3 外部复位

外部/RST 引脚提供了使用外部电路强制 MCU 进入复位状态的手段。在/RST 引脚上加一个低电平有效信号将导致 MCU 进入复位状态。尽管在内部有弱上拉，但最好能提供一个外部上拉和/或对/RST 引脚去耦以防止强噪声引起复位。从外部复位状态退出后，PINRSF 标志（RSTSRC.0）被置‘1’。

7.4 时钟丢失检测器复位

时钟丢失检测器（MCD）实际上是由系统时钟触发的单稳态电路。如果系统时钟两个连续上升沿的时间间隔大于 100 微秒，单稳态电路将超时并产生复位。在发生时钟丢失检测器复位后，MCDRSF 标志（RSTSRC.2）的读出值为 1，表示本次复位源为 MCD；否则该位读出值为 0。向 MCDRSF 位写 1 将使能时钟丢失检测器；写 0 将禁止时钟丢失检测器。/RST 引脚的状态不受该复位的影响。

7.5 FLASH 错误复位

如果 FLASH 读/写/擦除操作的地址或程序读地址为非法地址，将发生系统复位。下述的任何一种情况都会导致 FLASH 操作错误：

- FLASH 写或擦除地址超出了用户代码空间。这种情况发生在 PSWE 被置‘1’，并且 MOVX 操作的地址大于 0x3DFF。
- FLASH 读地址超出了用户代码空间，即 MOVC 操作的地址大于 0x3DFF。
- 程序读超出了用户代码地址空间。这种情况发生在用户代码试图转移到大于 0x3DFF 的地址时。
- 当 FLASH 读、写或擦除被安全设置禁止时（见 8.3 安全选项）。

在发生 FLASH 错误复位后，FERROR 位（RSTSRC.6）被置位。/RST 引脚的状态不受该复位的影响。

7.6 软件复位

软件可以通过向 SWRSF 位（RSTSRC.4）写‘1’强制产生一次系统复位。在发生软件强制复位后，SWRSF 位的读出值为‘1’。/RST 引脚的状态不受该复位的影响。

7.7 USB 复位

向寄存器 RSTSRC 中的 USBRSF 位写‘1’将选择 USB0 为复位源。在 USB0 被选择为复位源的情况下，下面的任何事件都会导致产生系统复位。

1. 在 USB 网络上检测到 RESET 命令。USB 功能控制器（USB0）必须被使能为检测 RESET 命令。有关 USB 功能控制器的详细信息见“12. 通用串行总线控制器（USB0）”。

2. VBUS 引脚的电压与 VBPOL 位（位于寄存器 REG0CN）选择的极性匹配。有关 VBUS 检测电路的详细信息见“5. 稳压器（REG0）”。

发生 USB 复位后，USBRSF 位的读出值为 1。/RST 引脚的状态不受该复位的影响。

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 7.2 RSTSRC: 复位源寄存器

R/W	R	R	R/W	R	R/W	R/W	R	复位值
USBRSF	FERROR	—	SWRSF	—	MCDRSF	PORSF	PINRSF	可变
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xEF
位 7:	USBRSF: USB 复位标志							
	0: 读: 最后一次复位不是来自 USB; 写: 禁止 USB 复位。							
	1: 读: 最后一次复位是来自 USB; 写: 允许 USB 复位。							
位 6:	FERROR: FLASH 错误标志。							
	0: 最后一次复位不是来自 FLASH 读/写/擦除错误。							
	1: 最后一次复位是由于 FLASH 读/写/擦除错误。							
位 5:	未使用。读 = 0b, 写 = 忽略。							
位 4:	SWRSF: 软件强制复位和标志							
	0: 读: 最后一次复位不是来自写 SWRSF 位。写: 无作用							
	1: 读: 最后一次复位来自写 SWRSF 位。写: 强制产生一次系统复位。							
位 3:	未使用。读 = 0b, 写 = 忽略。							
位 2:	MCDRSF: 时钟丢失检测器标志							
	0: 读: 最后一次复位不是来自时钟丢失检测器超时。写: 禁止时钟丢失检测器。							
	1: 读: 最后一次复位来自时钟丢失检测器超时。写: 使能时钟丢失检测器; 检测到时钟丢失条件时触发复位。							
位 1:	PORSF: 上电复位/VDD 监视器复位标志							
	该位在上电复位后被置位。对该位写入可以使能/禁止 VDD 监视器作为复位源。 注意: 在 VDD 监视器被使能和稳定之前向该位写 1 可能导致系统复位。 见寄存器 VDM0CN(图 7.1)。							
	0: 读: 最后一次复位不是上电复位或 VDD 监视器复位。写: 禁止 VDD 监视器为复位源。							
	1: 读: 最后一次复位是上电或 VDD 监视器复位, 所有其它复位标志是不确定的。写: 使能 VDD 监视器为复位源。							
位 0:	PINRSF: 硬件引脚复位标志							
	0: 最后一次复位不是来自/RST 引脚。							
	1: 最后一次复位来自/RST 引脚。							
注: 不要使用读-修改-写指令操作该寄存器。								

C8051F326/7 全速 USB、16KB FLASH 微控制器

表 7.1 复位源电气特性

-40℃到+85℃（除非特别说明）

参 数	条 件	最小值	典型值	最大值	单位
F326 /RST 输出电压	$I_{OL} = 8.5 \text{ mA}, V_{IO} = 2.7 \sim 3.6 \text{ V}$ $I_{OL} = \text{TBD}, V_{IO} = 2.0 \text{ V}$	—	—	0.6 TBD	V
F327 /RST 输出电压	$I_{OL} = \text{TBD}, V_{IO} = 2.7 \sim 3.6 \text{ V}$	—	—	TBD	V
/RST 输入高电平*		$0.7 \times V_{IO}$	—	—	V
/RST 输入低电平*		—	—	$0.3 \times V_{IO}$	V
F326 /RST 拉电流		10	26	40	μA
F327 /RST 拉电流		—	26	40	μA
VDD监视器门限(V_{RST})		2.40	2.55	2.70	V
时钟丢失检测器超时	从最后一个系统时钟上升沿到开始复位	100	240	500	μs
复位时间延迟	从复位源释放到开始执行位于 0x0000 地址的代码之间的延时	5.0	—	—	μs
产生系统复位的最小 /RST 低电平时间		15	—	—	μs
VDD 监视器启动时间		100	—	—	μs
VDD 监视器电源电流		—	45	TBD	μA
*注：对于 F327， $V_{IO} = V_{DD}$ 。					

8. FLASH 存储器

C8051F326/7 内部有可重复编程的 FLASH 存储器，用于程序代码和非易失性数据存储。可以通过 C2 接口或由软件使用 MOVX 指令对 FLASH 存储器进行在系统编程，每次一个字节。一个 FLASH 位一旦被清 ‘0’，必须经过擦除才能再回到 ‘1’ 状态。在进行重新编程之前，一般要将数据字节擦除（置为 0xFF）。为了保证操作正确，写和擦除操作由硬件自动定时，不需要进行数据查询来判断写/擦除操作何时结束。在 FLASH 写/擦除操作期间，程序暂停执行。表 8.1 给出了 FLASH 存储器的电气特性。

8.1 FLASH 存储器编程

对 FLASH 存储器编程的最简单的方法是使用由 Silicon Laboratories 或第三方供应商提供的编程工具，通过 C2 接口编程，这是对未被初始化过的器件的唯一编程方法。有关对 FLASH 程序存储器编程的 C2 命令的详细信息见“15. C2 接口”。

为了保证 FLASH 内容的正确性，强烈建议在使用用户软件对 FLASH 存储器进行写和/或擦除操作的系统中使能片内 VDD 监视器。

8.1.1 FLASH 锁定和关键码功能

从用户软件写和擦除 FLASH 受 FLASH 锁定和关键码功能的保护。在进行 FLASH 操作之前，必须按顺序向 FLASH 锁定和关键码寄存器（FLKEY）写入正确的关键码。关键码为：0xA5，0xF1。写关键码的时序并不重要，但必须按顺序写。如果写关键码的顺序不对或写入了错误的键码，FLASH 写和擦除操作将被禁止，直到下一次系统复位。如果在正确写入关键码之前进行了 FLASH 写或擦除操作，FLASH 写和擦除也将被禁止。每次 FLASH 写和擦除操作之后，FLASH 锁定功能复位；在进行下一次 FLASH 写或擦除操作之前，必须重新写关键码。图 8.2 给出了 FLKEY 寄存器的详细说明。

8.1.2 FLASH 擦除

可以用软件使用 MOVX 指令对 FLASH 存储器编程，像一般的操作数一样为 MOVX 指令提供待编程的地址和数据字节。在使用 MOVX 指令对 FLASH 存储器写入之前，必须先允许 FLASH 写操作。允许 FLASH 写操作的过程是：1) 按顺序向 FLASH 锁定寄存器（FLKEY）写入 FLASH 关键码；2) 将程序存储写允许位 PSWE（PSCTL.0）设置为逻辑 ‘1’（这将使 MOVX 操作指向目标 FLASH 存储器）。PSWE 位将保持置位状态，直到被软件清除。

写 FLASH 存储器可以清除数据位，但不能使数据位置 ‘1’，只有擦除操作能将 FLASH 中的数据位置 ‘1’。所以在写入新值之前，必须先擦除待编程的地址。FLASH 存储器是以 512 字节的扇区为单位组织的，一次擦除操作将擦除整个扇区（将扇区内的所有字节置为 0xFF）。擦除一个扇区（页）的步骤如下：

1. 禁止中断（建议）。
2. 向 FLKEY 写第一个关键码：0xA5。
3. 向 FLKEY 写第二个关键码：0xF1。
4. 置 1 PSEE 位（寄存器 PSCTL），以允许 FLASH 扇区擦除。
5. 置 1 PSWE 位（寄存器 PSCTL），以允许 FLASH 写入。
6. 用 MOVX 指令向待擦除页内的任何一个地址写入一个数据字节。

- 7. 清除 PSWE 位（寄存器 PSCTL）。
- 8. 清除 PSEE 位（寄存器 PSCTL）。

8.1.3 FLASH 写

用软件对 FLASH 字节编程的步骤如下：

- 1. 禁止中断（建议这样做）。
- 2. 擦除包含目标地址的 FLASH 页。
- 3. 向 FLKEY 写第一个关键码：0xA5。
- 4. 向 FLKEY 写第二个关键码：0xF1。
- 5. 置 ‘1’ 寄存器 PSCTL 中的 PSWE 位。
- 6. 清除寄存器 PSCTL 中的 PSEE 位。
- 7. 用 MOVX 指令向扇区内的目标地址写入一个数据字节。
- 8. 清除寄存器 PSCTLPSWE 位。

重复步骤 3-8，直到写完每个字节。在完成了对 FLASH 的数据写入后，PSWE 位应被清 ‘0’，以使 MOVX 指令不再指向程序存储器。

表 8.1 FLASH 存储器电气特性

参 数	条 件	最小值	典型值	最大值	单 位
FLASH 规模	C8051F326/7	16384*	—	—	字节
擦写寿命		20k	100k	—	擦/写
擦除时间	25 MHz 系统时钟	10	15	20	ms
写入时间	25 MHz 系统时钟	40	55	70	μs
*注：位于 0x3E00 ~ 0x3FFF 的 512 字节保留。					

8.2 非易失性数据存储

FLASH 存储器除了用于存储程序代码之外还可以用于非易失性数据存储。这就允许在程序运行时计算和存储类似标定系数这样的数据。数据写入时用 MOVX 指令，读出时用 MOVC 指令。注意：MOVX 读指令总是指向 XRAM。

尽管 FLASH 存储器可以每次写一个字节，但必须首先擦除整个扇区。为了修改一个多字节数据集的某一个字节，整个数据集必须被保存到一个临时存储区。接下来将扇区擦除，更新数据集，最后将数据集写回到原扇区。

8.3 安全选项

CIP-51 提供了安全选项以保护 FLASH 存储器不会被软件意外修改，以及防止产权程序代码和常数被读取。程序存储器写允许（PSCTL 寄存器中的 PSWE）和程序存储器擦除允许（PSCTL 寄存器中的 PSEE）位保护 FLASH 存储器不会被软件意外修改。在用软件修改 FLASH 存储器的内容之前，PSWE 必须被置为逻辑 1；在用软件擦除 FLASH 存储器之前，PSWE 位和 PSEE 位都必须被置为逻辑 1。此外，CIP-51 还提供了可以防止通过 C2 接口读取产权程序代码和常数这一安全功能。

保存在 FLASH 用户空间的最后一个字节中的安全锁定字节保护 FLASH 存储器，使其不能被非保护代码或通过 C2 接口读、写或擦除。FLASH 安全机制允许用户从 0 页（地址 0x0000 ~ 0x01FF）开始锁定 n 个 512 字节的 FLASH 页，其中 n 是安全锁定字节的反码。注意：当任何一个其他 FLASH 页被锁定时，包含 FLASH 安全锁定字节的页也被锁定。见下面的例子。

安全锁定字节：	11111101 b
反码：	00000010 b
<hr/>	
被锁定的 FLASH 页：	2
被锁定的地址：	0x0000 - 0x03FF

有关 FLASH 安全的重要注意事项：

1. 对安全锁定字节的任何位清 0 将锁定所选择的页及包含锁定字节的 FLASH 页；
2. 不能通过 C2 接口对被锁定的页进行读、写或擦除操作；
3. 位于未被锁定的存储器空间的用户固件不能对被锁定的页进行读、写或擦除操作；
4. 位于被锁定的页内的用户固件可以对被锁定的和未被锁定的任何页进行读和写操作，保留区除外；
5. 位于被锁定的页内的用户固件可以擦除任何被锁定的和未被锁定的页，保留区和包含锁定字节的页除外；
6. 被锁定的页只能通过 C2 接口用器件擦除命令解锁；
7. 如果用户固件的 FLASH 访问操作被禁止（根据上面的 3、4、5 条），将产生 FLASH 错误系统复位。

C8051F326/7 全速 USB、16KB FLASH 微控制器

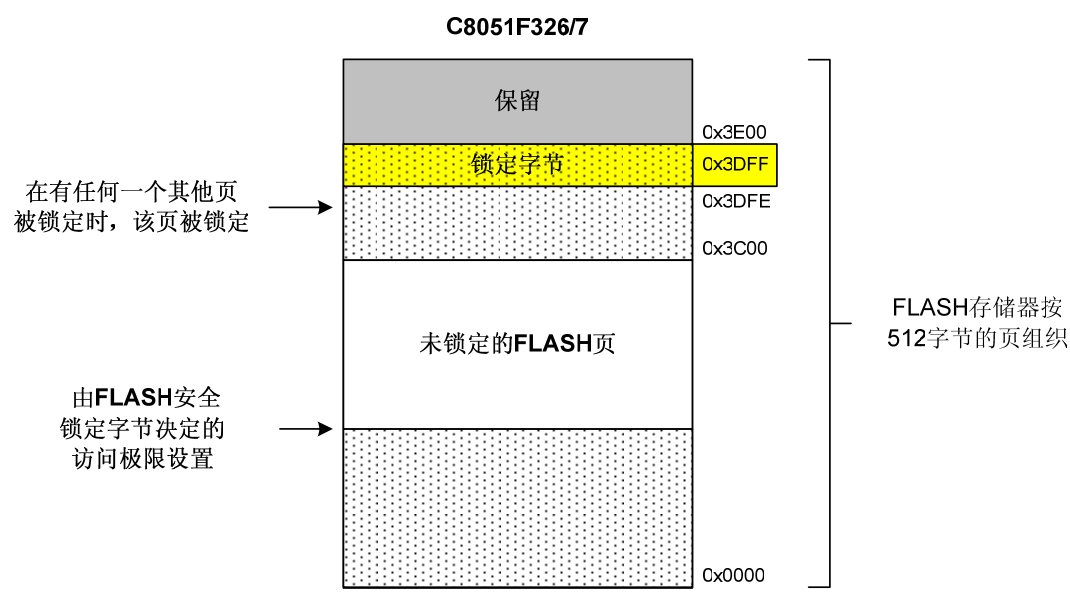


图 8.1 FLASH 程序存储器结构和安全字节

SFR 定义 8.1 PSCTL: 程序存储读写控制

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	—	—	—	—	保留	PSEE	PSWE	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x8F
<p>位 7-3: 未使用。读 = 00000b, 写 = 忽略。</p> <p>位 2: 保留。读 = 0b, 必须写 0b。</p> <p>位 1: PSEE: 程序存储擦除允许 将该位置 1 后允许擦除 FLASH 存储器中的一个页(前提是 PSWE 位也被置 1)。如果该位被置 1, 并且 FLASH 写操作被允许 (PSWE 为逻辑 1), 则用 MOVX 指令进行一次写操作将擦除包含 MOVX 指令寻址地址的那个 FLASH 页。用于写操作的数据可以是任意值。 0: 禁止擦除 FLASH 存储器。 1: 允许擦除 FLASH 存储器。</p> <p>位 0: PSWE: 程序存储写允许 将该位置 1 后允许用 MOVX 指令向 FLASH 存储器写一个字节。在写数据之前必须先进行擦除。 0: 禁止写 FLASH 存储器。 1: 允许写 FLASH 存储器; MOVX 写指令寻址 FLASH 存储器。</p>								

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 8.2 FLKEY: FLASH 锁定和关键码寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xB7
<p>位 7-0: FLKEY: FLASH 锁定和关键码寄存器</p> <p>写:</p> <p>在进行 FLASH 擦除和写操作之前必须写该寄存器。在该寄存器被写入关键码 0xA5 和 0xF1 之前, FLASH 保持锁定状态。写操作的时间并不重要, 但必须按顺序写。每次写或擦除操作都必须写关键码。如果写 FLKEY 操作不正确或在正确写入关键码之前进行了 FLASH 操作, 则 FLASH 将被锁定, 直到下一次系统复位。</p> <p>读:</p> <p>位 1-0 指示当前的 FLASH 锁定状态</p> <p>00: FLASH 写/擦除被锁定。</p> <p>01: 第一个关键码已被写入 (0xA5)。</p> <p>10: FLASH 处于解锁状态 (允许写/擦除)</p> <p>11: FLASH 写/擦除操作被禁止, 直到下一次复位。</p>								

SFR 定义 8.3 FLSCL: FLASH 定时预分频

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
FOSE	保留	保留	保留	保留	保留	保留	保留	10000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xB6
<p>位 7: FOSE: FLASH 单稳态定时器使能</p> <p>该位用于使能 FLASH 读操作单稳态定时器。当 FLASH 单稳态定时器被禁止时, FLASH 读出放大器在整个 FLASH 读周期内被使能。当系统时钟频率低于 10MHz 时, 禁止 FLASH 单稳态定时器会增加系统功耗。</p> <p>0: 禁止 FLASH 单稳态定时器。</p> <p>1: 使能 FLASH 单稳态定时器。</p> <p>位 6-0: 保留。读 = 0, 必须写 0。</p>								

9. 外部 RAM

C8051F326/7 器件内部有 1280 字节的 XRAM。该 XRAM 空间被分成用户 RAM（地址为 0x0000 ~ 0x03FF）和 USB FIFO 空间。USB FIFO 空间只能通过 USB FIFO 寄存器访问。

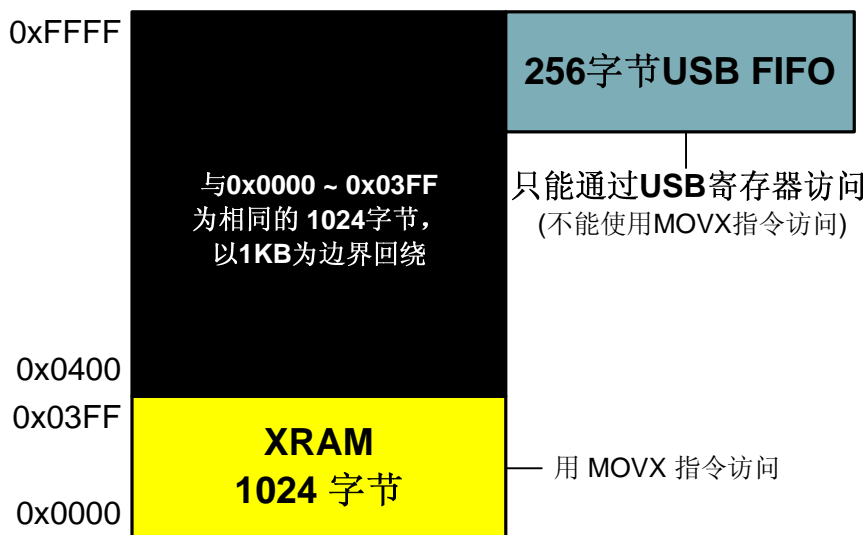


图 9.1 外部数据存储器组织

9.1 访问用户 XRAM

可以用外部传送指令（MOVX）和数据指针（DPTR）访问 XRAM，或者用 MOVX 间接寻址方式。如果 MOVX 指令使用一个 8 位地址操作数（例如 @R1），则 16 位地址的高字节由外部存储器接口控制寄存器（EMI0CN，如图 9.1 所示）提供。注：MOVX 指令还用于写 FLASH 存储器，详见“8. FLASH 存储器”。缺省情况下 MOVX 指令访问 XRAM。

对于任何一种寻址方式，16 位外部数据存储器地址的高 6 位是被“忽略”的。因此，1024 字节的 RAM 以取模的方式映射到整个 64KB 的外部数据存储器地址范围。例如，位于地址 0x0000 的 XRAM 字节也位于 0x0400、0x0800、0x0C00、0x1000 等地址。

9.2 访问 USB FIFO

位于 XRAM 高地址的 256 字节为 USB FIFO 空间。图 9.2 给出了 FIFO 空间 and 用户 XRAM 的扩展示意图。通过 USB FIFO 寄存器访问 FIFO 空间，有关访问 FIFO 空间的详细信息见“12.5 FIFO 管理”。FIFO 存储块工作在 USB 时钟域，因此在访问 FIFO 空间时，USB 时钟必须处于活动状态。

注意：在访问 FIFO 空间时，USB 时钟必须处于活动状态。

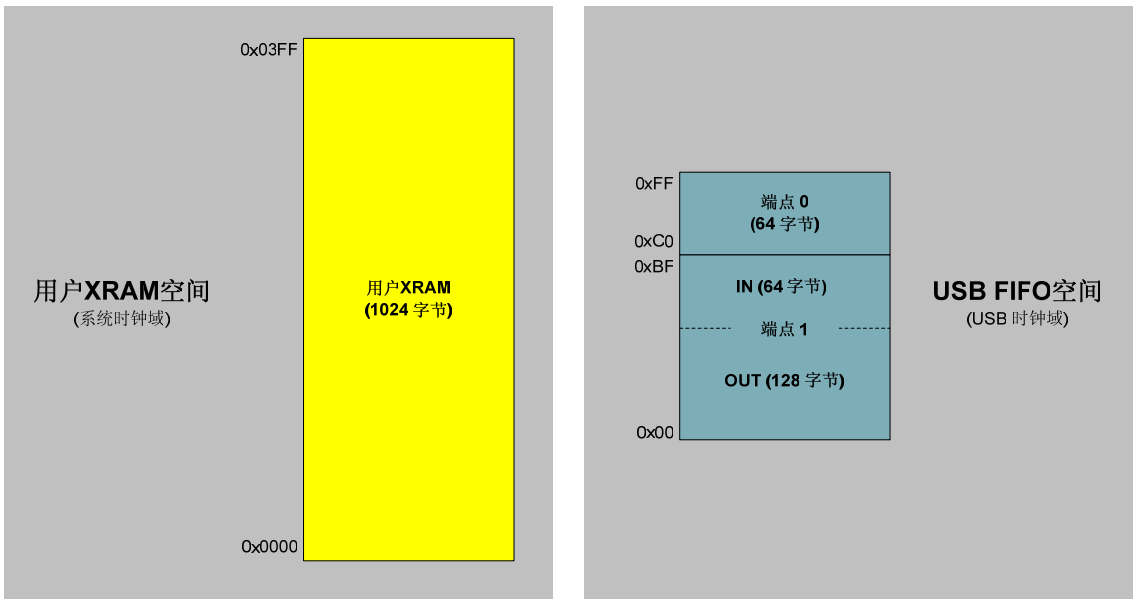


图 9.1 XRAM 存储器扩展示意图

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 9.1 EMI0CN: 外部存储器接口控制

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
-	-	-	-	-	-	PGSEL1	PGSEL0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xAA
<p>位 7-2: 未用。读 = 00000b, 写 = 忽略。</p> <p>位 1-0: PGSEL[1:0]: XRAM 页选择位</p> <p>当使用 8 位的 MOVX 命令时, XRAM 页选择位提供 16 位外部数据存储器地址的高字节, 实际上是选择一个 256 字节的 RAM 页。由于该寄存器的高 6 位被忽略, 所以 1K 地址块以取模的方式映射到整个 64KB 外部数据存储器地址空间。</p>								

10. 振荡器

C8051F326/7 有一个可编程内部振荡器、一个外部时钟输入电路、一个低频内部振荡器和一个 4 倍时钟乘法器。内部振荡器可以通过 OSCICN 和 OSCICL 寄存器来使能/禁止和校准，如图 10.1 所示。低频内部振荡器可以通过 OSCLCN 寄存器来使能/禁止和校准，如图 10.3 所示。系统时钟（SYSCLK）可以来自内部振荡器、外部时钟、低频内部振荡器或 4 倍时钟乘法器二分频。USB 时钟（USBCLK）可以来自内部振荡器/2、外部时钟或 4 倍时钟乘法器。表 10.3 给出了内部振荡器的电气特性。

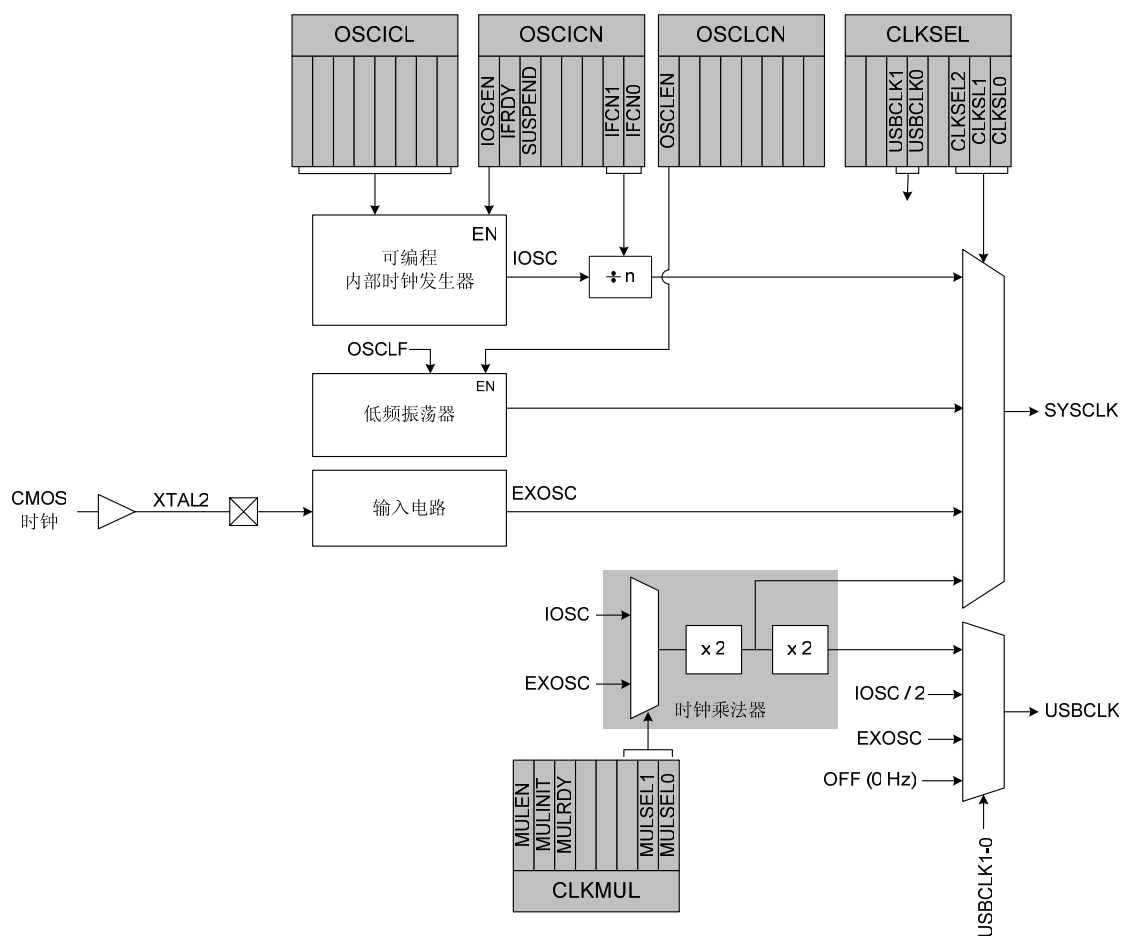


图 10.1 振荡器框图

C8051F326/7 全速 USB、16KB FLASH 微控制器

10.1 可编程内部振荡器

C8051F326/7 器件包含一个可编程内部振荡器，该振荡器在系统复位后被默认为系统时钟。内部振荡器的周期可以通过 OSCICL 寄存器编程。对于 C8051F326/7 器件，OSCICL 已经过工厂校准，使频率为 12MHz。表 10.3 给出了精密内部振荡器的电气特性。注意：系统时钟可以从内部振荡器分频得到，分频系数由寄存器 OSCICN 中的 IFCN 位设定，可为 1、2、4 或 8。复位后的缺省分频系数为 8。

10.1.1 C8051F326/7 的内部振荡器编程

OSCICL 复位值已经过工厂校准，对应 12MHz 的内部振荡器频率，精度为±1.5%，该频率适合作为 USB 时钟（见 10.4 节）。软件可以使用 OSCICL 寄存器调整内部振荡器的频率。

注意：一旦内部振荡器的频率被修改，则它不能再被用作 USB 时钟。在发生任何器件复位后，内部振荡器的频率恢复为初始的出厂校准频率，只有在这个频率内部振荡器才适合作为 USB 时钟。

10.1.2 内部振荡器挂起方式

通过向寄存器 OSCICN 中的 SUSPEND 位写 ‘1’ 可以将内部振荡器置于挂起方式。在挂起方式，内部振荡器停止运行，直到检测到一个非空闲 USB 事件或 VBUS 输入信号与 VBPOL 位（位于寄存器 REG0CN）所选择的极性匹配。注意：要检测 USB 事件，USB 收发器必须被使能。

SFR 定义 10.1 OSCICN：内部振荡器控制寄存器

R/W	R/W	R/W	R	R	R	R/W	R/W	复位值
IOSCEN	IFRDY	SUSPEND	—	—	—	IFCN1	IFCN0	11000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xB2
位 7: IOSCEN: 内部振荡器使能位								
0: 内部振荡器禁止。								
1: 内部振荡器使能。								
位 6: IFRDY: 内部振荡器频率准备好标志								
0: 内部振荡器未运行在编程频率。								
1: 内部振荡器按编程频率运行。								
位 5: SUSPEND: 强制挂起控制位								
向该位写 1 将强制内部振荡器停止运行。在发生 USB 非空闲事件(即 RESUME 命令)或 VBUS 中断事件（见 SFR 定义 5.1）后内部振荡器重新启动。								
位 4-2: 未用。读 = 000b, 写 = 忽略。								
位 1-0: IFCN1-0: 内部振荡器频率控制位								
00: SYSCLK 为内部频振荡器 8 分频。								
01: SYSCLK 为内部振荡器 4 分频。								
10: SYSCLK 为内部振荡器 2 分频。								
11: SYSCLK 为内部振荡器输出（不分频）。								

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 10.2 OSCICL：内部振荡器校准寄存器

R	R	R	R/W	R/W	R/W	R/W	R/W	复位值
—	—	—	OSCCAL					可变
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xB3
<p>位 7-5： 未用。读 = 000b，写 = 忽略。</p> <p>位 4-0： OSCCAL： 内部振荡器校准值。 这些位决定内部振荡器的周期</p> <p>注：如果 OSCCAL 的复位值与 ΔOSCCAL 之和大于 31 或小于 0，则器件不能产生所需要的频率。</p> <p>注：当时钟恢复电路被使能时，该寄存器的内容不确定。有关时钟恢复电路的详细信息见“12.4 USB 时钟配置”。</p>								

10.2 内部低频（L-F）振荡器

C8051F326/7 器件包含一个低频振荡器。OSCLCN 寄存器（见 SFR 定义 10.3）用于使能/禁止该振荡器。

SFR 定义 10.3 OSCLCN：内部 L-F 振荡器控制寄存器

R/W	R	R	R	R	R	R	R	复位值
OSCLCN	—	—	—	—	—	—	—	0xxxxxxx
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xE3
<div>位 7: OSCLCN：内部 L-F 振荡器使能位。 0：内部 L-F 振荡器禁止。 1：内部 L-F 振荡器使能。</div> <div>位 6-0: 未用。读 = 0000000b，写 = 忽略。</div>								

10.3 4 倍时钟乘法器

4 倍时钟乘法器允许使用 12MHz 振荡器产生全速 USB 通信所需要的 48MHz 时钟(见“12.4 USB 时钟配置”)。时钟乘法器输出的二分频也可以被用作系统时钟。关于系统时钟和 USB 时钟源选择见 10.4 节。

用 CLKMUL 寄存器配置 4 倍时钟乘法器。配置和使能 4 倍时钟乘法器的步骤如下：

- 1. 通过向寄存器 CLKMUL 写 0x00 来复位时钟乘法器。
- 2. 用 MULSEL 位选择时钟乘法器的输入。
- 3. 用 MULEN 位使能时钟乘法器 (CLKMUL |= 0x80)。
- 4. 延时大于 5 μs。
- 5. 用 MULINIT 位初始化时钟乘法器 (CLKMUL |= 0xC0)。
- 6. 查询等待 MULRDY => ‘1’。

注意：当使用外部时钟作为 4 倍时钟乘法器的输入时，外部源必须在乘法器被初始化之前达到稳定。

SFR 定义 10.4 CLKMUL：时钟乘法器控制寄存器

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	复位值
MULEN	MULINIT	MULRDY	—	—	—	—	MULSEL	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xB9
位 7:	MULEN：时钟乘法器使能位 0：时钟乘法器禁止。 1：时钟乘法器使能。							
位 6:	MULINIT：时钟乘法器初始化控制位 0：当时钟乘法器被使能时，该位应为 ‘0’。时钟乘法器被使能后向该位写 ‘1’ 将初始化时钟乘法器。当时钟乘法器稳定后，MULRDY 的读出值为 ‘1’。							
位 5:	MULRDY：时钟乘法器准备好标志 该只读位指示时钟乘法器的状态。 0：时钟乘法器未准备好。 1：时钟乘法器已准备好（锁定）。							
位 4-2:	未用。读 = 000b，写 = 忽略。							
位 1-0:	MULSEL：时钟乘法器输入选择位 这两位选择去时钟乘法器的时钟。							
MULSEL		选择的时钟						
0		内部振荡器						
1		外部时钟						

10.4 系统时钟和 USB 时钟选择

内部振荡器的起动时间很短，因此可以在使能内部振荡器的 OSCICN 写操作后立即选择内部振荡器为系统时钟或 USB 时钟。如果选择外部时钟作为系统或 USB 时钟，则启动时间由外部时钟的指标决定。

10.4.1 系统时钟选择

寄存器 CLKSEL 中的 CLKSL[2:0]位选择用作系统时钟的振荡源。当选择外部时钟作为系统时钟时，CLKSL[2:0]必须被设置为 001b。当选择内部振荡器作为系统时钟时，外部时钟仍然可以给外设（定时器、UART、USB）提供时钟。系统时钟可以在内部振荡器、外部时钟、低频振荡器和 4 倍时钟乘法器之间自由切换，只要所选择的振荡器被使能并可提供稳定时钟。

10.4.2 USB 时钟选择

寄存器 CLKSEL 中的 USBCLK[1:0]位选择用作 USB 时钟的振荡源。USB 时钟可以来自 4 倍时钟乘法器输出、内部振荡器/2 或外部时钟。USB 时钟源可以被关闭。当 USB0 工作在全速方式时，USB 时钟必须是 48 MHz；当 USB0 工作在低速方式时，USB 时钟必须是 6 MHz。USB 的时钟选择见图 10.5。

下面给出了全速和低速方式下 USB 时钟配置的几个例子：

表 10.1 典型 USB 全速方式时钟设置

内部振荡器		
时钟信号	输入源选择	寄存器位设置
USB 时钟	时钟乘法器	USBCLK = 00b
时钟乘法器输入	内部振荡器*	MULSEL = 0b
内部振荡器	不分频	IFCN = 11b
外部时钟		
时钟信号	输入源选择	寄存器位设置
USB 时钟	时钟乘法器	USBCLK = 10b
时钟乘法器输入	外部时钟*	MULSEL = 1b
端口 I/O	12 MHz CMOS 时钟	INPUTEN = 1b (GPIOCN.6)
*对于该配置，时钟恢复电路必须被使能。		

表 10.2 典型 USB 低速方式时钟设置

内部振荡器		
时钟信号	输入源选择	寄存器位设置
USB 时钟	内部振荡器/2	USBCLK = 01b
内部振荡器	不分频	IFCN = 11b
外部时钟		
时钟信号	输入源选择	寄存器位设置
USB 时钟	外部时钟	USBCLK = 10b
端口 I/O	6 MHz CMOS 时钟	INPUTEN = 1b (GPIOCN.6)

SFR 定义 10.5 CLKSEL：时钟选择寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	—	USBCLK		—	CLKSL			00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xA9

位 7-6: 未用。读 = 00b, 写 = 忽略。

位 5-4: USBCLK1-0: USB 时钟选择位

这些位选择去 USB0 的时钟。当 USB0 工作于全速方式时，所选时钟应为 48 MHz。当 USB0 工作于低速方式时，所选时钟应为 6 MHz。

USBCLK	选择的时钟
00	4 倍时钟乘法器
01	内部振荡器/2
10	外部时钟
11	时钟关闭（0Hz）

位 3: 未用。读 = 0b, 写 = 忽略。

位 2-0: CLKSL2-0: 系统时钟选择位

这些位选择系统时钟源。

CLKSL	选择的时钟
000	内部振荡器（由寄存器 OSCICN 中的 IFCN 位选择分频系数）
001	外部时钟
010	4 倍时钟乘法器/2
011	低频振荡器
1xx	保留

C8051F326/7 全速 USB、16KB FLASH 微控制器

表 10.3 内部振荡器电气特性

-40℃到+85℃（除非特别说明）

参 数	条 件	最小值	典型值	最大值	单位
内部振荡器频率	复位频率	11.82	12	12.18	MHz
内部振荡器电源电流(自 VDD)	OSCICN.7 = 1	—	574	—	μA
USB时钟频率 ¹	全速	47.88	48	48.12	MHz
	低速	5.91	6	6.09	
内部低频振荡器（使用工厂校准设置）					
内部振荡器频率	复位频率	—	88	—	KHz
内部振荡器电源电流(自 VDD)	25℃, VDD = 3.0V OSCLCN.7 = 1	—	17	—	μA
电源电压敏感度	恒温	—	-3±0.1 ²	—	%/V
温度敏感度	恒压	—	20±8	—	ppm/℃
注：					
1. 只适用于外部振荡源。					
2. 表示平均值±一个标准差。					

11. 端口输入/输出

片内数字资源可以通过 15 个 I/O 引脚使用。端口引脚组织如图 11.1 所示。每个端口引脚都可以被作为通用 I/O（GPIO）使用。某些端口引脚可被专门用作特殊信号，如/SYSCLK、UART 的 TX 和 RX、XTAL2 外部时钟输入。

所有端口 I/O 都耐 5V 电压（参见图 11.2 的端口 I/O 单元电路）。端口 I/O 单元可以被配置为漏极开路或推挽方式（在端口输出方式寄存器 PnMDOUT 中设置，n = 0, 2, 3）。表 11.1 给出了端口 I/O 的电气特性。

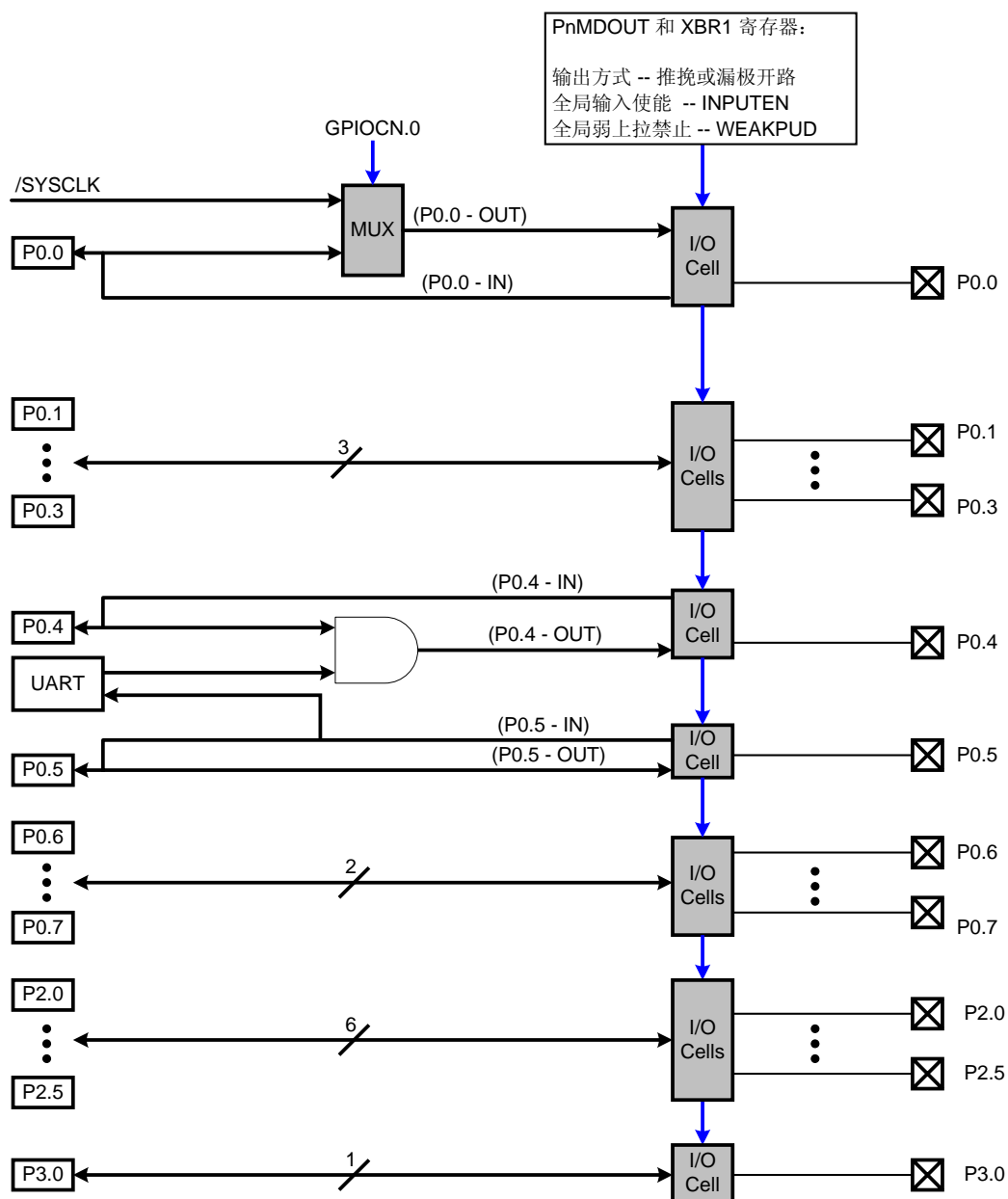


图 11.1 端口 I/O 功能框图

C8051F326/7 全速 USB、16KB FLASH 微控制器

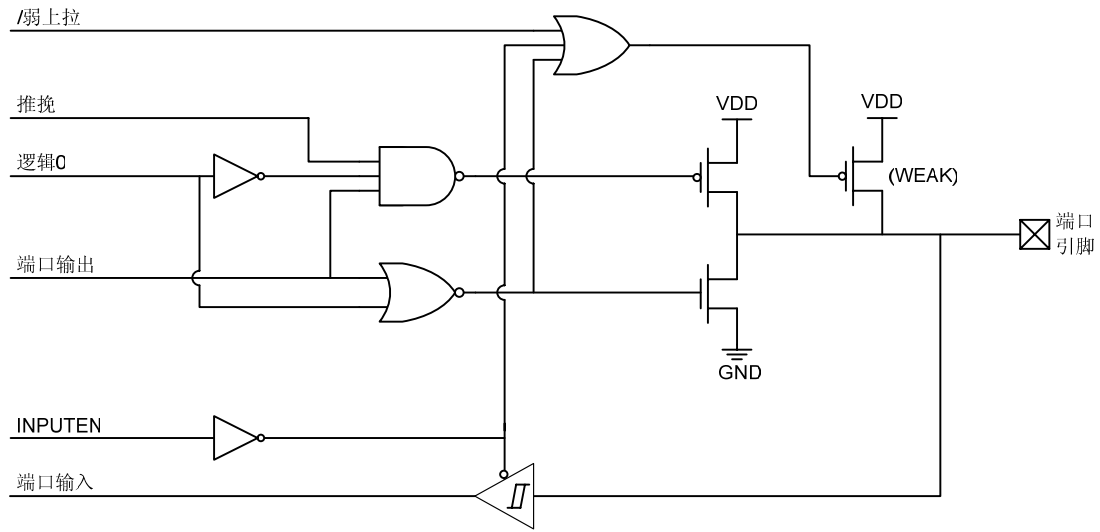


图 11.2 端口 I/O 单元框图

11.1 端口 I/O 初始化

端口 I/O 初始化包括以下步骤：

1. 选择端口引脚用作输入还是输出。
2. 如果是输出，选择输出方式：漏极开路或推挽。
3. 根据所希望的输入或输出配置来设置 PnMDOUT 和 Pn 锁存器。
4. 选择是否让/SYSCLK 输出到 P0.0 引脚，并设置 GPIOCN.0。
5. 使能全局输入（INPUTEN = 1）。

端口引脚可被用作数字输入或输出。要将一个端口引脚配置为数字输入，向 PnMDOUT 寄存器中的对应位写 0，并向对应的端口锁存器（Pn）写 1。读端口引脚时，由引脚的实际电压值决定是逻辑 0 还是逻辑 1；端口锁存器只能写入。

数字输出引脚可被配置为漏极开路或推挽方式。在漏极开路方式（PnMDOUT 中的对应位为 0），当端口锁存器为逻辑 0 时低输出驱动器导通，当端口锁存器为逻辑 1 时低输出驱动器截止。无论端口锁存器的设置如何，高输出驱动器总是断开。在漏极开路方式，当端口锁存器为逻辑 1 时，输出端口引脚为高阻状态。如果要引脚用作输出，应接一个外部上拉电阻。该方式在与 5V 逻辑接口时有用。

每个端口引脚都有内部上拉，当 WEAKPUD 位为 0、端口输出方式被配置为漏极开路、且端口锁存器为逻辑 1（引脚为高阻输入）时，内部上拉被使能。如果端口引脚被配置为推挽方式或端口锁存器为逻辑 0，则内部上拉被禁止以避免不必要的功耗。

在推挽方式（PnMDOUT 中的对应位为 1），总是有一个输出驱动器导通。当端口锁存器为逻辑 0 时，低输出驱动器导通，高输出驱动器断开；当端口锁存器为逻辑 1 时，低输出驱动器断开，高输出驱动器导通。注意，在推挽方式，输出引脚的电压反映输出端口锁存器的逻辑电平。该方式不能用于驱动比高于 VIO 或 VDD 的电平。

在每个引脚都被正确配置为输入或输出后。特殊信号可以被连接到所选择端口引脚。特殊信号包括/SYSCLK（在 P0.0）、XTAL2（在 P0.3）、UART TX（在 P0.4）和 UART RX（在 P0.5）。GPIOCN.1 被置 1 时，/SYSCLK 信号可被连接到 P0.0。XTAL2 时钟输入总是连接到 P0.3。UART TX 信号总是被使能，并与 P0.4 锁存器位相与。当使用 UART 时，P0.4 锁存器位应为逻辑 1，以使 UART 能控制 TX 引脚。如果端口锁存器在某一时刻被写入 0，则 TX 信号将被强制为逻辑 0。当不使用 UART 时，TX 信号的值总是为逻辑 1，P0.4 可被用作 GPIO。

注意：置 1GPIOCN 中的 INPUTEN 位将全局使能数字输入。在全局输入被使能之前，所有端口引脚都保持为输出方式，不能用于检测端口引脚的逻辑电平。要使用 UART RX、XTAL2 或/INT0 输入，INPUTEN 位必须被置 1。

11.2 通用端口 I/O

通过对应的特殊功能寄存器访问端口 0、端口 2 和端口 3。这些寄存器既可以按位寻址也可以按字节寻址。向端口写入时，数据被锁存到端口数据寄存器中，以保持引脚上的输出数据值不变。如果 INPUTEN 被置 1，读操作返回端口输入引脚的逻辑状态。但在对端口 SFR 执行下面的读-修改-写指令（ANL、ORL、XRL、JBC、CPL、INC、DEC、DJNZ）和对端口 SFR 中的某一位执行 MOV、CLR、SETB 期间例外。这些指令读端口寄存器（而不是引脚）的值，修改后再写回端口 SFR。

SFR 定义 11.1 GPIOCN：全局端口 I/O 控制寄存器

R/W	R/W	R	R	R	R	R	R	复位值
WEAKPUD	INPUTEN	—	—	—	—	—	SYSCLK	01000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址： 0xE2
<p>位 7: WEAKPUD: 端口 I/O 弱上拉禁止位。 0: 弱上拉使能（端口锁存器被设置为逻辑 0 或被配置为推挽方式的 I/O 引脚除外）。 1: 弱上拉禁止。</p> <p>位 6: INPUTEN: 全局数字输入使能位。 0: 端口 I/O 输入路径禁止；端口引脚只能用于输出。 1: 端口 I/O 输入路径使能。</p> <p>位 5-1: 未用。读 = 00000b, 写 = 忽略。</p> <p>位 0: SYSCLK: /SYSCLK 使能位 0: /SYSCLK 不连接到 P0.0 引脚，P0.0 锁存器连接到 P0.0 引脚。 1: /SYSCLK 连接到 P0.0 引脚，P0.0 锁存器不连接到 P0.0 引脚。</p>								

SFR 定义 11.2 P0：端口 0 寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	11111111
位7	位6	位5	位4	位3	位2	位1	位0 (可位寻址)	SFR地址： 0x80
<p>位 7-0: P0.[7:0] 写 — 输出出现在 I/O 引脚。 0: 逻辑低电平输出。 1: 逻辑高电平输出。（若相应的 P0MDOUT.n 位 = 0，则为高阻态）。 读 — 如果 INPUTEN = 0，则总是返回 0。否则，直接读端口引脚。 0: P0.n 为逻辑低电平。 1: P0.n 为逻辑高电平。</p>								

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 11.3 P0MDOUT: 端口 0 输出方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xA4
位 7-0: P0.7 – P0.0 输出方式配置位（分别对应）。 0: 对应的 P0.n 输出为漏极开路。 1: 对应的 P0.n 输出为推挽方式。								

SFR 定义 11.4 P2: 端口 2 寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	—	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	11111111
位7	位6	位5	位4	位3	位2	位1	位0 (可位寻址)	SFR地址: 0xA0
位 7-6: 未用。 位 5-0: P2.[5:0] 写 — 输出出现在 I/O 引脚。 0: 逻辑低电平输出。 1: 逻辑高电平输出。（若相应的 P2MDOUT.n 位 = 0，则为高阻态）。 读 — 如果 INPUTEN = 0，则总是返回 0。否则，直接读端口引脚。 0: P2.n 为逻辑低电平。 1: P2.n 为逻辑高电平。								

SFR 定义 11.5 P2MDOUT: 端口 2 输出方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	—							00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xA6
位 7-6: 未用。读 = 00b，写 = 忽略。 位 5-0: P2.5 – P2.0 输出方式配置位（分别对应）。 0: P2.n 输出为漏极开路。 1: P2.n 输出为推挽方式。								

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 11.6 P3: 端口 3 寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	—	—	—	—	—	—	P3.0	11111111
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址:
							(可位寻址)	0xB0
位 7-1: 未用。读 = 0000000b, 写 = 忽略。								
位 0: P3.0								
写 — 输出出现在 I/O 引脚。								
0: 逻辑低电平输出。								
1: 逻辑高电平输出。(若相应的 P3MDOUT.n 位 = 0, 则为高阻态)。								
读 — 如果 INPUTEN = 0, 则总是返回 0。否则, 直接读端口引脚。								
0: P3.n 为逻辑低电平。								
1: P3.n 为逻辑高电平。								

SFR 定义 11.7 P3MDOUT: 端口 3 输出方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	—	—	—	—	—	—		00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址:
								0xA7
位 7-1: 未使用。读 = 0000000b。写 = 忽略。								
位 0: P3.0 输出方式配置位。								
0: P3.0 输出为漏极开路。								
1: P3.0 输出为推挽方式。								

C8051F326/7 全速 USB、16KB FLASH 微控制器

表 11.1 端口 I/O 直流电气特性 (C8051F326)

VDD = 2.7V – 3.6V, -40°C 到 +85°C (除非特别说明)。

参 数	条 件	最小值	典型值	最大值	单 位
VIO = 2.7 ~ 3.6V					
输出高电压 (V _{OH})	I _{OH} = -10μA, 端口I/O为推挽方式 I _{OH} = -3mA, 端口I/O为推挽方式 I _{OH} = -10mA, 端口I/O为推挽方式	VIO-0.1 VIO-0.7 —	— — VIO-0.8	— — —	V
输出低电压 (V _{OL})	I _{OL} = 10μA I _{OL} = 8.5mA I _{OL} = 25mA	— — —	— — 1.0	0.1 0.6 —	V
输入高电压 (V _{IH})		2.0	—	—	V
输入低电压 (V _{IL})		—	—	0.8	V
输入漏电流	弱上拉禁止 弱上拉使能, V _{IN} = 0 V	— —	— 25	±1 50	μA
VIO = 1.8V					
输出高电压 (V _{OH})	I _{OH} = -10μA, 端口I/O为推挽方式 I _{OH} = -1mA, 端口I/O为推挽方式	VIO-0.1 VIO-0.4	— —	— —	V
输出低电压 (V _{OL})	I _{OL} = 10μA I _{OL} = 3mA	— —	— —	0.1 0.4	V
输入高电压 (V _{IH})		VDDx0.7	—	—	V
输入低电压 (V _{IL})		—	—	VDDx0.3	V
输入漏电流	弱上拉禁止 弱上拉使能, V _{IN} = 0 V	— —	— 6	±1 15	μA

表 11.2 端口 I/O 直流电气特性 (C8051F327)

VDD = 2.7V – 3.6V, -40°C 到 +85°C (除非特别说明)。

参 数	条 件	最小值	典型值	最大值	单 位
输出高电压 (V _{OH})	I _{OH} = -10μA, 端口I/O为推挽方式 I _{OH} = -3mA, 端口I/O为推挽方式 I _{OH} = -10mA, 端口I/O为推挽方式	VDD-0.1 VDD-0.7 —	— — VDD-0.8	— — —	V
输出低电压 (V _{OL})	I _{OL} = 10μA I _{OL} = 8.5mA I _{OL} = 25mA	— — —	— — 1.0	0.1 0.6 —	V
输入高电压 (V _{IH})		2.0	—	—	V
输入低电压 (V _{IL})		—	—	0.8	V
输入漏电流	弱上拉禁止 弱上拉使能, V _{IN} = 0 V	— —	— 25	±1 50	μA

12. 通用串行总线控制器（USB0）

C8051F326/7 器件集成了全速/低速 USB 功能控制器，用于实现 USB 接口的外部设备。USB 功能控制器（USB0）由串行接口引擎（SIE）、USB 收发器（包括匹配电阻和可配置上拉电阻）、256B FIFO 存储块和时钟恢复电路（可以不用晶体）组成，不需要外部元件。USB 功能控制器和收发器符合通用串行总线规范 2.0 版。

注：C8051F326/7 不能被用作 USB 主设备

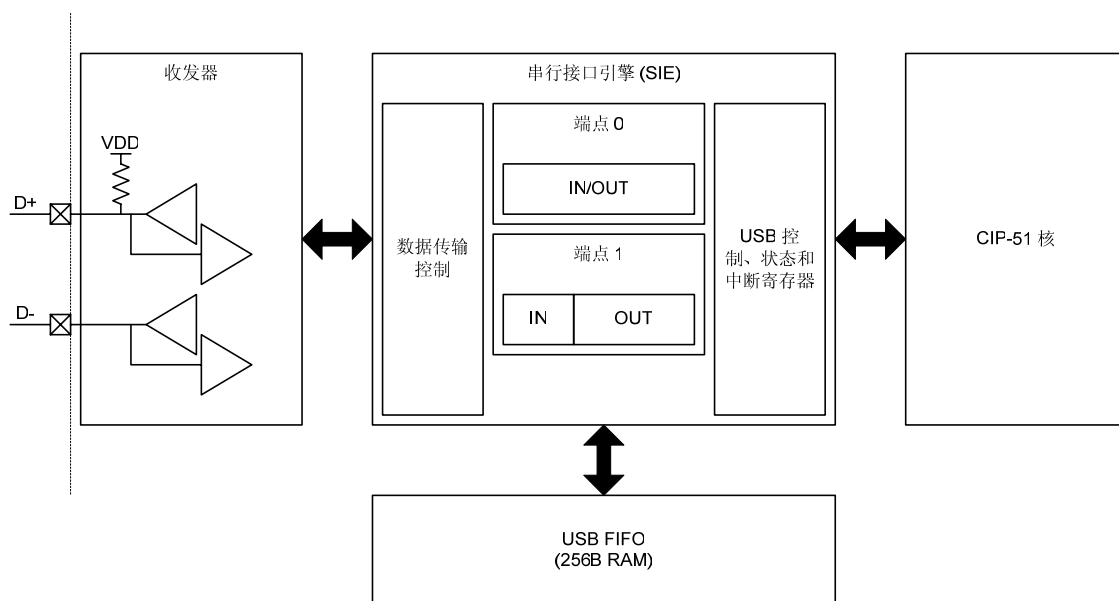


图 12.1 USB0 原理框图

注：本文档假设读者对 USB 协议有全面的了解。本文档所使用的术语和缩略语在 USB 规范中有定义。我们希望您能在进一步阅读之前重温 USB 规范的最新版本。

12.1 端点地址

共有 3 个端点管道。控制端点（端点 0）总是作为双向 IN/OUT 端点。端点 1 被作为一个 64 字节的 IN 管道和一个 128 字节的 OUT 管道。

表 12.1 端点地址

端点	相应的管道	USB 协议地址
端点 0	端点 0 输入（IN）	0x00
	端点 0 输出（OUT）	0x00
端点 1	端点 1 输入（IN）	0x81
	端点 1 输出（OUT）	0x01

12.2 USB 收发器

用 USB0XCN 寄存器（见图 12.1）配置 USB 收发器。配置内容包括收发器使能/禁止、上拉电阻使能/禁止及器件速度选择（全速或低速）。当位 SPEED = 1 时，USB0 工作在全速 USB 方式，片内上拉电阻（如果被使能）出现在 D+引脚。当位 SPEED = 0 时，USB0 工作在低速 USB 方式，片内上拉电阻（如果被使能）出现在 D-引脚。寄存器 USB0XCN 的位 4~0 用于收发器测试（见图 12.2 的说明）。上拉电阻只有在有 VBUS 信号时被使能（有关 VBUS 检测的详细信息见“5.2 VBUS 检测”）。

注意：USB 时钟应在收发器被使能之前就处于活动状态。

C8051F326/7 全速 USB、16KB FLASH 微控制器

USB 寄存器定义 12.1 USB0XCN: USB0 收发器控制寄存器

R/W	R/W	R/W	R/W	R/W	R	R	R	复位值
PREN	PHYEN	SPEED	PHYTST1	PHYTST0	DFREC	Dp	Dn	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0xD7

位 7:

PREN: 内部上拉电阻使能

上拉电阻的位置 (D+或 D-) 由 SPEED 位决定。

0: 禁止内部上拉电阻 (器件实际上脱离 USB 网络)。

1: 当 VBUS 有信号时使能内部上拉电阻 (器件与 USB 网络连接)。

位 6:

PHYEN: 物理层使能

该位使能/禁止 USB0 物理层收发器。

0: 禁止收发器 (挂起)。

1: 使能收发器 (正常)。

位 5:

SPEED: USB0 速度选择

该位选择 USB0 的速度。

0: USB0 工作在低速方式。内部上拉电阻 (如果被使能) 出现在 D-引脚。

1: USB0 工作在全速方式。内部上拉电阻 (如果被使能) 出现在 D+引脚。

位 4-3:

PHYTST1-0: 物理层测试

这两位用于测试 USB0 收发器。

PHYTST[1:0]	方 式	D+	D-
00b	方式 0: 正常 (非测试方式)	X	X
01b	方式 1: 强加差分 ‘1’	1	0
10b	方式 2: 强加差分 ‘0’	0	1
11b	方式 3: 强加单端 ‘0’	0	0

位 2:

DFREC: 差分接收器

该位的状态指示出现在 D+和 D-引脚的当前差分值 (当 PHYEN = ‘1’ 时)。

0: 差分 ‘0’ 出现在总线上。

1: 差分 ‘1’ 出现在总线上。

位 1:

Dp: D+信号状态

该位指示 D+引脚的当前逻辑电平。

0: D+信号当前为逻辑 0。

1: D+信号当前为逻辑 1。

位 0:

Dn: D-信号状态

该位指示 D-引脚的当前逻辑电平。

0: D-信号当前为逻辑 0。

1: D-信号当前为逻辑 1。

12.3 USB 寄存器访问

对表 12.2 中列出的 USB0 控制器寄存器的访问是通过两个特殊功能寄存器来实现的，这两个特殊功能寄存器是：USB0 地址寄存器（USB0ADR）和 USB0 数据寄存器（USB0DAT）。USB0ADR 寄存器选择读/写 USB0DAT 寄存器时的目标 USB 寄存器。

对端点控制/状态寄存器的访问是通过先写 USB 寄存器 INDEX（用目标端点号）实现的。目标端点号被写入 INDEX 寄存器后，就可以访问与目标端点对应的控制/状态寄存器。表 12.2 列出了端点控制/状态寄存器。端点控制/状态寄存器见表 12.2 中的“索引寄存器”部分。

注意：访问 USB 寄存器时，USB 时钟必须是活动的。

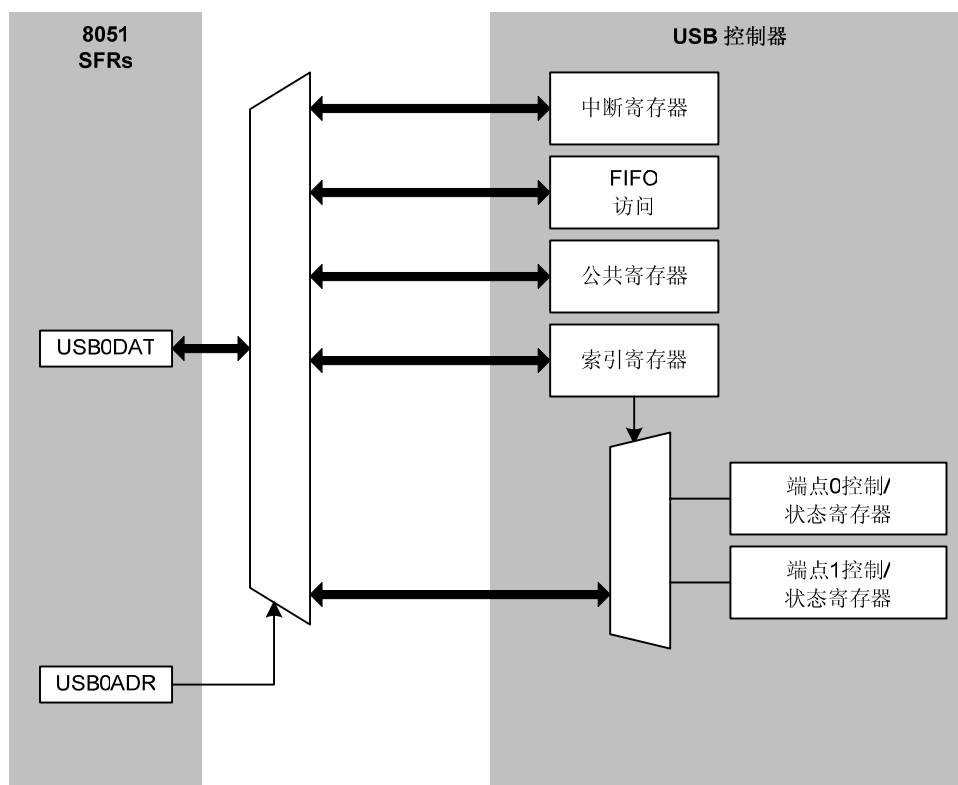


图 12.2 USB0 寄存器访问示意图

C8051F326/7 全速 USB、16KB FLASH 微控制器

USB 寄存器定义 12.2 USB0ADR: USB0 间接地址寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
BUSY	AUTORD	USBADDR						00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x96
<p>位 7: BUSY: USB0 寄存器读忙标志 该位在间接 USB0 寄存器访问期间使用。软件应通过向该位写 1 来启动对 USB0 寄存器的读操作, 目标 USB0 寄存器的地址由 USBADDR 位(USB0ADR.[5-0])给出。可以在同一个对 USB0ADR 的写操作中写目标地址和 BUSY 位。在 BUSY 位被置 1 后, 当目标寄存器的数据已经在 USB0DAT 寄存器中准备好时, 硬件将 BUSY 位清 0。软件在写 USB0DAT 之前应先检查 BUSY 位是否为 0。 写: 0: 无作用。 1: 启动 USB0 间接寄存器读操作, 目标地址由 USBADDR 位给出。 读: 0: USB0DAT 寄存器数据有效。 1: USB0 正在访问间接寄存器; USB0DAT 寄存器数据无效。</p>								
<p>位 6: AUTORD: USB0 寄存器自动读标志位 该位用于 FIFO 块读。 0: 每次 USB0 间接寄存器读操作都必须手动写 BUSY 位。 1: 当软件读 USB0DAT 时, 下一个间接寄存器读操作将被自动启动(USBADDR 位不变)。</p>								
<p>位 5-0: USBADDR: USB0 间接寄存器地址 这些位保持用于间接访问 USB0 寄存器的 6 位地址。表 12.2 列出了 USB0 寄存器和它们的间接地址。读和写 USB0DAT 将访问由 USBADDR 位指向的目标寄存器。</p>								

C8051F326/7 全速 USB、16KB FLASH 微控制器

USB 寄存器定义 12.3 USB0DAT: USB0 数据寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
USB0DAT								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x97

该 SFR 用于间接读和写 USB0 寄存器。

间接写步骤:

1. 查询 BUSY 位 (USB0ADR.7) => ‘0’。
2. 将目标 USB0 寄存器地址装入到寄存器 USB0ADR 中的 USBADDR 位。
3. 向 USB0DAT 写数据。
4. 重复上述步骤 (当写同一个 USB0 寄存器时, 步骤 2 可以被跳过)。

间接读步骤:

1. 查询 BUSY 位 (USB0ADR.7) => ‘0’。
2. 将目标 USB0 寄存器地址装入到寄存器 USB0ADR 中的 USBADDR 位。
3. 向寄存器 USB0ADR 中的 BUSY 位写 ‘1’ (步骤 2 和步骤 3 可以在同一个写操作完成)。
4. 查询 BUSY 位 (USB0ADR.7) => ‘0’。
5. 从 USB0DAT 读数据。
6. 重复从步骤 2 开始的步骤 (当读同一个 USB0 寄存器时, 步骤 2 可以被跳过; 当 AUTORD 位 (USB0ADR.6) 为逻辑 1 时, 步骤 3 可以被跳过)。

USB 寄存器定义 12.4 INDEX: USB0 端点索引寄存器 (USB 寄存器)

R	R	R	R	R/W	R/W	R/W	R/W	复位值
—	—	—	—	EPSEL				00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x0E

位 7-4: 未使用。读 = 0000b, 写 = 忽略。

位 3-0: EPSEL: 端点选择
当访问 USB0 索引寄存器时, 这些位选择目标端点。

索引号	目标端点
0x0	0
0x1	1
0x2 - 0xF	保留

C8051F326/7 全速 USB、16KB FLASH 微控制器

表 12.2 USB0 控制器寄存器

USB 寄存器名称	USB 寄存器地址	说 明
中断寄存器		
IN1INT	0x02	端点 0 和端点 1 输入中断标志
OUT1INT	0x04	端点 1 输出中断标志
CMINT	0x06	公共 USB 中断标志
IN1IE	0x07	端点 0 和端点 1 输入中断使能
OUT1IE	0x09	端点 1 输出中断使能
CMIE	0x0B	公共 USB 中断使能
公共寄存器		
FADDR	0x00	功能地址
POWER	0x01	电源管理
FRAMEL	0x0C	帧号低字节
FRAMEH	0x0D	帧号高字节
INDEX	0x0E	端点索引选择
CLKREC	0x0F	时钟恢复控制
FIFO _n	0x20 - 0x21	端点 0 – 1 FIFO
索引寄存器		
E0CSR	0x11	端点 0 控制/状态
EINCSRL		端点输入控制/状态低字节
EINCSRH	0x12	端点输入控制/状态高字节
EOUTCSRL	0x14	端点输出控制/状态低字节
EOUTCSRH	0x15	端点输出控制/状态高字节
E0CNT	0x16	端点 0 FIFO 中接收的字节数
EOUTCNTL		端点输出包计数低字节
EOUTCNTH	0x17	端点输出包计数高字节

12.4 USB 时钟配置

USB0 具有全速和低速 USB 通信功能。通信速度通过特殊功能寄存器 USB0XCN 中的 SPEED 位选择。当 USB0 工作在低速方式时，USB0 时钟必须为 6 MHz；当 USB0 工作在高速方式时，USB0 时钟必须为 48 MHz。时钟选择见“10. 振荡器”。USB0 时钟通过特殊功能寄存器 CLKSEL 选择（见图 10.6）。在使能时钟恢复电路之前，USB 收发器必须被使能。

时钟恢复电路用输入 USB 数据流调整内部振荡器，这使内部振荡器（和 4 倍时钟乘法器）能满足 USB 时钟的精度要求。时钟恢复功能应在下面的配置中使用。

通信速度	USB 时钟	4 倍时钟乘法器
全速	4 倍时钟乘法器	内部振荡器
低速	内部振荡器/2	N/A

当 USB0 工作在低速方式并使用时钟恢复功能时，软件必须向 CRLOW 写‘1’，以使能低速时钟恢复功能。在低速方式一般不需要使用时钟恢复功能。

当 USB 网络上有强噪声时，可以使用单步方式帮助时钟恢复电路锁定。在一般的 USB 环境中不需要（或不建议）使用该方式。

USB 寄存器定义 12.5 CLKREC：时钟恢复控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
CRE	CRSSEN	CRLOW			保留			00001001
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x0F
位 7： CRE：时钟恢复使能 该位使能/禁止 USB 时钟恢复功能。 0：禁止时钟恢复。 1：使能时钟恢复。								
位 6： CRSSEN：时钟恢复单步 该位在时钟恢复期间强制采用“单步”方式进行振荡器校准。 0：正常校准方式。 1：单步方式。								
位 5： CRLOW：低速时钟恢复方式 如果在 USB0 工作于低速方式时使用时钟恢复功能，则该位必须被置‘1’。 0：全速方式。 1：低速方式。								
位 4-0： 保留。读 = 可变，必须写 01001b。								

12.5 FIFO 管理

256 字节的片内 XRAM 被用作 USB0 的 FIFO 空间。FIFO 空间在端点 0 和端点 1 之间的分配如图 12.3 所示。分配给端点 1 的 FIFO 空间被分割为一个输入(IN)端点和一个输出(OUT)端点。

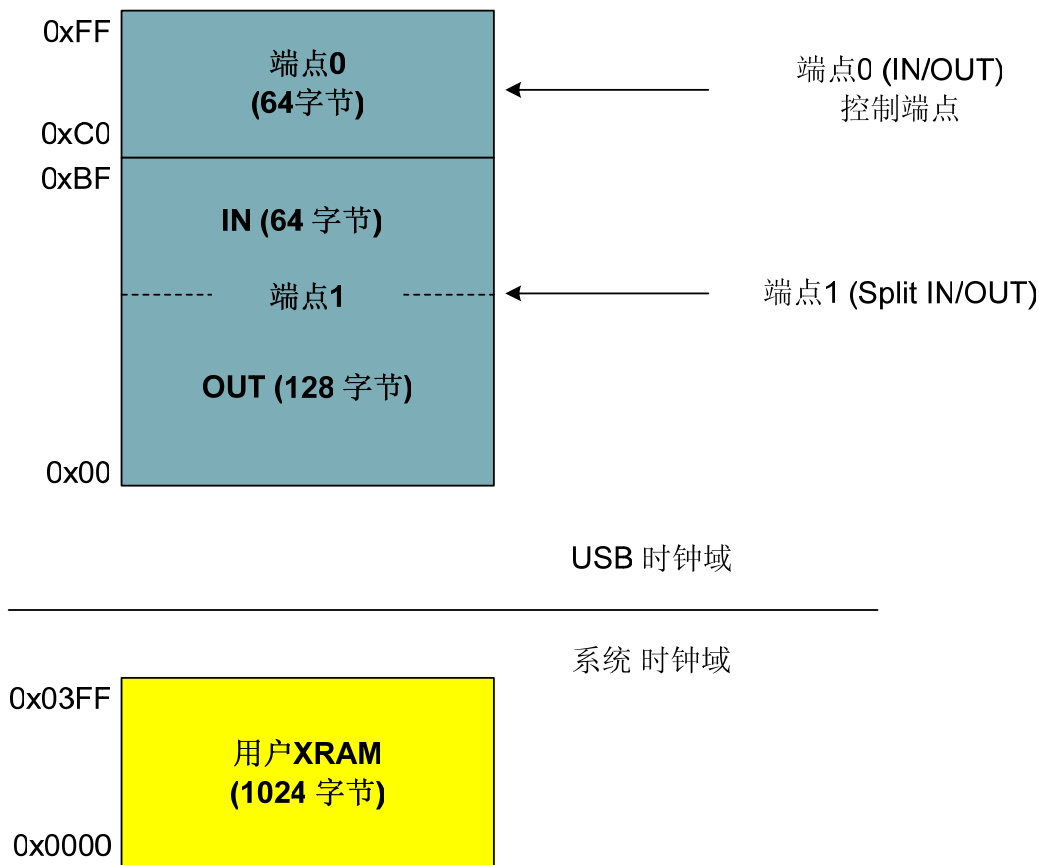


图 12.3 USB FIFO 分配

12.5.1 FIFO 分割方式

端点 1 的 FIFO 空间被分割成两半，高 64 字节 FIFO 空间由输入 (IN) 端点使用，低 128 字节 FIFO 空间由输出 (OUT) 端点使用。

端点 0 的 FIFO 空间是不分割的，64 字节 FIFO 空间组成单个 IN FIFO 或 OUT FIFO。端点 0 在某一时刻只能在一个方向传输数据。端点方向 (IN/OUT) 由端点的 EINC_SRH 寄存器 (见图 12.20) 中的 DIRSEL 位决定。

12.5.2 FIFO 双缓冲

端点 1 的 FIFO 空间可以被配置为双缓冲模式。在该模式，最大数据包的字节数为 FIFO 空间的一半，FIFO 中可以同时有两个数据包。该模式只适用于端点 1。输入（IN）端点和/或输出（OUT）端点都可以被配置为双缓冲模式。表 12.3 列出了每种 FIFO 配置下的最大数据包的字节数。

表 12.3 FIFO 配置

端点号	分割方式使能?	最大输入数据包字节数 (双缓冲禁止/使能)	最大输出数据包字节数 (双缓冲禁止/使能)
0	N/A	64	
	Y	64/32	128/64

12.5.3 FIFO 访问

对每个端点 FIFO 的访问是通过对应的 FIFO_n 寄存器来实现的。对一个端点 FIFO_n 寄存器的读操作将数据从该端点的 FIFO 中卸载一个字节；对一个端点 FIFO_n 寄存器的写操作将一个字节加载到该端点的 FIFO。当一个端点 FIFO 被配置为分割方式时，对端点 FIFO_n 寄存器的读操作从输出端点 FIFO 中卸载一个字节，对端点 FIFO_n 寄存器的写操作将一个字节加载到输入端点 FIFO。

USB 寄存器定义 12.6 FIFO_n: USB0 端点 FIFO 访问寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
FIFODATA								00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x20-0x21
使用 USB 地址 0x20 – 0x21 访问 2 对端点 FIFO。								
输入/输出端点 FIFO				USB 地址				
0				0x20				
1				0x21				
对 FIFO 地址的写操作将数据加载到相应端点的 IN FIFO。								
对 FIFO 地址的读操作将数据从相应端点的 OUT FIFO 中卸载。								

12.6 功能地址

FADDR 寄存器中保存有当前的 USB0 的功能地址。USB0 接收到的 SET_ADDRESS 命令中有 7 位功能地址，软件应在接收到该命令时将主机分配的 7 位功能地址写入 FADDR 寄存器。只有在当前传输结束后（一般是在 SET_ADDRESS 命令传输的状态段），写入 FADDR 的新地址才生效（生效之前 USB0 不响应新地址）。当软件向 FADDR 寄存器写入新地址时，UPDATE 位（FADDR.7）被硬件置 ‘1’。当新地址生效后，硬件将 UPDATE 位清 ‘0’。

USB 寄存器定义 12.7 FADDR：USB0 功能地址寄存器

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
UPDATE	功能地址							00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x00
位 7:	UPDATE：功能地址更新							
	当软件写 FADDR 寄存器时被硬件置 ‘1’。当新地址生效后，硬件将该位清 ‘0’。							
	0：最后写入 FADDR 的地址已经生效。							
	1：最后写入 FADDR 的地址尚未生效。							
位 6-0:	功能地址							
	保持 USB0 的 7 位功能地址。当端点 0 收到 SET_ADDRESS 标准设备请求时，软件应写该地址。设备请求完成后新地址生效。							

12.7 功能配置和控制

USB 寄存器 POWER（见图 12.8）用于在设备级控制 USB0（使能/禁止，复位/挂起/恢复控制等）。

USB 复位：当检测到总线上的复位信号时，USBRST 位（POWER.3）被硬件置‘1’。检测到复位信号时会发生以下过程：

1. USB0 地址被复位（FADDR = 0x00）。
2. 端点 FIFO 被清空。
3. 控制/状态寄存器被复位为 0x00（E0CSR、EINCSRL、EINCSRH、EOUTCSRL、EOUTCSRH）。
4. USB 寄存器 INDEX 被复位为 0x00。
5. 所有 USB 中断（挂起中断除外）都被使能，其中断标志被清除。
6. 产生 USB 复位中断（如果被使能）。

向 USBRST 位写‘1’将产生异步 USB0 复位。在发生异步复位后，所有 USB 寄存器复位到它们的缺省值。

挂起方式：在挂起检测被使能的情况下（SUSEN = ‘1’），当总线上检测到挂起信号时，USB0 进入挂起方式，并产生中断（如果中断被使能，即 SUSINTE = ‘1’）。挂起中断服务程序（ISR）应执行与应用相关的配置任务，如禁止相应的外设和/或配置低速方式的时钟源。有关内部振荡器配置（包括内部振荡器的挂起方式功能）的详细信息见“10. 振荡器”。

当有下面任何一种情况发生时，USB0 退出挂起方式：1）检测到或产生恢复信号，2）检测到复位信号，3）发生设备或 USB 复位。如果 USB0 被挂起，则在发生上述事件时内部振荡器退出挂起方式。

恢复信号：当总线上检测到恢复信号时，USB0 退出挂起方式，并产生恢复中断（如果中断被使能，即 RESINTE = ‘1’）。软件可以通过向 RESUME 位（POWER.2）写‘1’来强制产生远程唤醒。当强制产生远程唤醒时，软件应在远程唤醒被启动（RESUME = ‘1’）后 10 – 15ms 向 RESUME 位写‘0’，以结束恢复信号。

ISO 更新：当软件向 ISOUP 位（POWER.7）写‘1’后，ISO 更新功能被使能。在 ISO 更新功能被使能时，写入到一个 ISO IN 端点的新数据包要等到收到一个新的帧起始（SOF）信号后才被发送。如果 ISO 输入端点在收到 SOF 之前收到一个 IN 令牌，USB0 将发送一个长度为零的数据包。当 ISOUP = ‘1’时，所有 ISO 端点的 ISO 更新功能都被使能。

USB 使能：在上电复位（POR）后 USB0 被禁止。通过清除 USBINH 位（POWER.4）来使能 USB0。USBINH 一旦被写入‘0’，则只能在发生下列事件之一时被置‘1’：1）上电复位（POR），2）向 USBRST 位（POWER.3）位写‘1’导致的异步 USB0 复位。

软件应在使能 USB0 之前执行所有的 USB0 配置。建议的配置步骤如下：

1. 选择并使能 USB 时钟源。
2. 通过向 USBRST 位写‘1’来复位 USB0。
3. 配置并使能 USB 收发器。
4. 执行任何一种 USB0 功能配置（中断、挂起检测）。
5. 通过向 USBINH 位写‘0’来使能 USB0。

C8051F326/7 全速 USB、16KB FLASH 微控制器

USB 寄存器 12.8 POWER: USB0 电源控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	复位值
ISOUD	-	-	USBINH	USBRST	RESUME	SUSMD	SUSEN	00010000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x01
位 7:	ISOUD: ISO 更新 该位影响所有 IN 同步端点。 0: 当软件向 INPRDY 写 1 时, USB0 在收到下一个 IN 令牌后发送数据包。 1: 当软件向 INPRDY 写 1 时, USB0 等待 SOF 令牌, 然后发送数据包。如果在 SOF 令牌之前收到 IN 令牌, 则 USB0 将发送长度为 0 的数据包。							
位 6-5:	未使用。读 = 00b。写 = 忽略。							
位 4:	USBINH: USB0 禁止 该位在发生上电复位 (POR) 或异步 USB0 复位 (见位 3: USBRST) 后被置 1。 软件应在完成所有 USB0 和收发器的初始化后将该位清 0。软件不能将该位置 1。 0: 使能 USB。 1: 禁止 USB。所有 USB 通信都被忽略。							
位 3:	USBRST: 复位检测 向该位写 1 强制产生异步 USB0 复位。读该位返回总线复位状态信息。 读: 0: 总线上未检测到复位信号。 1: 总线上检测到复位信号。							
位 2:	RESUME: 强制恢复 软件可以强制在总线上产生恢复信号, 将 USB0 从挂起方式唤醒。在挂起方式 (SUSMD =1) 向该位写 1 将强制 USB0 在总线上产生恢复信号 (一个远程唤醒事件)。软件应在 10 – 15ms 后向 RESUME 位写 0, 以结束恢复信号。软件向 RESUME 位写 0 后将产生中断, 并且硬件将 SUSMD 清 0。							
位 1:	SUSMD: 挂起方式 当 USB0 进入挂起方式时, 该位被硬件置 1。当软件向 RESUME 位写 0 (后面跟随一次远程唤醒) 或在检测到总线上的恢复信号后读 CMINT 寄存器时硬件将该位清 0。 0: USB0 不处于挂起模式。 1: USB0 处于挂起模式。							
位 0:	SUSEN: 挂起检测使能 0: 禁止挂起检测。USB0 将忽略总线上的挂起信号。 1: 使能挂起检测。如果检测到总线上的挂起信号, 则 USB0 进入挂起方式。							

C8051F326/7 全速 USB、16KB FLASH 微控制器

USB 寄存器定义 12.9 FRAMEL: USB0 帧号低字节

R	R	R	R	R	R	R	R	复位值
帧号低字节								00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x0C
位 7-0: 帧号低字节								
该寄存器包含最后接收帧号的位 7-0。								

USB 寄存器定义 12.10 FRAMEH: USB0 帧号高字节

R	R	R	R	R	R	R	R	复位值
-	-	-	-	-	帧号高位			00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x0D
位 7-3: 未使用。读 = 00000b。写 = 忽略。								
位 2-0: 帧号高字节								
该寄存器包含最后接收帧号的位 10-8。								

12.8 中断

USB0 的中断标志（只读）位于图 12.11 ~ 12.13 所示的 USB 寄存器中。相应的中断使能位位于图 12.14 ~ 12.16 所示的 USB 寄存器中。当这些 USB 中断标志中的任何一个被置 ‘1’ 时，就会产生 USB0 中断。通过特殊功能寄存器 EIE1 来使能 USB0 中断（见“6.3 中断系统”）。

注意：读一个中断标志寄存器将使该寄存器中的所有标志位复位为 ‘0’。

USB 寄存器定义 12.11 IN1INT：USB0 输入端点中断寄存器

R	R	R	R	R	R	R	R	复位值
—	—	—	—	—	—	IN1	EP0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x02
位 7-2： 未使用。读 = 000000b。写 = 忽略。								
位 1： IN1： IN 端点端点 1 中断标志								
该位在软件读 IN1INT 寄存器后被清 0。								
0： IN 端点 1 中断无效。								
1： IN 端点 1 中断有效。								
位 0： EP0： 端点 0 中断标志								
该位在软件读 IN1INT 寄存器后被清 0。								
0： 端点 0 中断无效。								
1： 端点 0 中断有效。								

USB 寄存器定义 12.12 OUT1INT：USB0 输出端点中断寄存器

R	R	R	R	R	R	R	R	复位值
—	—	—	—	—	—	OUT1	—	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x04
位 7-2： 未使用。读 = 000000b。写 = 忽略。								
位 1： OUT1： OUT 端点 1 中断标志								
该位在软件读 OUT1INT 寄存器后被清 0。								
0： OUT 端点 1 中断无效。								
1： OUT 端点 1 中断有效。								
位 0： 未使用。读 = 0b。写 = 忽略。								

C8051F326/7 全速 USB、16KB FLASH 微控制器

USB 寄存器定义 12.13 CMINT: USB0 公共中断寄存器

R	R	R	R	R	R	R	R	复位值
—	—	—	—	SOF	RSTINT	RSUINT	SUSINT	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x06
位 7-4: 未使用。读 = 0000b。写 = 忽略。								
位 3: SOF: 帧起始中断标志								
该位在收到 SOF 令牌时被硬件置 ‘1’。当硬件期望收到一个 SOF 事件时, 就会产生中断, 即使实际的 SOF 信号丢失或损坏。								
该位在软件读 CMINT 寄存器后被清 ‘0’。								
0: 帧起始中断无效。								
1: 帧起始中断有效。								
位 2: RSTINT: 复位中断标志								
当总线上检测到复位信号时该位被硬件置 ‘1’。								
该位在软件读 CMINT 寄存器后被清 ‘0’。								
0: 复位中断无效。								
1: 复位中断有效。								
位 1: RSUINT: 恢复中断标志								
在 USB0 处于挂起方式期间, 当总线上检测到恢复信号时该位被硬件置 ‘1’。								
该位在软件读 CMINT 寄存器后被清 ‘0’。								
0: 恢复中断无效。								
1: 恢复中断有效。								
位 0: SUSINT: 挂起中断标志								
在挂起检测被使能的情况下 (寄存器 POWER 中的 SUSEN 位), 当总线上检测到挂起信号时该位被硬件置 ‘1’。								
该位在软件读 CMINT 寄存器后被清 ‘0’。								
0: 挂起中断无效。								
1: 挂起中断有效。								

USB 寄存器定义 12.14 IN1IE: USB0 输入端点中断允许寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	—	—	—	—	—	IN1E	EP0E	00000011
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x07
位 7-2: 未使用。读 = 000000b。写 = 忽略。								
位 1: IN1E: IN 端点 1 中断允许								
0: 禁止 IN 端点 1 中断。								
1: 允许 IN 端点 1 中断。								
位 0: EP0E: 端点 0 中断允许								
0: 禁止端点 0 中断。								
1: 允许端点 0 中断。								

C8051F326/7 全速 USB、16KB FLASH 微控制器

USB 寄存器定义 12.15 OUTIE: USB0 输出端点中断允许寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	—	—	—	—	—	OUTIE	—	00001110
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x09
位 7-2: 未使用。读 = 000000b。写 = 忽略。								
位 1: OUT1: OUT 端点 1 中断允许								
0: 禁止 OUT 端点 1 中断。								
1: 允许 OUT 端点 1 中断。								
位 0: 未使用。读 = 0b。写 = 忽略。								

USB 寄存器定义 12.16 CMIE: USB0 公共中断允许寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
—	—	—	—	SOFE	RSTINTE	RSUINTE	SUSINTE	00000110
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x0B
位 7-4: 未使用。读 = 0000b。写 = 忽略。								
位 3: SOFE: 帧起始中断允许								
0: 禁止帧起始中断。								
1: 允许帧起始中断。								
位 2: RSTINTE: 复位中断允许								
0: 禁止复位中断。								
1: 允许复位中断。								
位 1: RSUINTE: 恢复中断允许								
0: 禁止恢复中断。								
1: 允许恢复中断。								
位 0: SUSINTE: 挂起中断允许								
0: 禁止挂起中断。								
1: 允许挂起中断。								

12.9 串行接口引擎

串行接口引擎 (SIE) 执行所有低层 USB 协议，在数据被成功发送或接收后中断处理器。当接收数据时，SIE 在接收完一个完整的数据包后中断处理器；相应的握手信号由 SIE 自动产生。当发送数据时，SIE 在发送完一个完整的数据包并且收到相应的握手信号后中断处理器。

当收到损坏/错误的数据包时，SIE 不中断处理器。

12.10 端点 0

通过 USB 寄存器 E0CSR (见图 12.17) 管理端点 0。访问 E0CSR 寄存器时，INDEX 寄存器必须被装入 0x00 值。

当发生下列情况时产生端点 0 中断：

1. 当收到一个数据包 (OUT 或 SETUP) 并将其装入端点 0 FIFO 时。OPRDY 位 (E0CSR.0) 被硬件置 ‘1’。
2. 当一个 IN 数据包从端点 0 FIFO 成功卸载并发送到主机时。INPRDY 位被硬件清 ‘0’。
3. 一次 IN 传输结束 (该中断在数据交换的状态阶段产生)。
4. 在控制传输因违反协议而结束后，硬件将 STSTL 位 (E0CSR.2) 置 ‘1’。
5. 因控制传输在固件置 ‘1’ DATAEND 位 (E0CSR.3) 之前结束，导致硬件将 SUEND 位 (E0CSR.4) 置 ‘1’。

E0CNT 寄存器 (见图 12.21) 保持端点 0 FIFO 中的接收数据字节数。

硬件将自动检测协议错误并发送 STALL 条件作为应答。固件可以强制产生一个 STALL 条件以放弃当前的传输。当产生 STALL 条件时，STSTL 位被置 ‘1’ 并产生中断。下述情况可以导致硬件产生 STALL 条件。

1. 当 DATAEND 位被置 ‘1’ 后，主机在 OUT 数据阶段发送一个 OUT 令牌。
2. 当 DATAEND 位被置 ‘1’ 后，主机在 IN 数据阶段发送一个 IN 令牌。
3. 主机发送一个超过端点 0 最大数据包大小的数据包。
4. 主机在一次 IN 传输的状态段发送一个非零长度的 DATA1 数据包。

固件将 SDSTL 位 (E0CSR.5) 置 ‘1’。

12.10.1 端点 0 SETUP 传输

所有控制传输都必须从一个 SETUP 包开始。SETUP 包与 OUT 包类似，包含一个由主机发送的 8 字节数据域。包含一个非 8 字节命令域的任何 SETUP 包都会被 USB0 自动丢弃。当 SETUP 包的数据被装入到端点 0 FIFO 时会产生端点 0 中断。软件应从端点 0 FIFO 卸载命令，对命令译码，执行所要求的任务，并将 SOPRDY 位置 ‘1’ 以指示执行完 OUT 包服务。

12.10.2 端点 0 IN 传输

当 USB0 收到一个要求发送数据到主机的 SETUP 请求时，主机会发送一个或多个 IN 请求。对于第一个 IN 事务，固件应将一个 IN 数据包装入到端点 0 FIFO 并将 INPRDY 位（E0CSR.1）置‘1’。IN 数据包被成功发送后会产生中断。注意：如果在固件将一个数据包装入到端点 0 FIFO 之前收到 IN 请求，则不会产生中断。如果所请求的数据超过了端点 0 最大数据包的大小（已报告给主机），则数据应被分成多个包；除最后一个数据包外，每个包应为最大数据包长度。如果所请求的数据是端点 0 最大数据包长度的整数倍，则最后一个数据包应为零长度包，指示传输结束。固件在将一次传输的最后一个数据包装入端点 0 FIFO 之后，应将 DATAEND 位置‘1’。

对于特定的控制传输，在收到第一个 IN 令牌后，端点 0 处于发送方式。在该方式，主机应向端点 0 只发送 IN 令牌。在端点 0 处于发送方式时，如果收到一个 SETUP 或 OUT 令牌，则 SUEND 位（E0CSR.4）被置‘1’。

在发生下列事件之前，端点 0 将一直保持在发送方式：

1. USB0 接收一个端点 0 SETUP 或 OUT 令牌。
2. 固件发送一个小于端点 0 最大数据包长度的数据包。
3. 固件发送一个零长度的数据包。

当执行上述的(2)和(3)时，固件应将 DATAEND 位（E0CSR.3）置‘1’。

如果在 IN FIFO 中没有准备好的数据包（INPRDY = ‘0’），则 SIE 会发送一个 NAK 作为对 IN 令牌的应答。

12.10.3 端点 0 OUT 传输

当 USB0 收到一个要求主机发送数据到 USB0 的 SETUP 请求时，主机会发送一个或多个 OUT 请求。当 USB0 成功接收到一个 OUT 数据包时，硬件将 OPRDY 位（E0CSR.0）置‘1’并产生端点 0 中断。在发生该中断后，固件应将 OUT 数据包从端点 0 FIFO 卸载并将 SOPRDY 位（E0CSR.6）置‘1’。

如果所请求的数据超过了端点 0 最大数据包长度，则数据应被分成多个包。如果所请求的数据是端点 0 最大数据包长度的整数倍（已报告给主机），则主机发送的最后一个数据包应为零长度包，以指示传输结束。

对于特定的控制传输，在收到第一个 OUT 令牌后，端点 0 处于接收方式。在该方式，主机应向端点 0 只发送 OUT 令牌。在端点 0 处于接收方式时，如果收到一个 SETUP 或 IN 令牌，则 SUEND 位（E0CSR.4）被置‘1’。

在发生下列事件之前，端点 0 将一直保持在接收方式：

1. SIE 接收一个 SETUP 或 IN 令牌。
2. 主机发送一个小于端点 0 最大数据包长度的数据包。
3. 主机发送一个零长度的数据包。

当接收到所期望数量的数据时，固件应将 DATAEND 位（E0CSR.3）置‘1’。如果主机在固件将 DATAEND 位置‘1’之后发送一个 OUT 包，则 SIE 将发送一个 STALL 条件。在 STALL 条件被发送后，STSTL 位（E0CSR.2）被置‘1’并产生端点 0 中断。

C8051F326/7 全速 USB、16KB FLASH 微控制器

USB 寄存器定义 12.17 E0CSR: USB0 端点 0 控制寄存器

R/W	R/W	R/W	R	R/W	R/W	R/W	R	复位值
SSUEND	SOPRDY	SDSTL	SUEND	DATAEND	STSTL	INPRDY	OPRDY	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x11
位 7:	SSUEND: SETUP 结束事件服务位 写: 软件应在处理完 SETUP 结束事件 (SUEND) 后将该位置 '1'。当软件向 SSUEND 写 '1' 后, 硬件将 SUEND 位清 '0'。 读: 该位的读出值总是为 '0'。							
位 6:	SOPRDY: OPRDY 服务位 写: 软件应在处理完接收到的端点 0 数据包后将该位置 '1'。当软件向 SOPRDY 写 '1' 后, 硬件将 OPRDY 位清 '0'。 读: 该位的读出值总是为 '0'。							
位 5:	SDSTL: 发送 STALL 软件可以通过向该位写 '1' 来结束当前的数据传输 (因为错误条件、不希望的传输请求等)。当 STALL 信号被发送后, 硬件将该位清 '0'。							
位 4:	SUEND: SETUP 结束标志 当一次控制传输在软件向 DATAEND 位写 '1' 之前结束时, 硬件将该只读位置 '1'。当软件向 SSUEND 写 '1' 后, 硬件将该位清 '0'。							
位 3:	DATAEND: 数据结束 软件应在下列情况向该位写 '1'。 1. 当发送最后一个数据包后固件向 INPRDY 写 '1' 时。 2. 当发送一个零长度数据包后固件向 INPRDY 写 '1' 时。 3. 当接收完最后一个数据包后固件向 SOPRDY 写 '1' 时。 该位被硬件自动清 '0'。							
位 2:	STSTL: 已发送 STALL 发送 STALL 信号后, 硬件将该位置 '1'。该位必须用软件清 '0'。							
位 1:	INPRDY: 输入 (IN) 数据包准备好 软件应在将一个要发送的数据包装入端点 0 FIFO 后向该位写 '1'。在发生下列条件之一时硬件将该位清 '0'。 1. 数据包已发送。 2. 数据包被一个输入 SETUP 包覆盖。 3. 数据包被一个输入 OUT 包覆盖。							
位 0:	OPRDY: 输出 (OUT) 数据包准备好 当收到一个数据包时, 硬件将该只读位置 '1' 并产生中断。该位只在软件向 SOPRDY 位写 '1' 时被清 '0'。							

USB 寄存器定义 12.18 E0CNT：USB0 端点 0 数据计数寄存器

R	R	R	R	R	R	R	R	复位值
—	E0CNT							00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址： 0x16
位 7： 未使用。读 = 0b。写 = 忽略。 位 6-0： E0CNT：端点 0 数据计数 该 7 位值指示端点 0 FIFO 中接收的数据字节数。该值只在 OPRDY 位为 ‘1’ 期间有效。								

12.11 配置端点 1

对端点 1 的配置和控制是通过一组控制/状态寄存器来实现的，这些寄存器是：IN 寄存器 EINCSSL、EINCSRH 和 OUT 寄存器 EOUTCSRL、EOUTCSRH。端点控制/状态寄存器被映射到 USB 寄存器地址空间，由 INDEX 寄存器定义（见图 12.4）。

12.12 控制端点 1 输入（IN）

对端点 1 输入（IN）的管理是通过 USB 寄存器 EINCSSL 和 EINCSRH 来实现的。IN 端点可以被用于中断、批量或同步传输。通过向寄存器 EINCSRH 中的 ISO 位写 ‘1’ 来使能同步（ISO）方式。硬件对批量和中断传输的控制与此完全相同。

在发生下述条件之一时，会产生端点 1 输入（IN）中断：

1. 当一个 IN 数据包被成功发送到主机时。
2. 当目标 FIFO 未空，而软件向 FLUSH 位（EINCSSL.3）写 ‘1’ 时。
3. 硬件产生一个 STALL 条件。

12.12.1 端点 1 输入（IN）中断或批量方式

当 ISO 位（EINCSRH.6）为逻辑 0 时，端点 1 工作在批量或中断方式。一旦一个端点被配置为批量/中断 IN 方式（通常在端点 0 的 SET_INTERFACE 命令之后），固件应将一个 IN 数据包装入端点的 IN FIFO 并将 INPRDY 位（EINCSSL.0）位置 ‘1’。在收到一个 IN 令牌后，硬件将发送数据，清除 INPRDY 位，并产生中断。

在没有向端点 FIFO 写任何数据的情况下向 INPRDY 写 ‘1’ 会导致在收到下一个 IN 令牌时发送一个零长度的数据包。

可以通过向 SDSTL 位（EINCSSL.4）写 ‘1’ 来停止批量或中断管道。当 SDSTL = ‘1’ 时，硬件对所有的 IN 请求都以 STALL 条件回应。每当硬件产生一个 STALL 条件时，会产生一个中断，且 STSTL 位（EINCSSL.5）被置 ‘1’。STSTL 位必须用固件清 ‘0’。

当端点 FIFO 中有一个打开的数据包 SLOT 时，硬件自动将 INPRDY 位清 ‘0’。注意：如果目标端点被使能为双缓冲模式，则固件可以在同一时间向 IN FIFO 装入两个数据包。在这种情况下，在固件将第一个数据包装入 FIFO 并将 INPRDY 位置 ‘1’ 后，硬件会立即清 ‘0’ INPRDY 位。在这种情况下没有中断产生，只在数据包已被发送时产生中断。

C8051F326/7 全速 USB、16KB FLASH 微控制器

当固件向 FCDT 位 (EINCSR.H.3) 写 ‘1’ 时, 每个 IN 数据包的会不停地 Toggle, 与从主机接收到的握手信号无关。该特性一般在中断端点中使用, 为同步端点提供速率反馈通信。当 FCDT= ‘0’ 时, 数据 toggle 位只在主机发送了 ACK (对一个 IN 数据包应答) 后被 toggle。

12.12.2 端点 1 输入 (IN) 同步方式

当 ISO 位 (EINCSR.H.6) 被置 ‘1’ 时, 目标端点工作在同步 (ISO) 方式。一旦一个端点被配置为 ISO 输入 (IN) 方式, 主机将每帧发送一个 IN 令牌 (数据请求); 数据在每帧中的位置可能不同。因此建议将 ISO IN 端点使能为双缓冲方式。

当端点 FIFO 中有一个打开的数据包 SLOT 时, 硬件自动将 INPRDY 位 (EINCSRL.0) 清 ‘0’。注意: 如果目标端点被使能为双缓冲模式, 则固件可以在同一时间向 IN FIFO 装入两个数据包。在这种情况下, 在固件将第一个数据包装入 FIFO 并将 INPRDY 位置 ‘1’ 后, 硬件会立即清 ‘0’ INPRDY 位。在这种情况下没有中断产生, 只在数据包已被发送时产生中断。

如果在 USB0 从主机接收一个 IN 令牌时, 在端点 FIFO 中没有已准备好的数据包, 则 USB0 会发送一个零长度的数据包并将 UNDRUN 位 (EINCSRL.2) 置 ‘1’

ISO 更新功能 (见 12.7 节) 在开始一个双缓冲的 ISO IN 端点时可能有用。如果在固件向端点 FIFO 写第一个数据包时主机已经建立了 ISO IN 管道 (已经开始发送 IN Token), 则在固件向 FIFO 写完第二个 (双缓冲) 数据包之前, 下一个 IN 令牌可以到达, 第一个数据包已经发送。ISO 更新功能保证写到端点 FIFO 中的任何数据包不在当前帧发送, 只在接收到 SOF 信号后发送该数据包。

C8051F326/7 全速 USB、16KB FLASH 微控制器

USB 寄存器定义 12.19 EINCSSL: USB0 IN 端点控制寄存器低字节

R	W	R/W	R/W	W	R/W	R/W	R/W	复位值
—	CLRDT	STSTL	SDSTL	FLUSH	UNDRUN	FIFONE	INPRDY	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x11
位 7:	未使用。读 = 0b, 写 = 忽略。							
位 6:	CLRDT: 数据 Toggle 清除位 写: 软件应向该位写 ‘1’, 以将 IN 端点数据 toggle 复位为 ‘0’。 读: 该位的读出值总是为 ‘0’。							
位 5:	STSTL: 已发送 STALL 当 STALL 信号被发送后, 硬件将该位置 ‘1’。FIFO 被清空, INPRDY 位被清 ‘0’。 该标志必须用软件清 ‘0’。							
位 4:	SDSTL: 发送 STALL 软件应向该位写 ‘1’ 以产生 STALL 信号作为对一个 IN 令牌的应答。软件应向 该位写 ‘0’ 以结束 STALL 信号。该位对 ISO 方式没有影响。							
位 3:	FLUSH: FIFO 清空 向该位写 ‘1’ 将从 IN 端点 FIFO 中清除待发送的下一个数据包。FIFO 指针被复 位, INPRDY 位被清除。如果 FIFO 中包含多个数据包, 软件必须对每个数据包 向 FLUSH 写 ‘1’。当 FIFO 清空完成后, 硬件将 FLUSH 位清 ‘0’。							
位 2:	UNDRUN: 数据不足 该位的功能取决于 IN 端点的方式: ISO: 在 INPRDY = ‘0’ 并且收到一个 IN 令牌后发送了一个零长度数据包时, 该位被置 ‘1’。 中断/批量: 当作为对一个 IN 令牌的应答而返回一个 NAK 时该位被置 ‘1’。 该位必须用软件清 ‘0’。							
位 1:	FIFONE: FIFO 非空 0: IN 端点 FIFO 为空。 1: IN 端点 FIFO 包含一个或多个数据包。							
位 0:	INPRDY: IN 数据包准备好 软件应在将一个数据包装入 IN 端点 FIFO 后向该位写 ‘1’。在发生下列条件之一 时硬件将该位清 ‘0’。 1. 数据包已发送。 2. 双缓冲被使能 (DBIEN = ‘1’) 且有一个打开的 FIFO 数据包 SLOT。 3. 如果端点处于同步方式并且 ISOUD = ‘1’, 在收到下一个 SOF 之前 INPRDY 的读出值总是为 0。 当硬件因发送数据包而将 INPRDY 清 ‘0’ 时会产生中断 (如果被允许)。							

C8051F326/7 全速 USB、16KB FLASH 微控制器

USB 寄存器定义 12.20 EINCSRH: USB0 IN 端点控制寄存器高字节

R/W	W	R	R	R/W	R	R	R	复位值
DBIEN	ISO	—	—	FCDT	—	—	—	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x12
位 7:	DBIEN: IN 端点双缓冲使能 0: 禁止所选 IN 端点的双缓冲功能。 1: 使能所选 IN 端点的双缓冲功能。							
位 6:	ISO: 同步传输使能 该位使能/禁止在当前端点进行同步传输 0: 端点被配置为批量/中断传输。 1: 端点被配置为同步传输。							
位 5-4:	未使用。读 = 00b, 写 = 忽略。							
位 3:	FCDT: 强制数据 Toggle。 0: 端点数据 toggle 只在发送完一个数据包后收到 ACK 时切换。 1: 端点数据 toggle 在每发送完一个数据包后被强制切换, 不管是否收到 ACK。							
位 2:	SPLIT: FIFO 分割使能 当 SPLIT = ‘1’ 时, 所选端点的 FIFO 被分割。所选 FIFO 的高半部分为 IN 端点使用, 所选 FIFO 的低半部分为 OUT 端点使用。							
位-0:	未使用。读 = 000b, 写 = 忽略。							

12.13 控制端点 1 输出 (OUT)

对端点 1 输出 (OUT) 的管理是通过 USB 寄存器 EOUTCSRL 和 EOUTCSRH 来实现的。OUT 端点可以被用于中断、批量或同步传输。通过向寄存器 EOUTCSRH 中的 ISO 位写 ‘1’ 来使能同步 (ISO) 方式。硬件对批量和中断传输的控制与此完全相同。

在发生下述条件之一时, 会产生端点 1 输出 (OUT) 中断:

1. 硬件将 OPRDY 位 (EOUTCSRL.0) 置 ‘1’。
2. 硬件产生一个 STALL 条件。

12.13.1 端点 1 输出 (OUT) 中断或批量方式

当 ISO 位 (EOUTCSRH.6) 为逻辑 0 时, 目标端点工作在批量或中断方式。一旦一个端点被配置为批量/中断输出 (OUT) 方式 (通常在端点 0 的 SET_INTERFACE 命令之后), 硬件将 OPRDY 位 (EOUTCSRL.0) 位置 ‘1’ 并在收到一个 OUT 令牌和数据包后产生中断。当前 OUT 数据包 (已准备好从 FIFO 卸载的数据包) 中的字节数由寄存器 EOUTCNTH 和 EOUTCNTL 给出。响应该中断时, 固件应将数据包从 OUT FIFO 中卸载并将 OPRDY 位清 ‘0’。

可以通过向 SDSTL 位 (EOUTCSRL.5) 写 ‘1’ 来停止批量或中断管道。当 SDSTL = ‘1’ 时, 硬件对所有的 OUT 请求都以 STALL 条件回应。每当硬件产生一个 STALL 条件时, 会产生一个中断, 且 STSTL 位 (EOUTCSRL.6) 被置 ‘1’。STSTL 位必须用固件清 ‘0’。

当 OUT FIFO 中有一个准备好的数据包时, 硬件自动将 OPRDY 位置 ‘1’。注意: 如果端点 1 被使能为双缓冲模式, 则 OUT FIFO 中可以同时有两个数据包准备好。在这种情况下, 在固件将第一个数据包卸载并将 OPRDY 位清 ‘0’ 后, 硬件会立即将 OPRDY 位置 ‘1’。在这种情况下会产生第二个中断。

12.13.2 端点 1 输出 (OUT) 同步方式

当 ISO 位 (EOUTCSRH.6) 被置 ‘1’ 时, 目标端点工作在同步 (ISO) 方式。一旦一个端点被配置为 ISO OUT 方式, 主机将每帧发送一个数据包; 但数据包在每帧中的位置可能不同。因此建议将 ISO OUT 端点使能为双缓冲方式。

每次收到一个数据包时, 硬件将接收到的数据包装入端点 FIFO, 将 OPRDY 位 (EOUTCSRL.0) 置 ‘1’, 并产生中断 (如果被允许)。固件一般用该中断将数据包从端点 FIFO 卸载, 并将 OPRDY 位清 ‘0’。

如果在端点 FIFO 没有空间的情况下收到一个数据包, 则会产生中断, 并且 OVRUN 位 (EOUTCSRL.2) 被置 ‘1’。如果 USB0 接收到一个有 CRC 错误的 ISO 数据包, 则该数据包被装入端点 FIFO, OPRDY 被置 ‘1’, 产生中断 (如果被允许), DATAERR 位 (EOUTCSRL.3) 被置 ‘1’。软件应在每次将数据包从 ISO OUT 端点 FIFO 卸载后检查 DATAERR 位。

C8051F326/7 全速 USB、16KB FLASH 微控制器

USB 寄存器定义 12.21 EOUTCSRL: USB0 OUT 端点控制寄存器低字节

W	R/W	R/W	W	R	R/W	R	R/W	复位值
CLRDT	STSTL	SDSTL	FLUSH	DATERR	OVRUN	FIFOFUL	OPRDY	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x14
位 7:	CLRDT: 数据 Toggle 清除位 写: 软件应向该位写 ‘1’, 以将 OUT 端点数据 toggle 复位为 ‘0’。 读: 该位的读出值总是为 ‘0’。							
位 6:	STSTL: 已发送 STALL 当 STALL 信号被发送后, 硬件将该位置 ‘1’。该标志必须用软件清 ‘0’。							
位 5:	SDSTL: 发送 STALL 软件应向该位写 ‘1’ 以产生 STALL 信号。软件应向该位写 ‘0’ 以结束 STALL 信号。该位对 ISO 方式没有影响。							
位 4:	FLUSH: FIFO 清空 向该位写 ‘1’ 将从 OUT 端点 FIFO 中清除下一个数据包。FIFO 指针被复位, OPRDY 位被清除。如果 FIFO 中包含多个数据包, 软件必须对每个数据包向 FLUSH 写 ‘1’。当 FIFO 清空完成后, 硬件将 FLUSH 位清 ‘0’。							
位 3:	DATERR: 数据错误 在 ISO 方式, 如果接收到的数据包有 CRC 或位填充错误, 该位被硬件置 ‘1’。当软件清除 OPRDY 时, 该位被清 ‘0’。该位只在 ISO 方式有效。							
位 2:	OVRUN: 数据溢出 当一个输入数据包不能被装入到 OUT 端点 FIFO 时, 该位被硬件置 ‘1’。该位只在 ISO 方式有效。该位必须用软件清 ‘0’。 0: 无数据溢出。 1: 自该标志最后一次被清除以来, 因 FIFO 已满导致数据包丢失。							
位 1:	FIFOFUL: OUT FIFO 满标志 该位指示 OUT FIFO 的内容。如果端点被使能为双缓冲模式 (DBIEN = ‘1’), 当 FIFO 中有两个数据包时 FIFO 已满。如果 DBIEN = ‘0’), 当 FIFO 中有一个数据包时 FIFO 已满。 0: OUT 端点 FIFO 未滿。 1: OUT 端点 FIFO 已滿。							
位 0:	OPRDY: 输出 (OUT) 数据包准备好 当有数据包可用时硬件将该位置 ‘1’ 并产生中断。软件应在将每个数据包从 OUT 端点 FIFO 卸载后将该位清 ‘0’。							

C8051F326/7 全速 USB、16KB FLASH 微控制器

USB 寄存器定义 12.22 EOUTCSRH: USB0 OUT 端点控制寄存器高字节

R/W	R/W	R	R	R	R	R	R	复位值
DBOEN	ISO	—	—	—	—	—	—	00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x15
位 7: DBOEN: OUT 端点双缓冲使能 0: 禁止所选 OUT 端点的双缓冲功能。 1: 使能所选 OUT 端点的双缓冲功能。								
位 6: ISO: 同步传输使能 该位使能/禁止在当前端点进行同步传输 0: 端点被配置为批量/中断传输。 1: 端点被配置为同步传输。								
位 5-0: 未使用。读 = 000000b, 写 = 忽略。								

USB 寄存器定义 12.23 EOUTCNTRL: USB0 OUT 端点计数器寄存器低字节

R	R	R	R	R	R	R	R	复位值
EOCL								00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x16
位 7-0: EOCL: OUT 端点计数器低字节 EOCL 保持在当前 OUT 端点 FIFO 中的最后一个接收数据包之 10 位数据字节数的低 8 位。该数值只在 OPRDY = ‘1’ 时有效。								

USB 寄存器定义 12.24 EOUTCNTH: USB0 OUT 端点计数器寄存器高字节

R	R	R	R	R	R	R	R	复位值
—	—	—	—	—	—	EOCH		00000000
位7	位6	位5	位4	位3	位2	位1	位0	USB地址: 0x17
位 7-2: 未使用。读 = 000000b, 写 = 忽略。								
位 1-0: EOCH: OUT 端点计数器高字节 EOCH 保持在当前 OUT 端点 FIFO 中的最后一个接收数据包之 10 位数据字节数的高 2 位。该数值只在 OPRDY = ‘1’ 时有效。								

C8051F326/7 全速 USB、16KB FLASH 微控制器

表 12.4 USB 收发器电气特性

VDD=3.0 ~ 3.6V, -40°C到+85°C (除非特别说明)

参 数	符号	条 件	最小值	典型值	最大值	单 位
发送器						
输出高电平	V _{OH}		2.8	—	—	V
输出低电平	V _{OL}		—	—	0.8	V
输出交叉点	V _{CRS}		1.3	—	2.0	V
输出电阻	Z _{DRV}	驱动高电平 驱动低电平	—	38 38	—	Ω
上拉电阻	R _{PU}	全速 (D+上拉) 低速 (D-上拉)	1.425	1.5	1.575	kΩ
输出上升时间	T _R	低速 全速	75 4	— —	300 20	ns
输出下降时间	T _F	低速 全速	75 4	— —	300 20	ns
接收器						
差分输入敏感度	V _{DI}	(D+) - (D-)	0.2	—	—	V
差分输入共模范围	V _{CM}		0.8	—	2.5	V
输入漏电流	I _L	上拉电阻禁止	—	<1.0	—	μA
注： 时序图和符号定义请参见 USB 规范。						

13. UART0

UART0 是一个异步、全双工串口，它提供多种数据格式选项。UART0 具有一个专门的波特率发生器，包括一个 16 位定时器和可选的预分频器，能产生很宽范围的波特率（详见“13.1 波特率发生器”）。接收数据 FIFO 允许接收多达三个字节而不会发生数据丢失和溢出。。

UART0 有 6 个相关的特殊功能寄存器。三个用于波特率发生器（SBCON0、SBRLH0 和 SBRLLO），两个用于数据格式、控制和状态功能（SCON0、SMOD0），一个用于发送和接收数据（SBUF0）。用同一个 SBUF0 地址可以访问发送寄存器和接收寄存器。**写 SBUF0 时总是访问发送寄存器；读 SBUF0 时总是访问接收寄存器，不可能从发送数据寄存器中读取数据。**

如果 UART0 中断被允许，则每次发送完成（SCON0 中的 TI0 位被置‘1’）或接收到数据字节（SCON0 中的 RI0 位被置‘1’）时将产生中断。当 CPU 转向中断服务程序时硬件不清除 UART0 中断标志。中断标志必须用软件清除，这就允许软件查询 UART0 中断的原因（发送完成或接收完成）。如果接受 FIFO 中还有未读取的字节，则 RI0 位不能被软件清 0。

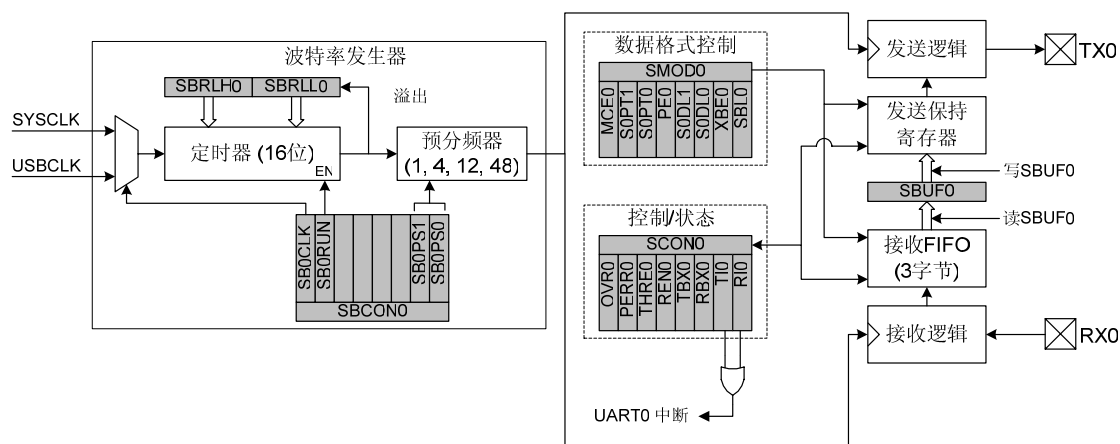


图 13.1 UART0 原理框图

13.1 波特率发生器

UART0 波特率由一个专用的 16 位定时器产生。该定时器可以用控制器核时钟(SYSCLK)或 USB 时钟(USBCLK)工作,并具有一个能产生 1、4、12 或 48 分频的预分频器。定时器和预分频器选项的组合允许使用很多时钟频率,因此波特率的选择范围很宽。

波特率发生器通过三个寄存器来配置:SBCON0、SBRLH0 和 SBRLLO。UART0 波特率发生器控制寄存器(SBCON0, SFR 定义 13.4)使能或禁止波特率发生器,选择波特率发生器的时钟源,选择定时器的预分频值。使用 UART0 时,波特率发生器必须被使能。寄存器 SBRLH0 和 SBRLLO 保持专用 16 位定时器的重载值。内部定时器从重载值开始向上计数,每个时钟加 1。在定时器溢出时(从 0xFFFF 到 0x0000),定时器被重载。UART0 的波特率由方程 13.1 定义,其中“BRG Clock”为所选择的波特率发生器时钟源。对于可靠的 UART 操作,UART 波特率不应被配置为大于 SYSCLK/16。

方程 13.1 UART0 波特率

$$\text{波特率} = \frac{\text{BRG Clock}}{(65536 - \text{SBRLH0} : \text{SBRLLO})} \times \frac{1}{2} \times \frac{1}{\text{Prescaler}}$$

表 13.1 给出了典型波特率和时钟频率的快速参考。

C8051F326/7 全速 USB、16KB FLASH 微控制器

表 13.1 对应标准波特率的波特率发生器设置

	目标波特率 (bps)	实际波特率 (bps)	波特率 误差	振荡器 分频系数	SB1PS[1:0] (预分频位)	SBRLH1:SBRL1 中的重载值
SYSCLK = 12 MHz	230400	230769	0.16%	52	11	0xFFE6
	115200	115385	0.16%	104	11	0xFFCC
	57600	57692	0.16%	208	11	0xFF98
	28800	28846	0.16%	416	11	0xFF30
	14400	14388	0.08%	834	11	0xFF5F
	9600	9600	0.0%	1250	11	0xFD8F
	2400	2400	0.0%	5000	11	0xF63C
	1200	1200	0.0%	10000	11	0xEC78
SYSCLK = 24 MHz	230400	230769	0.16%	104	11	0xFFCC
	115200	115385	0.16%	208	11	0xFF98
	57600	57692	0.16%	416	11	0xFF30
	28800	28777	0.08%	834	11	0xFF5F
	14400	14406	0.04%	1666	11	0xFCBF
	9600	9600	0.0%	2500	11	0xFB1E
	2400	2400	0.0%	10000	11	0xEC78
	1200	1200	0.0%	20000	11	0xD8F0
SYSCLK = 48 MHz	230400	230769	0.16%	208	11	0xFF98
	115200	115385	0.16%	416	11	0xFF30
	57600	57692	0.08%	834	11	0xFF5F
	28800	28846	0.04%	1666	11	0xFCBF
	14400	14423	0.02%	3334	11	0xF97D
	9600	9600	0.0%	5000	11	0xF63C
	2400	2400	0.0%	20000	11	0xD8F0
	1200	1200	0.0%	40000	11	0xB1E0

13.2 数据格式

UART0 提供多种数据格式选项。数据传输以起始位（逻辑低电平）开始，后面是数据位（LSB 在先）、一个奇偶位或额外位（如果选择），最后由一个或两个停止位（逻辑高电平）结束。数据长度可以在 5 ~ 8 位。一个奇偶位可以加到数据之后，由硬件自动产生和检测（奇、偶、传号、空号）。停止位长度可以选择一位或两位。UART0 还具有多机通信模式。所有数据格式选项都可以用 SMOD0 寄存器配置，如 SFR 定义 13.2 所示。图 13.2 示出了不使用奇偶位或额外位的 UART0 数据传输时序。图 13.3 示出了使能奇偶位（PE0 = 1）时的 UART0 数据传输时序。图 13.4 示出了使能额外位（XBE0 = 1）时的 UART0 数据传输时序。注意：当奇偶位被使能时，额外位功能不可用。只有在数据长度为 6、7 或 8 位时，才能选择使用第二停止位。

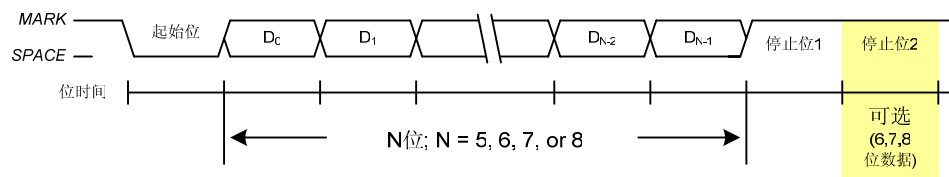


图 13.2 没有奇偶位或额外位的 UART0 时序

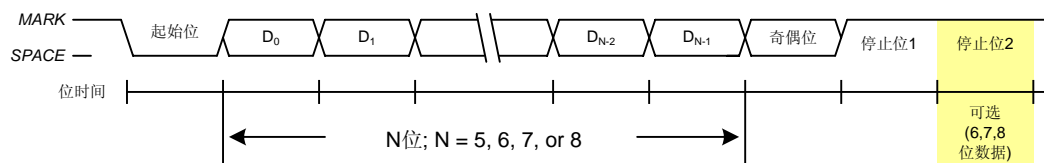


图 13.3 有奇偶位时的 UART0 时序

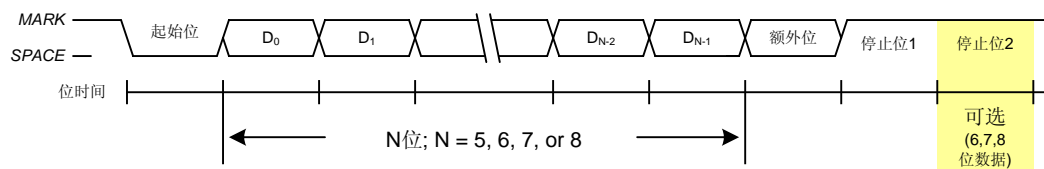


图 13.4 有额外位时的 UART0 时序

13.3 配置和操作

UART0 提供标准的异步、全双工通信。它可以工作在点对点串行方式，也可以作为多处理器串行接口的一个节点。当工作在点对点方式，即串行总线上只有两个器件时，SMOD0 中的 MCE0 位应被清 0。当作为多处理器通信总线上的一个器件工作时，MCE0 位和 XBE0 应被置 1。在上述两种工作方式，数据都是从 TX0 引脚发送，在 RX0 引脚接收。使用交叉开关和端口 I/O 寄存器配置 TX0 和 RX0 引脚，详见“11. 端口输入输出”

在典型的 UART 通信中，一个器件的发送（TX）输出被连接到其它器件的接收（RX）输入，可以直接连接，也可以通过总线收发器，如图 13.5 所示。

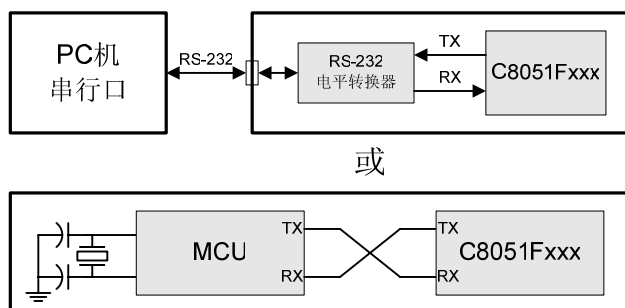


图 13.5 典型 UART 连接图

13.3.1 数据发送

当软件向 SBUF0 寄存器写入一个字节时开始数据发送。在发送结束时（停止位开始）发送中断标志 TI0（SCON0.1）被置‘1’。如果中断被使能，则在 TI0 置位时会产生中断。

如果额外位功能被使能（XBE0 = 1），且奇偶位功能被禁止（PE0 = 0），则 TBX0（SCON0.3）位将被发送（在额外位的位置）。当奇偶位功能被使能（PE0 = 1）时，硬件会根据所选择的奇偶位类型（用 S0PT[1:0]选择）产生奇偶位，并将其加到数据域之后。注意：当奇偶位被使能时，额外位功能不可用。

13.3.2 数据接收

在接收允许位 REN0（SCON0.4）被置‘1’后，数据接收可以在任何时刻开始。收到停止位后，如果满足下述条件则数据字节将被装入到接收 FIFO：接收 FIFO 必须未满（3 字节深度）；停止位必须为 1。在接收 FIFO 已满的情况下，接收的字节被丢弃，并会产生接收 FIFO 溢出错误（寄存器 SCON0 中的 OVR0 被置 1）。如果停止位为逻辑 0，则接收数据不会被保存到接收 FIFO 中。如果接收条件满足，则数据被保存到接收 FIFO 中，且 RI0 标志被置 1。注意：当 MCE0 = 1 时，只有在额外位也等于 1 时 RI0 才会被置 1。可以通过读 SBUF0 寄存器从接收 FIFO 中读取数据。SBUF0 寄存器中保存的是 FIFO 中最老的数据。在 SBUF0 被读取后，FIFO 中的下一个字节被装入到 SBUF0 中，FIFO 中空出的位置可以接收一个新字节。如果中断被使能，则在 RI0 置位时会产生中断。

C8051F326/7 全速 USB、16KB FLASH 微控制器

如果额外位功能被使能 ($XBE0 = 1$)，且奇偶位功能被禁止 ($PE0 = 0$)，则 FIFO 中最老字节的额外位可以从 RBX0 位 (SCON0.2) 读出。如果额外位功能未被使能，则 RBX0 代表 FIFO 中最老字节的停止位。如果奇偶位功能被使能 ($PE0 = 1$)，硬件会在接收数据时根据所选择的奇偶位类型 (用 S0PT[1:0] 选择) 检查接收到的停止位。如果接收到的字节具有奇偶错误，则 PERR0 标志被置 1。该标志必须用软件清 0。注意：当奇偶位被使能时，额外位功能不可用。

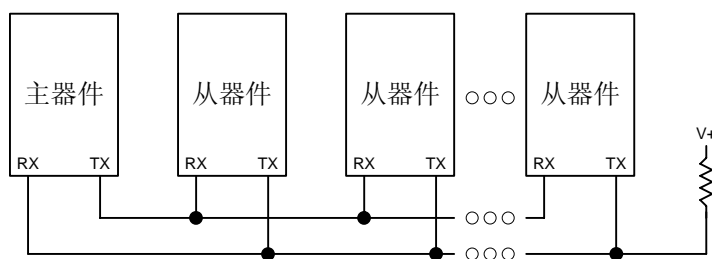
13.3.3 多机通信

UART0 通过使用额外数据位支持一个主处理器与一个或多个从处理器之间的多机通信。当主机要发送数据给一个或多个从机时，它先发送一个用于选择目标的地址字节。地址字节与数据字节的区别是额外位为逻辑 1；数据字节的额外位总是设置为逻辑 0。

如果从机的 MCE0 位 (SMOD0.7) 被置 1，则只有当 UART 接收到的额外位为逻辑 1 (RBX0 = 1) 并收到有效的停止位后 UART 才会产生中断，表示收到一个地址字节。在 UART 的中断处理程序中，软件将接收到的地址与从机自身的地址进行比较。如果地址匹配，从机将清除它的 MCE0 位以允许后面接收数据字节时产生中断。未被寻址的从机仍保持其 MCE0 位为 1，在收到后续的数据字节时不产生中断，从而忽略收到的数据。一旦接收完整个消息，被寻址的从机将它的 MCE0 位重新置 1，以忽略所有的数据传输，直到它收到下一个地址字节。

可以将多个地址分配给一个从机，或将一个地址分配给多个从机，从而允许同时向多个从机“广播”发送。主机可以被配置为接收所有的传输数据，或通过实现某种协议使主/从角色能临时变换以允许原来的主机和从机之间进行半双工通信。

图 13.6 UART 多机方式连接图



C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 13.1 SCON0: UART0 控制寄存器

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	复位值
OVR0	PERR0	—	REN0	TBX0	RBX0	TI0	RI0	00100000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: (可位寻址) 0x98
<div>位 7: OVR0: 接收 FIFO 溢出标志 该位用于指示接收 FIFO 溢出状态。 0: 未发生接收 FIFO 溢出。 1: 发生了接收 FIFO 溢出 (由于 FIFO 已满, 新接收的字符被丢弃)。 该位必须用软件清 0。</div>								
<div>位 6: PERR0: 奇偶错误标志 当奇偶位被使能时, 该位用于指示是否发生了奇偶错误。当 FIFO 中的最老字节的奇偶位与所选择的奇偶类型不匹配时, 该位被置 1。 0: 未发生奇偶错误。 1: 发生了奇偶错误。 该位必须用软件清 0。</div>								
<div>位 5: 未使用。读 = 1b。写 = 忽略。</div>								
<div>位 4: REN0: 接收允许 该位允许/禁止 UART 接收器。接收被禁止时仍可从接收 FIFO 中读数据。 0: UART0 接收禁止。 1: UART0 接收允许。</div>								
<div>位 3: TBX0: 额外发送位 当 XBE0 被置 1 时, 该位的逻辑电平被赋值给额外发送位。当奇偶位功能被使能时, 该位未使用。</div>								
<div>位 2: RBX0: 额外接收位 当 XBE0 被置 1 时, RBX0 被赋予额外位的值。当 XBE0 被清 0 时, RBX0 被赋予第一个停止位的值。当奇偶位功能被使能时, 该位无效。</div>								
<div>位 1: TI0: 发送中断标志 在数据被发送之后、停止位开始时该位被硬件置 1。当 UART0 中断被允许时, 置 1 该位将导致 CPU 转到 UART0 中断服务程序。该位必须用软件清 0。</div>								
<div>位 0: RI0: 接收中断标志 当 UART0 接收到一个字节数据时该位被硬件置 1 (在停止位的采样时刻)。当 UART0 中断被允许时, 置 1 该位将会使 CPU 转到 UART0 中断服务程序。该位必须用软件清 0。</div>								

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 13.2 SMOD0: UART0 方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
MCE0	S0PT1	S0PT0	PE0	S0DL1	S0DL0	XBE0	SBL0	00001100
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x9A
<p>位 7: MCE0: 多处理器通信使能 0: RI 在停止位为 1 时置位。 1: RI 在停止位和额外位均为 1 置位 (必须用 XBE0 使能额外位)。 注意: 当硬件奇偶位功能被使能时, 该位不可用。</p>								
<p>位 6-5: S0PT1[1:0]: 奇偶位类型 00: 奇校验。 01: 偶校验。 10: 传号。 11: 空号。</p>								
<p>位 4: PE0: 奇偶位使能 该位使能/禁止硬件奇偶位产生和检查。当奇偶位功能被使能时, 奇偶位类型由 S0PT1~0 位选择。 0: 硬件奇偶位禁止。 1: 硬件奇偶位使能。</p>								
<p>位 3-2: S0DL[1:0]: 数据长度 00: 5 位数据 01: 6 位数据 10: 7 位数据 11: 8 位数据</p>								
<p>位 1: XBE0: 额外位使能 当被使能时, TBX0 的值被加到数据域之后。 0: 额外位禁止。 1: 额外位使能。</p>								
<p>位 0: SBL0: 停止位长度 0: 短 — 停止位有效时间为一个位时间 (所有数据域长度)。 1: 长 — 停止位有效时间为两个位时间 (数据长度 = 6、7 或 8 位)。</p>								

SFR 定义 13.3 SBUF0: UART0 串行数据缓冲寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x99
<p>位 7-0: SBUF0[7:0]: UART0 数据缓冲器位 7-0 (MSB-LSB) 该 SFR 用于从 UART 发送数据和从 UART0 接收 FIFO 接收数据。 写: 当数据被写到 SBUF0 时, 它进入发送移位寄存器等待串行发送。向 SBUF0 写入一个字节即启动发送过程。 读: 读 SBUF0 时从接收 FIFO 中提取数据。读 SBUF0 时, 返回 FIFO 中最老的字节, 该字节被从 FIFO 中清除。FIFO 中最多可以保持 3 个字节。如果 FIFO 中还有数据字节可读, 则 RI0 位将保持在逻辑 1 状态, 不能被软件清除。</p>								

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 13.4 SBCON0: UART0 波特率发生器控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
SB0CLK	SB0RUN	保留	保留	保留	保留	SB0PS1	SB0PS0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x91
<p>位 7: SB0CLK: 波特率发生器时钟源选择 0: SYSCLK 用作波特率发生器时钟源。 1: USBCLK 用作波特率发生器时钟源。</p> <p>位 6: SB0RUN: 波特率发生器使能。 0: 波特率发生器禁止。UART0 不能工作。 1: 波特率发生器使能。</p> <p>位 5-2: 未使用。读 = 0000b。必须写 0000b。</p> <p>位 1-0: SB0PS[1:0]: 波特率预分频选择 00: 预分频 = 12。 01: 预分频 = 4。 10: 预分频 = 48。 11: 预分频 = 1。</p>								

SFR 定义 13.5 SBRLH0: UART0 波特率发生器高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x94
<p>位 7-0: SBRLH0.[7:0]: UART0 波特率发生器重载值高字节</p>								

SFR 定义 13.6 SBRL0: UART0 波特率发生器低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x93
<p>位 7-0: SBRL0.[7:0]: UART0 波特率发生器重载值低字节</p>								

14. 定时器

C8051F326/7 内部有两个 16 位计数器/定时器，与标准 8051 中的计数器/定时器兼容。这些定时器可以用于测量时间间隔和产生周期性的中断请求。定时器 0 和定时器 1 几乎完全相同，有四种工作方式。

定时器 0 和定时器 1 工作方式
13 位定时器
16 位定时器
8 位自动重载的定时器
两个 8 位定时器（仅限于定时器 0）

定时器 0 和定时器 1 有 5 个可选择的时钟源，由定时器时钟选择位（T1M-T0M）和时钟分频位（SCA1-SCA0）决定。时钟分频位定义一个预分频时钟，作为定时器 0 和/或定时器 1 的时钟源（见图 14.3）。定时器 0 和定时器 1 可被配置为使用这个预分频时钟或系统时钟。

定时器 0 和定时器 1 有门控方式，允许定时器只在一个外部中断有效（/INT0 对应定时器 0，/INT1 对应定时器 1）期间运行。该方式方便了对输入到 P0.2 的脉冲宽度的测量（定时器 0）和使用定时器 1 对低频振荡器进行校准。

14.1 定时器 0 和定时器 1 工作方式

每个定时器都是一个 16 位的寄存器，在被访问时以两个字节的形式出现：一个低字节（TL0 或 TL1）和一个高字节（TH0 或 TH1）。定时器控制寄存器（TCON）用于使能定时器 0 和定时器 1 以及指示它们的状态。通过将 IE 寄存器中的 ET0 位置‘1’来允许定时器 0 中断，通过将 ET1 位置‘1’来允许定时器 1 中断。这两个定时器都有四种工作方式，通过设置定时器方式寄存器（TMOD）中的方式选择位 T1M1-T0M0 来选择工作方式，每个定时器都可以被独立配置。下面对每种工作方式进行详细说明。

14.1.1 方式 0 — 13 位计数器/定时器

在方式 0，定时器 0 和定时器 1 被作为 13 位的定时器使用。图 14.1 给出了定时器 0 工作在方式 0 时的原理框图。下面介绍对定时器 0 的配置和操作。由于这两个定时器在工作上完全相同，定时器 1 的配置过程与定时器 0 一样。

TH0 寄存器保持 13 位定时器的 8 个 MSB，TL0 在 TL0.4-TL0.0 位置保持 5 个 LSB。TL0 的高 3 位（TL0.7-TL0.5）是不确定的，在读计数值时应屏蔽掉或忽略这 3 位。作为 13 位定时器寄存器，计到 0x1FFF（全 1）后再计一次将发生溢出，使计数值回到 0x0000，此时定时器溢出标志 TF0（TCON.5）被置位并产生中断（如果该中断被允许）。

当 GATE0（TMOD.3）为逻辑 0，或 GATE0 为逻辑 1 并且输入信号/INT0 有效时，置 1 TR0 位（TCON.4）将允许定时器 0 工作。设置 GATE0 为逻辑 1 允许定时器受外部输入信号/INT0 的控制，便于脉冲宽度测量。当 GATE0 被设置为逻辑 1 时，/INT0 输入引脚为 P0.2。

C8051F326/7 全速 USB、16KB FLASH 微控制器

TR0	GATE0	/INT0	计数器/定时器
0	X	X	禁止
1	0	X	允许
1	1	0 (P0.2 高)	禁止
1	1	0 (P0.2 低)	允许

X=任意

注意，置位 TR0 并不强制定时器 0 复位。应在定时器被允许前将定时器寄存器装入所需要的初值。

与上述的 TL0 和 TH0 一样，TL1 和 TH1 构成定时器 1 的 13 位寄存器。定时器 1 的配置和控制方法与定时器 0 一样，使用 TCON 和 TMOD 中的对应位。输入信号/INT1 为定时器 1 所用。有关/INT0 和/INT1 的详细说明见“8.3.2 外部中断”。

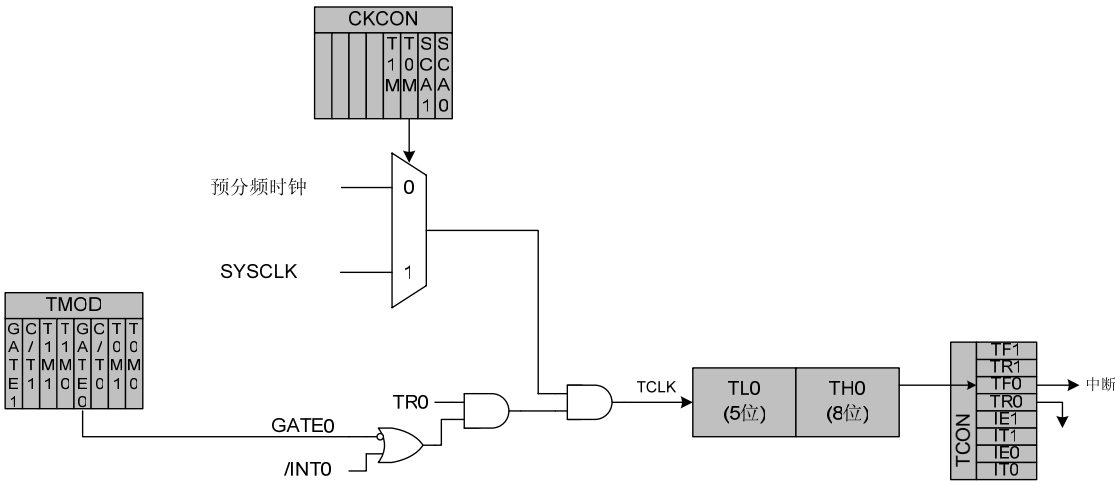


图 14.1 T0 方式 0 原理框图

14.1.2 方式 1 — 16 位计数器/定时器

方式 1 的操作与方式 0 完全一样，所不同的是定时器使用全部 16 位。用与方式 0 相同的方法允许和控制工作在方式 1 的定时器。

14.1.3 方式 2 — 自动重载的 8 位计数器/定时器

方式 2 将定时器 0 和定时器 1 配置为具有自动重新装入计数初值能力的 8 位定时器。TL0 保持计数值，而 TH0 保持重载值。当 TL0 中的计数值发生溢出（从全 ‘1’ 到 0x00）时，定时器溢出标志 TF0（TCON.5）被置位，TH0 中的重载值被重新装入到 TL0。如果中断被允许，在 TF0 被置位时将产生一个中断。TH0 中的重载值保持不变。为了保证第一次计数正确，必须在允许定时器之前将 TL0 初始化为所希望的计数初值。当工作于方式 2 时，定时器 1 的操作与定时器 0 完全相同。

在方式 2，定时器 1 和定时器 0 的配置和控制方法与方式 0 一样。当 GATE0（TMOD.3）为逻辑 ‘0’，或 GATE0 为逻辑 1 且输入信号/INT0 有效时，置 ‘1’ TR0 位（TCON.4）将允许定时器 0 工作。有关/INT0 和/INT1 的详细说明见“8.3.2 外部中断”。

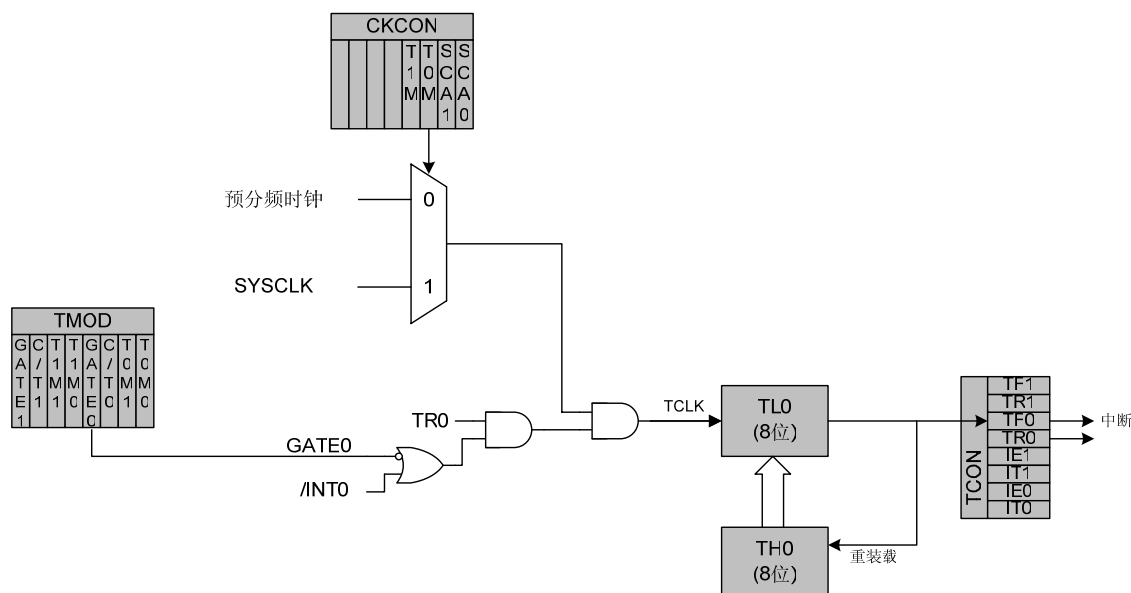


图 14.2 T0 方式 2 原理框图

14.1.4 方式 3 — 两个 8 位计数器/定时器（仅定时器 0）

在方式 3 时，定时器 0 被配置两个独立的 8 位定时器，计数值分别在 TL0 和 TH0 中。在 TL0 中的定时器使用 TCON 和 TMOD 中定时器 0 的控制/状态位：TR0、C/T0、GATE0 和 TF0。TL0 既可以使用系统时钟也可以使用一个外部输入信号作为时基。TH0 寄存器只能作为定时器使用，由系统时钟或分频时钟提供时基。TH0 使用定时器 1 的运行控制位 TR1，并在发生溢出时将定时器 1 的溢出标志位 TF1 置 ‘1’，所以它控制定时器 1 的中断。

定时器 1 在方式 3 时停止运行。在定时器 0 工作于方式 3 时，定时器 1 可以工作在方式 0、1 或 2，但不能设置 TF1 标志和产生中断。但是定时器 1 溢出可以用于为 **UART 产生波特率**。当定时器 0 工作在方式 3 时，定时器 1 的运行控制由其方式设置决定。为了在定时器 0 工作于方式 3 时使用定时器 1，应使定时器 1 工作在方式 0、1 或 2。可以通过将定时器 1 切换到方式 3 使其停止运行。

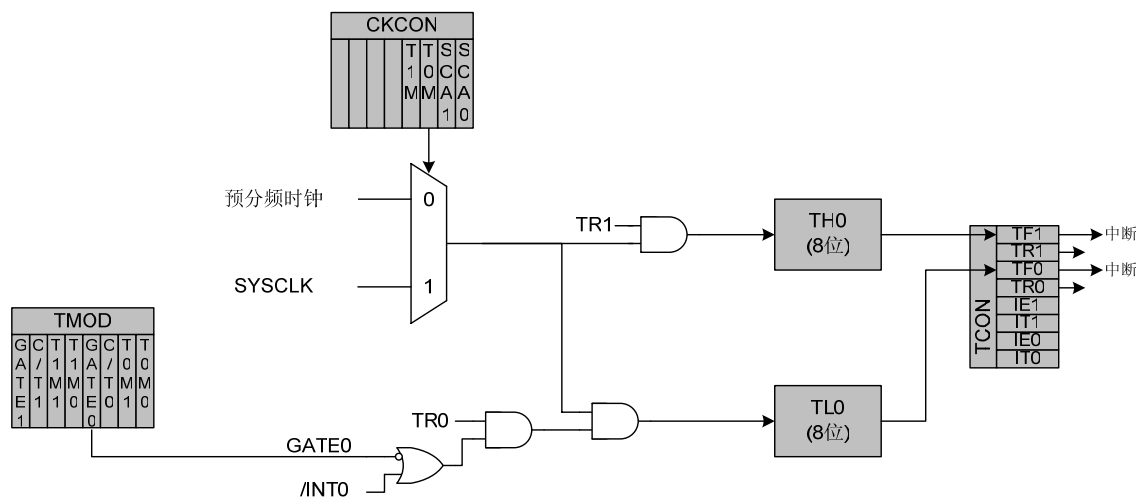


图 14.3 T0 方式 3 原理框图

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 14.1 TCON: 定时器控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00001010
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: (可位寻址) 0x88
位 7:	TF1: 定时器 1 溢出标志 当定时器 1 溢出时由硬件置位。该位可以用软件清 0, 但当 CPU 转向定时器 1 中断服务程序时该位被自动清 0。 0: 未检测到定时器 1 溢出。 1: 定时器 1 发生溢出。							
位 6:	TR1: 定时器 1 运行控制 0: 定时器 1 禁止。 1: 定时器 1 允许。							
位 5:	TF0: 定时器 0 溢出标志 当定时器 0 溢出时由硬件置位。该位可以用软件清 0, 但当 CPU 转向定时器 0 中断服务程序时该位被自动清 0。 0: 未检测到定时器 0 溢出。 1: 定时器 0 发生溢出。							
位 4:	TR0: 定时器 0 运行控制 0: 定时器 0 禁止。 1: 定时器 0 允许。							
位 3:	IE1: 外部中断 1 当检测到一个由 IT1 定义的边沿/电平时, 该标志由硬件置位。该位可以用软件清 0, 但当 CPU 转向外部中断 1 中断服务程序时该位被自动清 0(如果 IT1=1)。 当 IT1=0 时, 该标志在/INT1 有效时被置 ‘1’。							
位 2:	IT1: 中断 1 类型选择 该位选择/INT1 中断是边沿触发还是电平触发。 0: /INT1 为电平触发。 1: /INT1 为边沿触发。							
位 1:	IE0: 外部中断 0 当检测到一个由 IT0 定义的边沿/电平时, 该标志由硬件置位。该位可以用软件清 0, 但当 CPU 转向外部中断 0 中断服务程序时该位被自动清 0(如果 IT0=1)。 当 IT0=0 时, 该标志在/INT0 有效时被置 ‘1’。							
位 0:	IT0: 中断 0 类型选择 该位选择/INT0 中断是边沿触发还是电平触发。 0: /INT0 为电平触发。 1: /INT0 为边沿触发。							

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 14.2 TMOD：定时器方式寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
GATE1	保留	T1M1	T1M0	GATE0	保留	T0M1	T0M0	00000000
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x89

位 7: GATE1: 定时器 1 门控位

0: 当 TR1=1 时定时器 1 工作, 与/INT1 的逻辑电平无关。/INT1 在内部振荡器从挂起状态恢复时有效。

1: 只有当 TR1=1 并且/INT1 有效时定时器 1 才工作。/INT1 在每两个低频振荡器时钟周期被激活一次。这个激活频率为 40KHz。

位 6: 保留。读 = 0b。必须写 0b。

位 5-4: T1M1-T1M0: 定时器 1 方式选择

这些位选择定时器 1 的工作方式。

T1M1	T1M0	方式
0	0	方式 0: 13 位定时器
0	1	方式 1: 16 位定时器
1	0	方式 2: 自动重装载的 8 位定时器
1	1	方式 3: 定时器 1 停止运行

位 3: GATE0: 定时器 0 门控位

0: 当 TR0=1 时定时器 0 工作, 与/INT0 的逻辑电平无关。/INT0 输入引脚为 P0.2。

1: 只有当 TR0=1 并且/INT0 有效时定时器 0 才工作。/INT0 输入引脚为 P0.2。

位 2: 保留。读 = 0b。必须写 0b。

位 1-0: T0M1-T0M0: 定时器 0 方式选择

这些位选择定时器 0 的工作方式。

T0M1	T0M0	方式
0	0	方式 0: 13 位定时器
0	1	方式 1: 16 位定时器
1	0	方式 2: 自动重装载的 8 位定时器
1	1	方式 3: 两个 8 位定时器

SFR 定义 14.3 CKCON: 时钟控制寄存器

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值															
—	—	—	—	T1M	T0M	SCA1	SCA0	00000000															
位7	位6	位5	位4	位3	位2	位1	位0	SFR地址: 0x8E															
位 7-4: 未使用。读 = 0000b。写 = 忽略。																							
位 3: T1M: 定时器 1 时钟选择。 该位选择定时器 1 的时钟源。 0: 定时器 1 使用由分频位（SCA1-SAC0）定义的时钟。 1: 定时器 1 使用系统时钟																							
位 2: T0M: 定时器 0 时钟选择。 该位选择定时器 0 的时钟源。 0: 定时器 0 使用由分频位（SCA1-SAC0）定义的时钟。 1: 定时器 0 使用系统时钟																							
位 1-0: SCA1-SCA0: 定时器 0/1 分频位 如果定时器 0/1 被配置为使用分频时钟，则这些位控制时钟分频数。																							
<table><tr><th>SCA1</th><th>SCA0</th><th>分频时钟</th></tr><tr><td>0</td><td>0</td><td>系统时钟/12</td></tr><tr><td>0</td><td>1</td><td>系统时钟/4</td></tr><tr><td>1</td><td>0</td><td>系统时钟/48</td></tr><tr><td>1</td><td>1</td><td>外部时钟/8</td></tr></table>									SCA1	SCA0	分频时钟	0	0	系统时钟/12	0	1	系统时钟/4	1	0	系统时钟/48	1	1	外部时钟/8
SCA1	SCA0	分频时钟																					
0	0	系统时钟/12																					
0	1	系统时钟/4																					
1	0	系统时钟/48																					
1	1	外部时钟/8																					
注：外部时钟 8 分频与系统时钟同步。																							

C8051F326/7 全速 USB、16KB FLASH 微控制器

SFR 定义 14.4 TL0: 定时器 0 低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址: 0x8A
位 7-0: TL0: 定时器 0 低字节								
TL0 寄存器是 16 位定时器 0 的低字节。								

SFR 定义 14.5 TL1: 定时器 1 低字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址: 0x8B
位 7-0: TL1: 定时器 1 低字节								
TL1 寄存器是 16 位定时器 1 的低字节。								

SFR 定义 14.6 TH0: 定时器 0 高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址: 0x8C
位 7-0: TH0: 定时器 0 高字节								
TH0 寄存器是 16 位定时器 0 的高字节。								

SFR 定义 14.7 TH1: 定时器 1 高字节

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	复位值
位7	位6	位5	位4	位3	位2	位1	位0	00000000
								SFR地址: 0x8D
位 7-0: TH1: 定时器 1 高字节								
TH1 寄存器是 16 位定时器 1 的高字节。								

C8051F326/7 全速 USB、16KB FLASH 微控制器

15. C2 接口

C8051F326/7 内部有一个 Silicon Labs 2 线（C2）调试接口，支持 FLASH 编程和使用安装在最终应用系统中的器件进行在系统调试。C2 接口使用一个时钟信号（C2CK）和一个双向的 C2 数据信号（C2D）在器件和宿主机之间传送信息。有关 C2 协议的详细信息见 C2 接口规范。

15.1 C2 接口寄存器

下面对与 FLASH 编程功能有关的 C2 寄存器进行说明。对所有 C2 寄存器的访问都要通过 C2 接口实现（详见 C2 接口规范）。

C2 寄存器定义 15.1 C2ADD: C2 地址寄存器

								复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	
位 7-0: C2ADD 寄存器选择 C2 数据读和数据写命令的目标数据寄存器。								
地址		说明						
0x00		选择器件 ID 寄存器（数据读指令）。						
0x01		选择版本 ID 寄存器（数据读指令）。						
0x02		选择 C2 FLASH 编程控制寄存器（数据读/写指令）						
0xB4		选择 C2 FLASH 编程数据寄存器（数据读/写指令）						

C2 寄存器定义 15.2 DEVICEID: C2 器件 ID 寄存器

								复位值
								00001001
位7	位6	位5	位4	位3	位2	位1	位0	
该只读寄存器返回 8 位的器件 ID 号：0x0D（C8051F326/7）。								

C2 寄存器定义 15.3 REVID: C2 版本 ID 寄存器

								复位值
								00000000
位7	位6	位5	位4	位3	位2	位1	位0	
该只读寄存器返回 8 位的版本 ID 号：0x01（版本 B）。								

C8051F326/7 全速 USB、16KB FLASH 微控制器

C2 寄存器定义 15.4 FPCTL: C2 FLASH 编程控制寄存器

复位值
00000000

位7

位6

位5

位4

位3

位2

位1

位0

位 7-0: FPCTL: FLASH 编程控制寄存器。

该寄存器用于使能通过 C2 接口对 FLASH 编程。为了使能 C2 FLASH 编程，必须按顺序写代码：0x02、0x01。一旦 C2 FLASH 编程被使能，必须进行一次复位才能使系统恢复正常工作。

C2 寄存器定义 15.5 FPDAT: C2 FLASH 编程数据寄存器

复位值
00000000

位7

位6

位5

位4

位3

位2

位1

位0

位 7-0: FPDAT: C2 FLASH 编程数据寄存器。

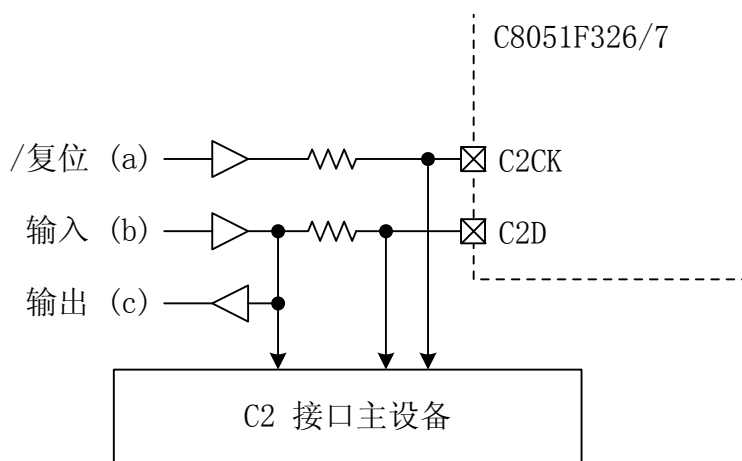
该寄存器用于在 C2 FLASH 访问期间传递 FLASH 编程命令、地址和数据。下面列出了有效的编程命令。

代码	命令
0x06	读 FLASH 块
0x07	写 FLASH 块
0x08	擦除 FLASH 页
0x03	擦除器件

15.2 C2 引脚共享

C2 协议允许 C2 引脚与用户功能共享，可以进行在系统调试和 FLASH 编程。这种共享之所以可能，是因为 C2 通信通常发生在器件的停止运行状态。在这种状态下片内外设和用户软件停止工作，C2 接口可以安全地“借用”C2CK (/RST) 和 C2D (P3.0) 引脚。在大多数情况下，需要使用外部电阻对 C2 接口和用户应用进行隔离。典型的隔离电路如图 15.1 所示。

图 15.1 典型 C2 引脚共享电路



在图 15.1 的配置中，我们假设：

1. 在目标器件的停止运行状态，用户输入 (b) 不能改变状态。
2. 目标器件的 /RST 引脚只能被作为输入使用。

对于具体的应用，可能还需要加入一些电阻。

联系信息

Silicon Laboratories Inc.

4635 Boston Lane
Austin, TX 78735
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032
Email: mcuinfo@silabs.com
Internet: www.silabs.com

新华龙电子有限公司

电话: 0755-83645240 83645242 83645244 83645251
技术支持: 0755-83645259
传真: 0755-83645243
地址: 深圳市福田区华强北路现代之窗大厦 A 座 13F C 室(518013)
Email: sales@xhl.com.cn
Email: shenzhen@xhl.com.cn
网站: www.xhl.com.cn
技术支持: support-sz@xhl.com.cn