



## MSP430F5xx 系列单片机 USB 模块用户手册(试行版)

版本: 1.0

发布日期: 2010.2. 最后更新日期:2010.2.10.

原文: TI Slau284.pdf (MSP430F5xx Family USB Module)

翻译: 周强 电子科技大学

编辑: DC 微控网总版主

注: 以下文章是翻译 TI slau284.pdf 文件中的部分内容。由于我们翻译水平有限, 有整理过程中难免有所不足或错误; 所以以下内容只供参考. 一切以原文为准。

文章更新详情请密切留意微控技术论坛。

## 1.1 USB 模块介绍

mSP430 单片机的 USB 模块具有以下特性：

\*完全符合 USB2.0 规范

- 集成 12Mbps 全速 USB 收发器
- 最多 8 个输出和 8 个输入节点
- 支持控制、中断和块传输模式
- 支持 USB 挂起、恢复和远程唤醒

\*拥有独立于 PMM 模块的电源系统

- 集成了 3.3V 输出的低功耗线性稳压器，该稳压器从 5V 的 VBUS 取电，输出足够驱动整个 mSP430 工作
- 集成了 1.8V 输出的低功耗线性稳压器为 PHY 和 PLL 模块供电
- 可工作与总线供电或自供电模式
- 3.3V 输出的线性稳压器电流限制功能
- USB 上电时自唤醒功能（系统没上电时）

\*内部 48MHZ 的 USB 时钟

- 集成可编程锁相环（PLL）
- 高度自由化的输入时钟频率，可使用低成本晶振

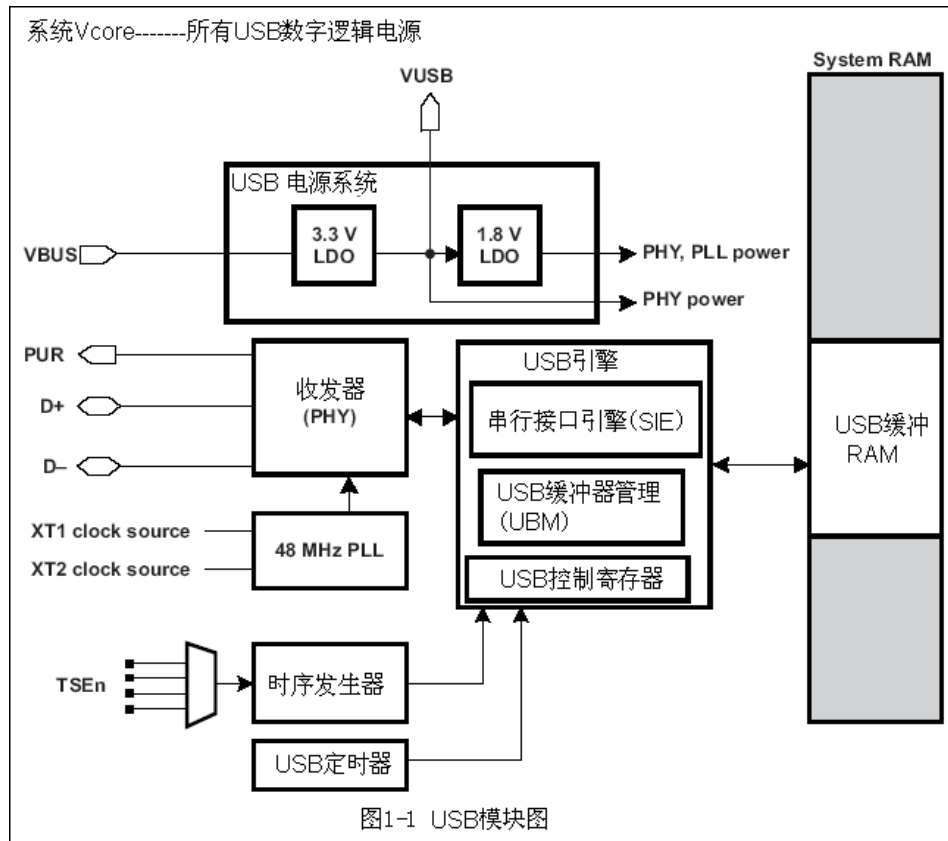
\*1904 字节独立 USB 端点缓存，可以每 8 个字节为单位进行配置

\*内置 62.5ns 精度的时间戳生成器

\*当 USB 模块禁止时：

- 缓存空间被映射到通用 RAM 空间，为系统提供额外 2KB 的 RAM
- USB 功能脚变为具有高电流驱动能力的通用 IO 口

\*USB 模块的结构图见 **Figure1-1**



## 1.2 USB 操作

USB 模块是一个全功能全速并且完全符合 USB2.0 协议的一个设备。USB 引擎完成所有 USB 相关的数据传输，它由 USB 串行接口引擎和 USB 缓冲管理器，USB 接收到的所有数据包被重新整理合并后放入接收缓存的 RAM 中，而在缓存中被标识准备就绪的数据被打包放入一系列的数据包后发送给其它 USB 主机。

USB 引擎需要一个精确的 48MHz 的时钟信号供采样输入的数据流使用，这个时钟信号由外部晶振源 (XT1 或 XT2) 产生的时钟信号通过锁相环后得到，但是要产生所需频率，要求锁相环的输入信号频率要大于 1.5MHz。锁相环的输出频率可以在很宽的范围内，非常灵活，允许用户在设计中使用低成本的晶振电路。

**注意：**有些设备的 XT1 仅支持低频率工作模式，PLL（锁相环）模块的输入只能是高频工作模式的时钟源，因此对于这些设备只能使用 XT2 作为 USB PLL 的输入，XT1（高频模式）和 XT2 的旁路模式（bypass mode）同样被 PLL 模块支持，具体请参考设备对应的芯片手册。

USB 缓存是 USB 接口和应用软件交换数据的地方，也是 7 个节点被调用的地方。缓存被设计成可被 CPU 或 DMA 以访问 RAM 的方式访问。

### 1.2.1 USB 收发器(PHY)

物理层的 USB 收发接口是一对直接从 3.3V 电压 VBUS 取电的差分线，数据线连接到外部 DP 和 DM 引脚，从而构成 USB 信号传输机制的接口。

当寄存器红 PUSEL 位置 1 时，DP 和 DM 被配置成 USB 的驱动线，受 USB 模块的控制，当 PUSEL 位被清零时，这两个引脚就变为具有强电流驱动能力的端口 U，其行为被 UPCR 寄存器控制。端口 U 从 VUSB 获取电源，独立于 DVCC。这两个引脚无论是用于 USB 功能还是用作通用 IO，都要使用内部稳压器或外部电源源给 VBUS 提供合适的供电。

### 使用 PUR 引脚将 D+ 上拉

当一个全速的 USB 设备连接到主机时，为了使主机能够识别，它必须将主机的 D+ 信号上拉。MSP430 单片机的 USB 模块有一个可软件控制的上拉引脚，通过外接一个电阻即可实现该功能。该功能通过控制寄存器的 PUR\_EN 位实现。如果该功能不需要软件控制，可以将 D+ 接至 VBUS 完成上拉。

### 电流过载时的保护

USB 设备必须能够忍受接入具有破坏性的线路时而不被损坏，因此，人们在供电线 GND 和 VBUS 上采取了保护措施。USB 设备的电气和物理特性应该能够不被此类事件所破坏。为此，MSP430 单片机的 USB 供电系统实现了一套电流限制机制来保证当此类短路事件发生时通过收发器的电流不会过大，有了这套机制，接口本身就不需要实现电流限制的功能了。注意，如果 VBUS 是使用外部供电电源而非内部稳压器输出，那么该外部源就要有一套自己的电流限制功能，为 USB 接口实现同样的保护功能。

### 端口 U 的控制

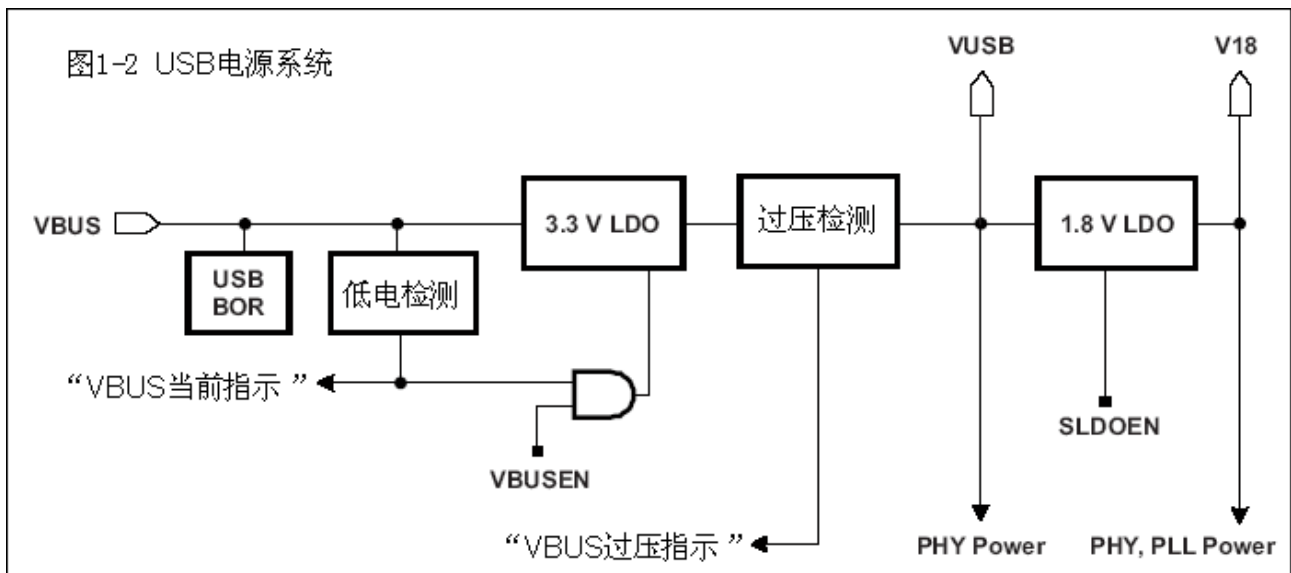
当 PUSEL 被清零时，端口 U（PU0、PU1 或 DP、DM）用作具有高电流驱动能力的通用 IO，PUDIR 控制着端口 U 的输出使能。该端口既可用于输出，也可用于输入，当配置为输入时，读取 PUIN0/1 的值可以获得输入值，如果用作输出，输出的值也会反映在 PUIN0/1 中。

当 PUDIR 置位时，端口 U 都被配置为输出口，使用 PUOUT0 和 PUOUT1 控制。当输出高电平时，输出和 VBUS 同轨（电平一致），电流驱动能力比普通的 IO 口要高很多，具体参数请参考相应芯片手册。

PUDIR 的缺省值是 0，因此 PU0 和 PU1 在 USB 模块禁用时呈高阻态。

### 1.2.2 USB 供电系统

USB 模块的供电系统内含双稳压器（3.3V 和 1.8V），当 5V 的 VBUS 可用时，允许整个 MSP430 从 VBUS 供电。作为可选的，供电系统可以只为 USB 模块供电，也可以在一个自供电设备中完全不被使用。供电系统的结构图如图 1.2。



内部 3.3V 稳压器从 5V 的 VBUS 取电并供给收发器和外部 VUSB 引脚。使用该稳压器能够避免使用外部

供电时高负载通过收发器和锁相环，因此在电池供电设备中非常有用。

内部 1.8V 稳压器取电于 VUSB 引脚（VUSB 引脚取电于内部 3.3V 稳压器或外部源），并给 USB 的锁相环和收发器提供电源。1.8V 的稳压器独立于 MSP430 电源管理模块内部的稳压器。

稳压器模块的输入输出如 Figure 1-2 所示，VBUS，VUSB 和 V18 需要连接外部的电容。V18 引脚仅设计成用来挂接一个外部负载电容，而不具备给其它模块供电的能力。

### 使能/禁止

3.3V 的稳压器通过设置 VUSBEN 来使能或禁止。但是如果检测到 VBUS 的电压过低或不存在，即使稳压器使能也会挂起。当 VBUS 电压升高至 USB 电源最低水平时，稳压器的电压参考和低电压检测将会工作。当 VBUS 电压升至更高达到起始电压 V<sub>launch</sub>，稳压器模块正常工作（参见 Figure 1-3）。

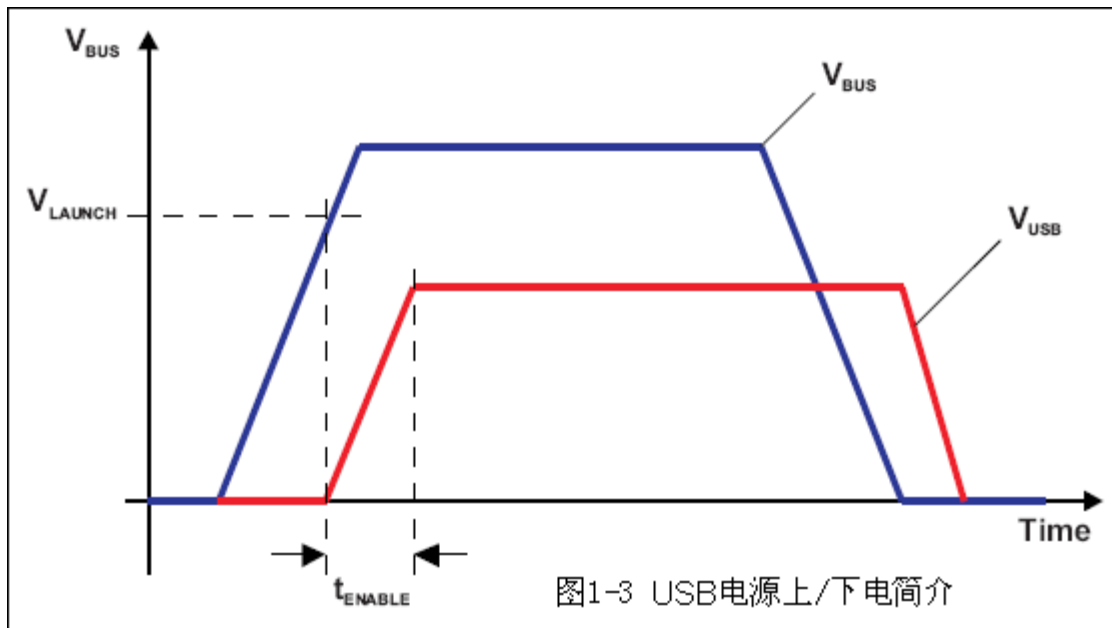


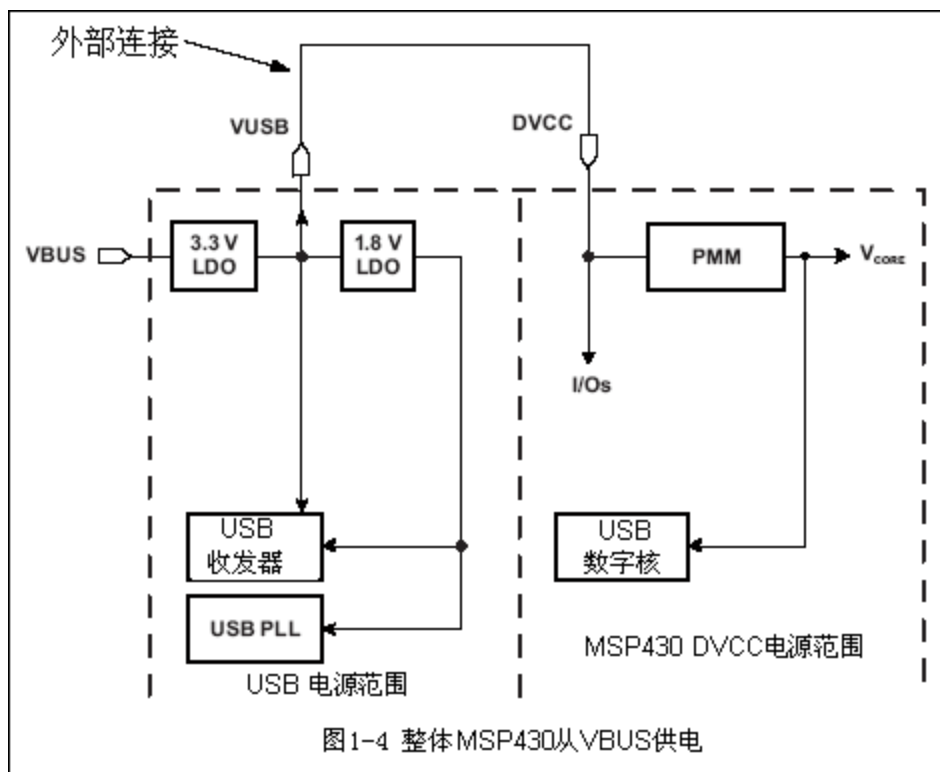
图1-3 USB电源上/下电简介

1.8V 稳压器可以通过设置 SLDOEN 控制使能和禁止。默认地，SLDOEN 自动随着 VBUS 的电压是否可用变化，这项特征通过 SLDOAON 控制。如果 VUSB 不是从内部 3.3V 稳压器而是从外部源供电，请谨记如果 VBUS 没有和外部 5V 电源连接，1.8V 的稳压器不会自动工作，这种情况下，VBUS 必须连接至 USB 总线电源或 SLDOAON 位清零，SLDOEN 置位。

当外部设备从 USB 的 VBUS 线获得电源时，需要在进入终端设备前经过一个肖恩特二极管，避免了电流从稳压器的输入端窜入 USB 总线的 VBUS，这就允许 MSP430 能够和挂起或未上电的 USB 总线设备保持电气特性的连接。

### 通过 USB 总线的 VBUS 为 MSP430 的其余部分供电

3.3V 稳压器的输出达到了与 DVCC 同轨，可以用于为整个 MSP430 设备供电。要实现该功能，要求 VUSB 和 DVCC 在外部被连接在一起，此时 3.3V 的稳压器输出供电给 DVCC（参见 Figure 1-4）。



当MSP430这样连接时，当VBUS上的电压上升到 $V_{launch}$ 及其以上时，如果 $V_{core}$ 信号没有电压供给，就意味着单片机系统尚未上电工作，3.3V稳压器和1.8V稳压器就会自动开始工作，给单片机系统提供电源。

注意如果单片机使用这种从VUSB取电的方式，当试图把单片机置于低功耗模式LPM5时，会导致系统立刻重起，这是因为当进入LPM5时，创造了上面所描述的自治条件的特性（ $V_{core}$ 无电压而VBUS可用），于是会引起系统立刻重起。

如果DVCC从VBUS获得电源，用户就要负责从VBUS上取得的电流的总大小小于 $I_{DET}$ 。

### 通过USB总线的VBUS为MSP430外部组件供电

MSP430内部3.3V的稳压器不仅能提供整个单片机正常工作的电流，而且有充足的余量通过VUSB引脚为MSP430外部的组件供电。

如果整个系统的工作总是伴随USB的工作，系统就不再需要其他的电源。但如果系统的USB只是偶尔被连接使用并且系统是有电池来供电，3.3V的稳压器能够接替电池的负载。再者，如果电池是可充电的，VBUS还可以对其进行充电。

### 限流和过载保护

内置3.3V稳压器的电流限制功能在线路短路时可以起到保护收发器的功能。短路或过载事件（当稳压器的输出电流达到或超过 $I_{DET}$ 时）通过VUOVIFG标志位传递给软件。当该事件发生时，由于电流的供应不足，USB设备的操作是不可靠的，此时软件很可能要终止USB操作，USB模块提供了该功能，通过设置OVLAOFF位，USB操作就通过VUSBEN清零自动被终止了。

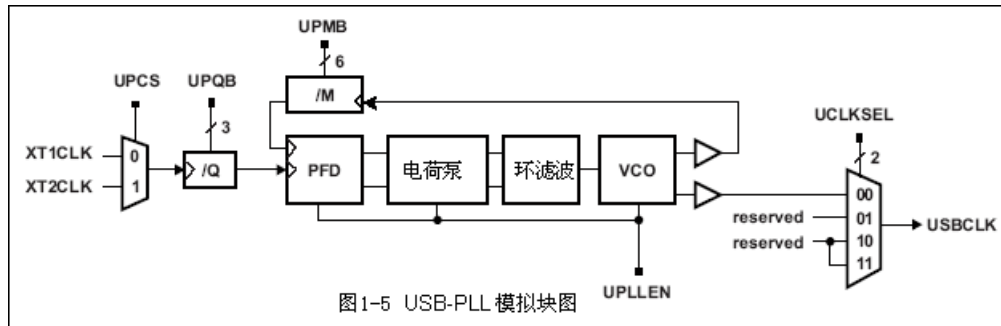
在过载条件下，VBUS和V18电压都会低于正常输出水平。如果DVCC仅仅只从VBUS取电，系统会不断重复地被触发重起，只要短路或过载条件存在。因此，固件应该在检测到过载后避免重新使能USB设备，直到该错误消失。

USB系统的VBUS和DVCC都配有降压电路，它们承载着更高的电压。

最后，使用者应注意保证从VBUS获取的电流不会超过 $I_{DET}$ 。

### 1.2.3 USB 锁相环(PLL)

PLL 锁相环模块为 USB 操作提供高精度低抖动的时钟。(参见 Figure1-5)



外部的参考时钟通过 UPCS 位进行选择，允许使用两个外部晶振之一作为参考时钟源。一个受 UPQB 位控制四位的预分频计数器允许对参考时钟进行分频产生 PLL 的更新时钟。UPMB 位控制着反馈回路上的分频因子和 PLL 的倍频因子（参见公式 1-1）。

$$f_{OUT} = CLK_{SEL} \times \frac{DIVM}{DIVQ} \quad \text{with} \quad \frac{CLK_{SEL}}{DIVQ} = f_{UPD} \geq 1.5 \text{ MHz} \quad 1-1$$

式中：

**CLKSEL** 是所选的参考时钟源频率（XT1CLK 或 XT2CLK）；

**DIVQ** 的值来源于 **Table1-1**；

**DIVM** 代表寄存器中 UPMB 域的值；

如果 USB 设备的操作是在总线供电的模式下，为了使 USB 的电流消耗小于 500uA，有必要禁止 PLL 工作，通过 UPLLEN 位可使能或禁止 PLL。为使能鉴相器，PFDEN 位必须置位。信号失锁，输入信号无效和超出正常工作频率会反映在对应的中断标志位 OOLIFG，LOSIFG 和 OORIFG 上。

**注意：** UCLKSEL 默认值为零，并应该总是这样，并且所有其它可能的组合值都是保留的，以备以后扩展所使用。

**Table 1-1. USB-PLL Pre-Scale Divider**

UPQB	DIVQ
000	1
001	2
010	3
011	4
100	6
101	8
110	12
111	16



Table 1-2. Register Settings to Generate 48 MHz Using Common Crystals

CLKSEL (MHz)	UPQB	UPMB	DIVQ	DIVM	CLKLOOP (MHz)	UPLLCLK (MHz)	ACCURACY (ppm)
1.5	000	011111	1	32	1.5	48	0
1.6	000	011101	1	30	1.6	48	0
1.7778	000	011010	1	27	1.7778	48	0
1.8432	000	011001	1	26	1.8432	47.92	-1570
1.8461	000	011001	1	26	1.8461	48	0
1.92	000	011000	1	25	1.92	48	0
2	000	010111	1	24	2	48	0
2.4	000	010011	1	20	2.4	48	0
2.6667	000	010001	1	18	2.6667	48	0
3	000	001111	1	16	3	48	0
3.2	001	011110	2	30	1.6	48	0
3.5556	001	011010	2	27	1.7778	48	0
3.579545	001	011010	2	27	1.79	48.32	6666
3.84	001	011001	2	25	1.92	48	0
4 <sup>(1)</sup>	001	010111	2	24	2	48	0
4.1739	001	010110	2	23	2.086	48	0
4.1943	001	010110	2	23	2.097	48.23	4884
4.332	001	010101	2	22	2.166	47.652	-7250
4.3636	001	010101	2	22	2.1818	48	0
4.5	010	011111	3	32	1.5	48	0
4.8	001	010011	2	20	2.4	48	0
5.33 $\approx$ (16/3)	001	010001	2	18	2.6667	48	0
5.76	010	011000	3	25	1.92	48	0
6	010	010111	3	24	2	48	0
6.4	011	011101	4	30	1.6	48	0
7.2	010	010011	3	20	2.4	48	0
7.68	011	011000	4	25	1.92	48	0
8	010	010001	3	18	2.6667	48	0
9	010	001111	3	16	3	48	0
9.6	011	010011	4	20	2.4	48	0
10.66 $\approx$ (32/3)	011	010001	4	18	2.6667	48	0
12	011	001111	4	16	3	48	0
12.8	101	011101	8	30	1.6	48	0
14.4	100	010011	6	20	2.4	48	0
16	100	010001	6	18	2.6667	48	0
16.9344	100	010000	6	17	2.8224	47.98	-400
16.94118	100	010000	6	17	2.8235	48	0
18	100	001111	6	16	3	48	0
19.2	101	010011	8	20	2.4	48	0
24	101	001111	8	16	3	48	0
25.6	111	011101	16	30	1.6	48	0
32	111	010111	16	24	2.6667	48	0

### 修改分频器分频系数

在设置所需 PLL 的频率时，更新 UPQB (DIVQ) 和 UPMB (DIVM) 值的动作必须同步进行，以避免寄生频率的残留。UPQB 和 UPMB 的值经计算后先写入缓冲寄存器；最后通过写 UPLLDIVB 同时更新 UPQB



和 UPMB 的值。

### PLL 错误的标志

PLL 可以检测三种错误：当频率在连续 4 个更新周期在同一方向上修正时，将检测到失锁错误；当频率在连续 16 个更新周期在同一方向上修正时，将检测到信号失效；当频率在连续 32 个更新周期内未被锁住，将检测到信号超出正常工作频率范围。这三种错误将触发它们对应的中断标志位（USBOOLIFG、USBLOSIFG、USBOORIFG）置位，如果对应的中断使能位（USBOOLIE、USBLOSIE、USBOORIE）置位，将触发相应的中断。

### PLL 启动顺序

推荐使用下面的操作顺序以获得最快的 PLL 启动：

- 1 使能 VBUS 和 V18。
- 2 等待外部电容充电 2 毫秒，以使 VUSB 就位。（在这期间可以初始化 USB 寄存器和缓存。）
- 3 激活 PLL，使用所需的分频值。
- 4 等待 2 毫秒并检查 PLL，如果仍然保持锁定状态，就可以被使用了。

### 1.2.4 USB 控制器引擎

USB 控制器引擎将到达 USB 设备的数据包转移到 USB 缓冲空间中，同时将有效数据从缓冲空间发送给 USB 接口。控制引擎拥有专用固定的缓冲空间为输入端点 0 和输出接点 0 所使用，端点 0 是默认的 USB 传输控制接点。

其余的 14 个端点（7 个输入，7 个输出）可能被指派一个或更多的 USB 缓冲空间，所有的缓冲空间都位于 USB 的缓冲存储器中，USB 缓冲存储器被设计成可多端口访问的，既可以被 USB 控制器访问，同时也可以被 CPU 和 DMA 访问。

每个端点都有一个专用的描述寄存器用来描述该端点的使用（参见 Figure 1-6）。各端点的配置通过设置对应的描述寄存器完成，这种包含下一次收发操作将用到的缓存地址的数据结构存放于 USB 缓冲存储器中。给一个端点指派 1 个或 2 个 64 字节的数据缓存块，在配置后不需要额外的软件干预，但如果需要指派 3 个或更多的数据缓存块，软件就必须在数据传输过程中改变描述寄存器的地址指针。

缓存中数据的空状态和满状态通过确认标志位同步，所有的事件都通过设置标志位并在中断使能时引发中断来告知。且发送事件的告知可以单独地被使能。

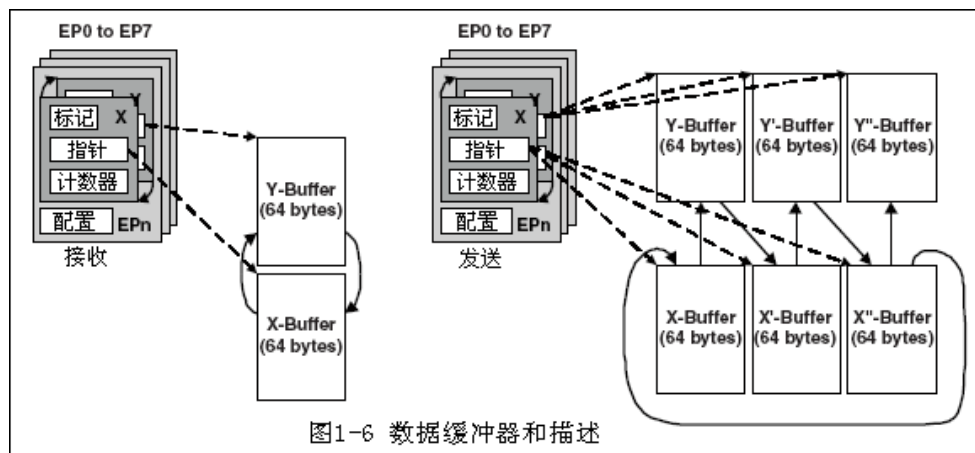


图1-6 数据缓冲器和描述

### USB 串行接口引擎 (SIE)

SIE 逻辑单元管理着 USB 总线上数据收发的协议。对于收到的数据包，SIE 将其数据包标志 PID 区域解码

并依据 PID 判断 PID 是否有效，同时判断数据包的类型。针对收到的数据包及其特征，SIE 计算其 CRC 校验值并与数据包中的 CRC 值进行比较来判断数据在传输过程中是否出错。

针对将要发送数据包及其特征，SIE 计算其 CRC 校验值并和数据包一起传输出去。对于发送出去的每个数据包，SIE 会在数据包前附上 8bit 的同步字节。另外，SIE 还会给所有将发送出去的数据包产生对应的 PID。SIE 另外一个重要的功能是负责所有收发数据的串并转换。

### USB 缓冲管理 (UBM)

USB 模块的缓冲管理 UBM 提供 SIE 到 USB 端点缓冲的控制逻辑。UBM 的一个主要功能是将 USB 功能地址译码，并以此来决定 USB 主机是否在访问特定的设备。另外，通过端点的地址和方向信号译码可以判断哪个 USB 端点正在被寻址访问。基于传输的方向信息和端点号，UBM 可以在对应 USB 端点的数据缓冲上写入或读取数据包。

### USB 缓存

USB 缓存包含了所有端点的数据缓冲和 SETUP 数据包。因为端点 1-7 的缓冲是灵活可变的，因此就有相应的缓冲配置寄存器来定义它们，这些寄存器也位于缓存中。（端点 0 特殊，被定义在 USB 控制寄存器空间）把这些寄存器信息存于普通的存储空间允许高效，高度灵活的应用，具有可针对特定应用进行高度灵活定制的特点。

该缓冲存储器被设计成“多端口存储器”，因此既可以被 USB 缓冲管理器访问，也可以被 CPU 和 DMA 访问。但 CPU 和 DMA 的访问优先级在此低于 SIE，如果 CPU/DMA 访问和 SIE 访问相冲突，CPU/DMA 访问将会进入等待状态进行延时。

当 USB 模块被禁止时（USBEN=0），该缓存的行为和普通 RAM 一样。当改变 USBEN 位的状态时（使能或禁止 USB 模块），要保证该 USB 缓存在之前的 4 个时钟周期和之后的 8 个时钟周期内不被访问，改变后，USB 缓存的访问方式将发生变化。

每个端点都被 6 个配置“寄存器”所定义（基于 RAM，严格地讲并不是真正的寄存器）。这些寄存器指定了端点的类型，缓冲的地址，缓冲的大小和数据包的字节数。它们定义和划分了一个大小是 1904 字节端点缓冲空间。另有 24 字节分配给了剩下的模块——EP0\_IN 缓冲，EP0\_OUT 缓冲和 SETUP 包缓冲（参见 Table1-3）。

Table 1-3. USB Buffer Memory Map

Memory	Short Form	Access Type	Address Offset
Start of buffer space	STABUFF	Read/Write	0000h
1904 bytes of configurable buffer space	USBIEPCNT_0	Read/Write	:
End of buffer space	TOPBUFF	Read/Write	076Fh
Output endpoint_0 buffer	USBOEP0BUF	Read/Write	0770h
		Read/Write	:
		Read/Write	0777h
Input endpoint_0 buffer	USBIEP0BUF	Read/Write	0778h
		Read/Write	:
		Read/Write	077Fh
Setup Packet Block	USBSUBLK	Read/Write	0780h
		Read/Write	:
		Read/Write	0787h

软件可依据端点的需要来配置各个缓冲，每个端点都可以配置成单缓冲或双缓冲空间。

不像端点 1-7 的描述寄存器都定义在 USB 的 RAM 中，端点 0 被 USB 控制寄存器中的四个寄存器所描述（两个输出，两个输入），因为这些寄存器的地址都是硬件固定的，因此端点 0 没有基址寄存器，端点 n(n 从 1 到 7)的选择被封装成选择寄存器对应位的设置。

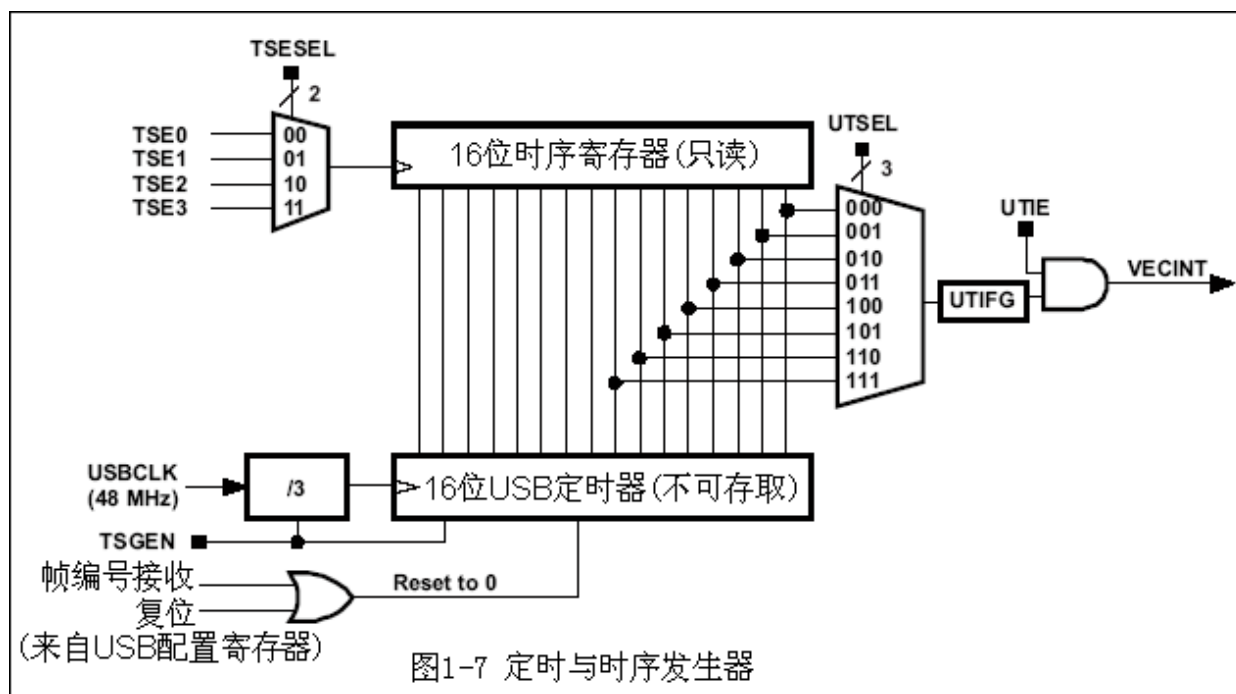
## USB 时间戳

USB 模块可以保存特定 USB 事件的时间戳（参见 Figure 1-7），这在软件为特定响应进行延时补偿时将非常有用，时间戳的值基于 USB 模块内部由 USBCLK 驱动的定时器。

可通过 TSESEL 位来选择四种事件来产生时间戳，当事件发生时，USB 定时器的值会传送给时间戳寄存器 USBTSREG，这样事件发生的确切时刻就被记录下来，触发可选的事件包括三个 DMA 通道和一个软件触发事件。USB 定时器不能被直接进行读取访问。

此外，USB 定时器的值可以被用于产生周期性的中断。因为 USBCLK 可以拥有和其它系统时钟不同的频率，这就给产生周期性系统中断多了一个选择。UTSEL 位选择从 USB 时钟进入定时器的分频系数，为使中断向量能够触发中断，中断使能位 UTIE 应该置位。

时间戳寄存器在收到一个整帧和一帧假始时应该归零，通过 TSGEN 位可使得能和禁止时间戳生成器。



挂起和恢复单元

USB 的挂起和恢复单元侦测 USB 总线上的挂起和恢复条件，这些事件被标识在相应的标识寄存器 SUSRIFG 和 RESRIFG 中，当中断使能位 SUSRIE 和 RESRIE 置位时，触发相应中断。

用来唤醒和恢复 USB 主机设备的远程唤醒机制，通过设置 USBCTL 的 RWUP 位来触发。

USB 标准要求总线供电的 USB 设备在挂起状态时从总线获得的电流不超过 500uA，为了达到总线供电设备的这个要求，通常需要把 PLL 模块禁止，禁止 PLL 能够消除不必要的电流消耗。在挂起状态时，USBCLK 自动选择 VLO (VLOCLK) 作为时钟源，允许 USB 模块侦测恢复条件的发生。（更多信息参见 2.6 节）

## 复位逻辑

一次上电清除操作（PUC）将使 USB 模块复位（整个 MSP430 都将复位）。USB 主机向总线发出的 USB 复位信号也会触发 USB 的复位事件，同时触发标志位 RSTRIFG 置位，在这种复位事件中，USB 缓存中的内容将会保持，不会被复位。

### 1.2.5 USB 中断向量

USB 模块使用单一的中断向量产生寄存器来处理多种 USB 中断。所有和 USB 相关的中断源触发中断向量，然后在 USBVECINT 中保存一个 6 位的向量用来标识中断源。每个中断源都产生一个不同的偏移值，没有中断挂起时中断向量返回 0；

中断向量寄存器和对应的标识寄存器在读取后将被更新和清零。读取中断向量寄存器时，优先级最高的中断将返回 0002H，优先级最低的返回 003EH，并且写该寄存器将清除所有的中断标志。

每个 USB 输入或输出的端点设备都有一个中断告知使能位，软件必须设置该位来定义它们对中断事件是否进行标志。为了产生一个中断，对应的中断使能位必须被置位。

**Table 1-4. USB Interrupt Vector Generation**

USBECINT Value	Interrupt Source	Interrupt Flag Bit	Interrupt Enable Bit	Indication Enable Bit
0000h	no interrupt	—	—	—
0002h	USB-PWR drop ind.	USBPWRCTL.VUOVLIFG	USBPWRCTL.VUOVLIE	—
0004h	USB-PLL lock error	USBPLLIR.USBPLLOOLIFG	USBPLLIR.USBPLLOOLIE	—
0006h	USB-PLL signal error	USBPLLIR.USBPLLOSIFG	USBPLLIR.USBPLLOSIE	—
0008h	USB-PLL range error	USBPLLIR.USBPLLOORIFG	USBPLLIR.USBPLLOORIE	—
000Ah	USB-PWR VBUS-on	USBPWRCTL.VBONIFG	USBPWRCTL.VBONIE	—
000Ch	USB-PWR VBUS-off	USBPWRCTL.VBOFFIFG	USBPWRCTL.VBOFFIE	—
000Eh	reserved	—	—	—
0010h	USB timestamp event	USBMAINTL.UTIFG	USBMAINTL.UTIE	—
0012h	Input Endpoint-0	USBIEPIFG.EP0	USBIEPIE.EP0	USBIEPCNF_0.USBIE
0014h	Output Endpoint-0	USBOEPIFG.EP0	USBOEPIE.EP0	USBOEPCNF_0.USBIE
0016h	RSTR interrupt	USBIFG.RSTRIFG	USBIE.RSTRIE	—
0018h	SUSR interrupt	USBIFG.SUSRIFG	USBIE.SUSRIE	—
001Ah	RESR interrupt	USBIFG.RESRIFG	USBIE.RESRIE	—
001Ch	reserved	—	—	—
001Eh	reserved	—	—	—
0024h	Input Endpoint-1	USBIEPIFG.EP1	USBIEPIE.EP1	USBIEPCNF_1.USBIE
0026h	Input Endpoint-2	USBIEPIFG.EP2	USBIEPIE.EP2	USBIEPCNF_2.USBIE
0028h	Input Endpoint-3	USBIEPIFG.EP3	USBIEPIE.EP3	USBIEPCNF_3.USBIE
002Ah	Input Endpoint-4	USBIEPIFG.EP4	USBIEPIE.EP4	USBIEPCNF_4.USBIE
002Ch	Input Endpoint-5	USBIEPIFG.EP5	USBIEPIE.EP5	USBIEPCNF_5.USBIE
002Eh	Input Endpoint-6	USBIEPIFG.EP6	USBIEPIE.EP6	USBIEPCNF_6.USBIE
0030h	Input Endpoint-7	USBIEPIFG.EP7	USBIEPIE.EP7	USBIEPCNF_7.USBIE
0032h	Output Endpoint-1	USBOEPIFG.EP1	USBOEPIE.EP1	USBOEPCNF_1.USBIE
0034h	Output Endpoint-2	USBOEPIFG.EP2	USBOEPIE.EP2	USBOEPCNF_2.USBIE
0036h	Output Endpoint-3	USBOEPIFG.EP3	USBOEPIE.EP3	USBOEPCNF_3.USBIE
0038h	Output Endpoint-4	USBOEPIFG.EP4	USBOEPIE.EP4	USBOEPCNF_4.USBIE
003Ah	Output Endpoint-5	USBOEPIFG.EP5	USBOEPIE.EP5	USBOEPCNF_5.USBIE
003Ch	Output Endpoint-6	USBOEPIFG.EP6	USBOEPIE.EP6	USBOEPCNF_6.USBIE
003Eh	Output Endpoint-7	USBOEPIFG.EP7	USBOEPIE.EP7	USBOEPCNF_7.USBIE

### 1.2.6 USB 功耗

USB 功能消耗的电能比它获取电能的 MSP430 系统的功耗还大，因为 MSP430 的应用场合大多是功耗敏感的，因此 MSP430 的 USB 模块被设计成从 VBUS 取电，并且只有连接到 USB 总线时才会出现高功耗的负载，这样有效地保护了电池。

USB 模块的两个最耗电的组件是收发器和 PLL。收发器在传输时会消耗大量的电能，但当处在不活动状态时，不进行数据的收发，此时将消耗极小的电流，这个量被定义成 **IDLE**。这个量非常小，以至于在总线供电的应用中，收发器可以在挂起模式时仍保持活动状态而不带来任何问题。幸运的是，收发器在获得收发所需的电流时会访问 VBUS 电源。

PLL 组件消耗很大一部分电流，不过它只需要在连接到主机时被激活，并且由主机的 USB 总线供电，当 PLL 禁止时（例：在 USB 挂起时），USBCLK 自动选择 VLO 作为时钟源。

### 1.2.7 进入挂起模式

当主机将 USB 设备挂起时，一个挂起中断将被产生 (SUSRIFG)，从这点来看，软件有 10ms 的时间来保证 USB 设备从 VBUS 获得的电流小于 500uA，为了满足这一条件，经常需要完成以下操作：

- \* 通过清除 UPLLEN 禁止 PLL

- \* 禁止 PLL 的高速晶振源

接着保证 RESRIE 位被置位是个不错的做法，这样的话当设备从挂起状态被恢复时就会产生一个中断。

## 1.3 寄存器

- 1.2.1 配置寄存器

- 1.2.2 控制寄存器

- 1.2.3 缓冲寄存器和存储器

待续...

### 参与微控网翻译义工

微控网翻译义工队伍长期需要有能力、有心人加入，参与详情请进入微控论坛查看。