

## 关于 USB2.0 模块的技术问答

为了简化 USB2.0 接口产品的设计,西安达泰电子先后开发了 USB20C 和 USB20D 模块。这个模块在市场上经过 5 年的推广应用,已经有上万家用户使用了这个产品。例如北京天惠公司用于其测试设备中,成都 57 所用于军工产品中。

另外有相关应用书籍及论文公开出版发表:

论文题目: [《基于USB20C模块的单片机系统与PC机的通信》](#)

作者: 曾金芳, 杨恢先, 李正义

作者单位: 湘潭大学信息工程学院, 湖南湘潭 411105

发表刊物: 《电子器件》第 28 卷 第 4 期 2005 年 12 月

书名: [《USB应用开发技术大全》](#)

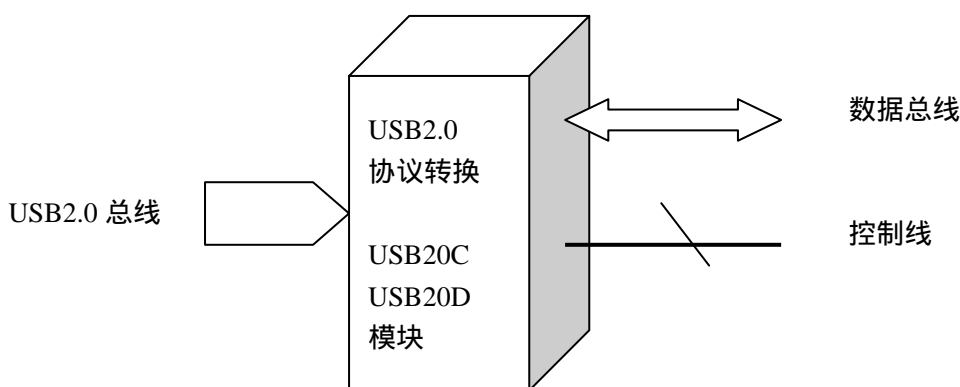
作者: 薛园园 编著

出版: 人民邮电出版社

设计 USB2.0 模块的出发点是将复杂的问题简单化。因为对于大多数应用工程师来说, 开发 USB2.0 接口的产品主要障碍在于熟悉复杂的 USB2.0 协议, 具体来说要解决两个问题。首先要自己编写 USB 设备的驱动程序, 这要求起码有相当的 VC 编程经验, 其次要编写 USB 接口的硬件(固件)程序, 对于熟悉单片机的编程只是基本要求, 更重要的是要真正掌握 USB2.0 的协议内容。所以使用 USB2.0 接口开发产品, 对于开发人员要求软硬件的高手才行。

由此可见, 如果不是对 USB 协议有特别爱好, 那么就没有必要把大量宝贵的时间浪费在研究 USB 的协议上。西安达泰电子有限责任公司长期从事 USB 数据采集板卡的研究, 针对 USB 接口产品的开发问题专门研制了 USB20C 和 USB20D 模块。

USB20C 模块可以被看作是一个 USB2.0 协议的转换器, 通过这个模块将电脑的 USB2.0 接口转换为一个透明的并行总线, 就象 ISA 总线一样。



对于用户来说, USB20C 模块就是一个黑匣子, 这个黑匣子对 USB 的协议进行了解释, 用户不需要了解详细的过程。如果使用过 ISA 总线的产品对此过程就非常容易了解, 而对于开发过单片机产品的用户来说, 就可以把这个模块当作一个外部芯片来看待, 例如 SRAM 或者 FIFO 等。

原来我们面对的是复杂的 USB 接口, 而现在从 USB20C 模块的用户接口看, 我们面对的仅仅是一个标准的并行总线而已。



## 常见问题解答：

**问题 1：USB20C 与 USB20D 有何区别？**

USB20C 模块是 2003 年推出的，USB20D 模块是 2006 年推出的。USB20D 模块是在 USB20C 的基础上，从硬件和软件功能方面进行了改进。USB20D 是 USB20C 的升级产品。二者的功能特点比较如下：

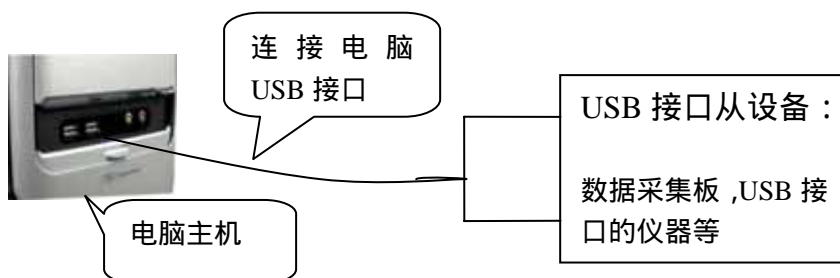
序号	USB20D 模块	USB20C 模块
	<b>硬件方面：</b>	
1	DMA 方式时数据总线可选择 16 位	DMA 方式时数据总线只能是 8 位
2	高 8 位数据线与地址线和控制线复用	数据总线没有复用
3	屏蔽了 I2C 总线输出	开放 I2C 总线输出
4	模块尺寸封装没变，与 USB20C 兼容	尺寸与 USB20D 兼容
	<b>软件方面：</b>	
1	可以在同一应用程序中控制多个 USB20D 模块	在同一应用程序中只能控制一个 USB20D 模块
2	可选 16 位数据总线进行 DMA 传输	只能 8 位数据总线 DMA 传输
3	提供了 FIFO 状态查询函数	没有 FIFO 状态查询函数
	<b>系统方面：</b>	
1	DMA 读写速度大于 25Mbyte/Sec，与电脑速度有关，目前双核计算机测试速度大于 30Mbyte/Sec。	DMA 读写速度大于 20Mbyte/Sec
2	驱动程序兼容所有 Windows 系统	对于有些 XP 版本驱动可能冲突

USB20D 模块的 DMA 模式数据传输速度大于 25Mbyte/Sec，明显高于 USB20C 模块。在 API 函数方面结合 USB20C 用户反馈的意见进行了改进，例如增加了 FIFO 状态查询函数。所以建议新用户采用 USB20D 模块开发其 USB2.0 产品。

**问题 2：关于 USB 接口的主从概念**

如果准备开发 USB2.0 接口的产品，首先要确认是 USB 从设备还是 USB 主设备。我们知道，USB 接口是有主从之分的。简单地说，电脑的 USB 接口是主，连接电脑 USB 接口的设备是从，只有主从设备之间可以直接通信。所以 USB20D 模块是从设备，如果您准备开发的产品是与电脑 USB 接口连接的，那么就是 USB 从设备。

USB 从设备比较多，例如 U 盘，USB 接口的打印机、键盘、鼠标、摄像头等等。自己开发的 USB 设备一般是非标的，例如 USB 接口的数据采集板，USB 接口的仪器等。



### 问题 3：如何使用 USB20D 快速开发产品？

选择 USB20D 模块开发 USB2.0 接口的从设备是最为快捷简单的途径,对于具有一定单片机或 FPGA 开发经验的设计人员来说,不到 1 周时间就可以调通 USB 接口的数据传输单元。参考随该产品提供的大量实例代码,并与西安达泰公司技术支持进行沟通开发周期会大大缩短。

下面以开发 USB2.0 接口的数据采集板为例说明过程：

#### 1 首先根据采样率确定主控芯片

如果 A/D 采样率小于 200KHz,那么采用单片机就可以了;如果 A/D 采样率大于 200KHz,建议采用 CPLD、FPGA、ARM、DSP 等作为主控芯片。

#### 2 设计硬件电路

把 USB20D 模块作为系统硬件的一个单元,主要负责数据的传输。如果希望进行高速连续的数据传输,那么建议在系统中设计比较大的缓存单元(FIFO、SRAM 或双口 RAM 等)。

#### 3 编写程序

程序分硬件程序和上位机程序两个部分。

硬件程序主要协调 A/D 转换器、存储器和 USB 接口的控制。对于 USB20D 模块来说编程就是非常简单的事,只要搞清楚 DMA 的时序就可以了。

上位机软件程序一般用高级语言编写,例如 VC、VB、Delphi,Labview 也可以,无论采用那种语言,都是调用 USB20D 附带的 Windows API 函数(以 USB20D.DLL 形式提供)。对于 PCI 或 ISA 总线编程熟悉的话,原理都是一样的。

### 问题 4：USB20D 模块的地址 I/O 模式解释

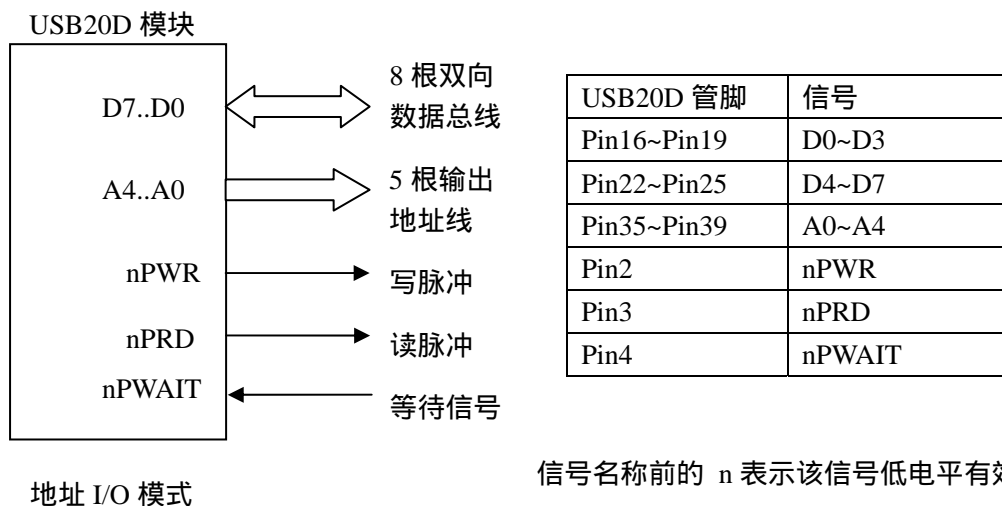
USB20D 模块设计了两种工作模式：地址 I/O 模式、DMA 模式。

地址 I/O 模式主要用于通过上位机设置设备的参数,例如采集板的采样率,记录长度等;或者检测设备的工作状态,例如判断采集板 A/D 转换是否结束。这种模式主要用于高速数据传输前的准备,速度要求不高的情况。当然如果不进行高速数据传输的话,也可以通过这个模式进行数据采集。

地址 I/O 模式涉及到 16 根信号线,其中 8 根双向数据线、5 根地址线、2 根读写输出线、1 根等待输入线。

8 根双向数据线：注意在 I/O 模式下是 8 位数据线

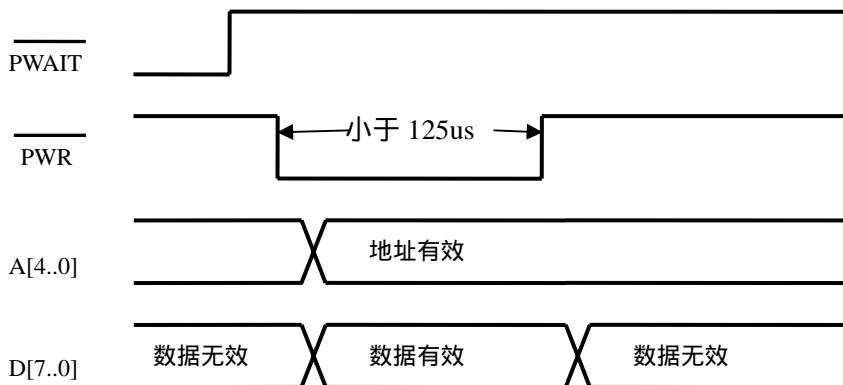
5 根地址线：这里地址线与一般意义上地址线不太一样，应该理解为口地址线。虽然 5 根线很少，但是有 32 个状态，所以就可以分配 32 个口地址，作为设备命令状态等操作足够用了，我们知道，PC 总线分配给并口的口地址也不过 4 个。单片机外扩器件时片选信号也是通过分配口地址实现的。



地址 I/O 模式的时序图解释：

在 I/O 模式下，数据线为 8 位双向总线，地址线有锁存。

#### 1. 无等待周期的 I/O 写时序图



无等待周期的应用一般是以 CPLD、FPGA、DSP 等高速处理器作为主控芯片时采用。这个时序对于上位机来说就是发送了一个 OutPut 命令，将数据输出到数据线上。例如：上位机软件调用这个函数

`USB20D_Output(HANDLE DevHandle, int Addr, int Data)`

假如 Addr=5，Data=18，那么执行该函数在 USB20D 的接口上

地址线状态：

Pin35(A0)=1，Pin37(A2)=1 高电平带锁存

Pin36(A1)=0，Pin38(A3)=0，Pin39(A4)=0，低电平带锁存。

地址线带锁存，所以可以用万用表测量。

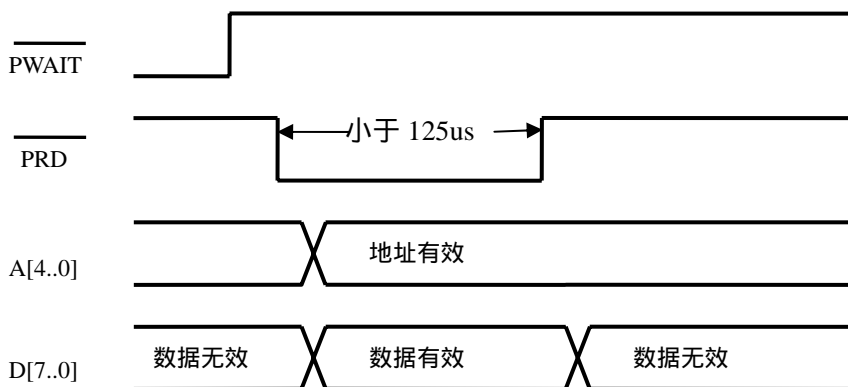
数据线状态：只有在 PWR 信号为低电平的时候有效，否则是无效数据

Pin16(D0)=1, Pin17(D1)=1, Pin22(D4)=1 高电平

Pin18(D2)=0, Pin19(D3)=0, Pin23(D5)=0, Pin24(D6)=0, Pin27(D7)=0 低电平

数据线没有锁存，所以只能用示波器跟踪。

## 2. 无等待周期的 I/O 读时序图



这个时序对于上位机来说就是发送了一个 InPut 命令，将数据输出到数据线上。例如：上位机软件调用这个函数

```
int Data = USB20D_Input(HANDLE DevHandle, int Addr);
```

假如 Addr=5，读到的 Data=18，那么执行该函数在 USB20D 的接口上地址线状态：

Pin35(A0)=1, Pin37(A2)=1 高电平带锁存

Pin36(A1)=0, Pin38(A3)=0, Pin39(A4)=0, 低电平带锁存。

地址线带锁存，所以可以用万用表测量。

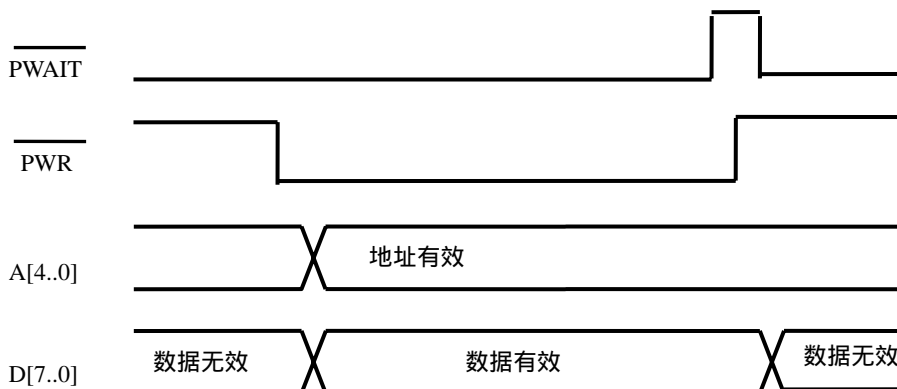
数据线状态：只有在 PRD 信号为低电平的时候有效，否则是无效数据

Pin16(D0)=1, Pin17(D1)=1, Pin22(D4)=1 高电平

Pin18(D2)=0, Pin19(D3)=0, Pin23(D5)=0, Pin24(D6)=0, Pin27(D7)=0 低电平

数据线没有锁存，所以只能用示波器跟踪。

## 3. 有等待周期的 I/O 写时序图



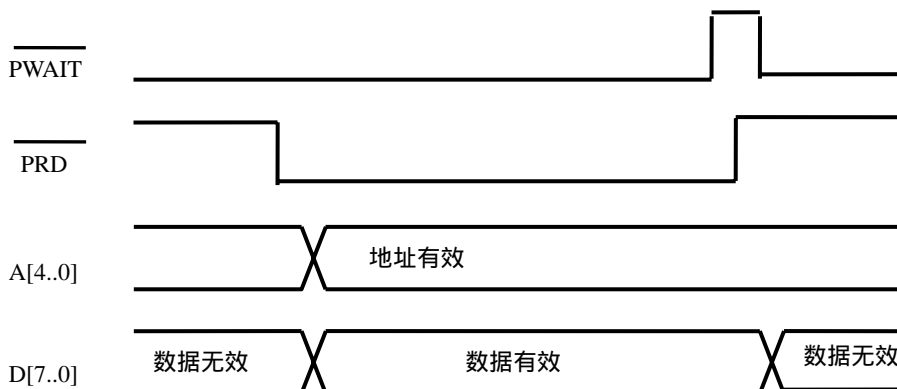
有等待周期的应用一般是以单片机慢速处理器作为主控芯片时采用。

这个时序与无等待周期的时序只有一个区别，nPWAIT 信号为低电平时，nPWR 输出的低电



平信号可以延长时间，这对于单片机的处理来说就相当于插入了等待时间。

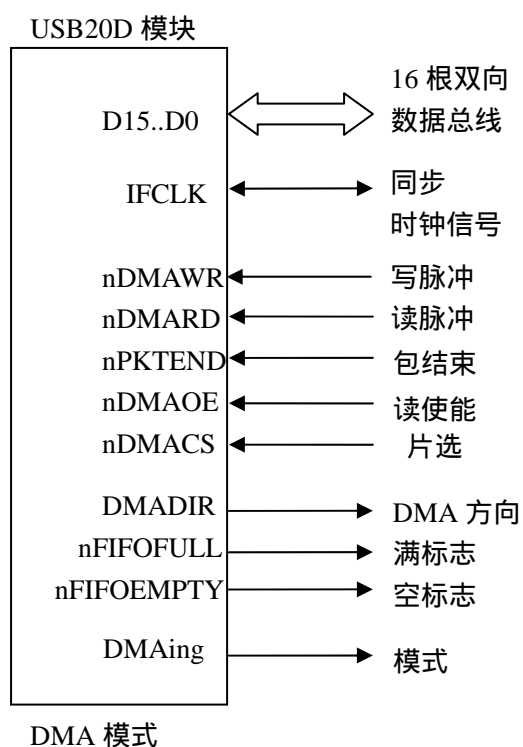
#### 4. 有等待周期的 I/O 读时序图



同上所述，这个时序与无等待周期的读时序只有一个区别，nPWAIT 信号为低电平时，nPRD 输出的低电平信号可以延长时间，这对于单片机的处理来说就相当于插入了等待时间。

#### 问题 5：USB20D 模块的 DMA 模式解释

DMA 模式是 USB20D 模块的最大优点，USB2.0 接口的高速数据传输就是通过 DMA 方式体现的，我们知道 USB2.0 的接口最大速率为 480Mbit/sec，理论字节速度为 48Mbyte/Sec。通过用目前双核电脑测试，USB20D 模块的 DMA 有效传输速度可以达到 35Mbyte/Sec。有效速度是不包括协议包的数据的，所以比理论速度慢一些，实际上接口速度仍是 480Mbit/sec。



USB20D 管脚	信号
Pin16~Pin19	D0~D3
Pin22~Pin25	D4~D7
Pin35~Pin39	D8~D12
Pin2	D13
Pin3	D14
Pin4	D15
Pin12	IFCLK
Pin9	nDMAWR
Pin8	nDMARD
Pin33	nPKTEND
Pin31	nDMAOE
Pin34	nDMACS
Pin32	DMADIR
Pin27	nFIFOFULL
Pin28	nFIFOEMPTY
Pin4	DMAing



在 DMA 模式下涉及到 26 根信号线，其中数据线 16 根，9 根信号线，1 根辅助线。箭头方向代表了信号的方向，信号的方向是以 USB20D 模块为参考系说明的，例如 nDMARD 是外部输入给 USB20D 模块的信号，nFIFOFULL 是 USB20D 模块输出的信号。信号名称前的 n 表示该信号低电平有效。

16 根双向数据线：低 8 位 (D7..D0) 与地址 I/O 模式共享，高 8 位是图中用红色标出 (D15..D8) 与 I/O 模式下的控制信号线复用的，这一点在设计电路时一定要注意到。

IFCLK 同步时钟 :DMA 模式有同步和异步两种方式，在同步方式下，一定要用到 IFCLK 信号，读写信号都是与 IFCLK 的沿同步的。IFCLK 既可以由 USB20D 模块产生 48MHz 或者 30MHz 的时钟输出，也可以由用户自己输入时钟，所以该信号是双向的。对于异步方式 IFCLK 无效。

nDMAWR 写信号：“写”是指用户通过 USB20D 模块的数据总线以 DMA 模式向电脑上传数据，所以是模块的输入信号，由用户生成。

nDMARD 读信号：“读”是指用户以 DMA 模式读取从电脑 USB 接口输出到 USB20D 模块的数据总线上的数据，nDMARD 也是模块的输入信号，由用户生成。

nDMAOE 读使能信号：从电脑 USB 接口输出到 USB20D 模块的数据总线上的数据在此信号有效时，数据总线上的数据才真正有效，当数据总线与其它设备复用时可以当作一个总线三态开关。

nDMACS 片选信号：在 DMA 模式下，该信号必须有效。

nPKTEND 包结束信号：在上传最后一包数据时，由用户处理器发送此信号通知本次 DMA 上传数据结束，这与 USB20D 模块内部缓冲 FIFO 的工作原理有关系。

对于 USB20D 模块的输出信号，结合下面的 DMA 模型和时序说明。特别注意 USB 数据的传输是半双工的串行数据，就是说在同一时刻只能有一个方向的数据。

站在 USB20D 的角度来说，有四种情况：同步 DMA 写数据、同步 DMA 读数据、异步 DMA 写数据、异步 DMA 读数据。

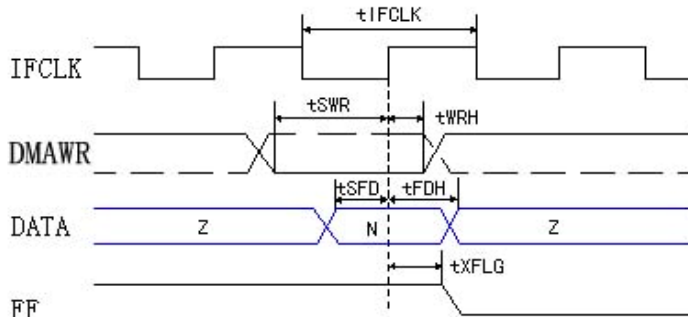
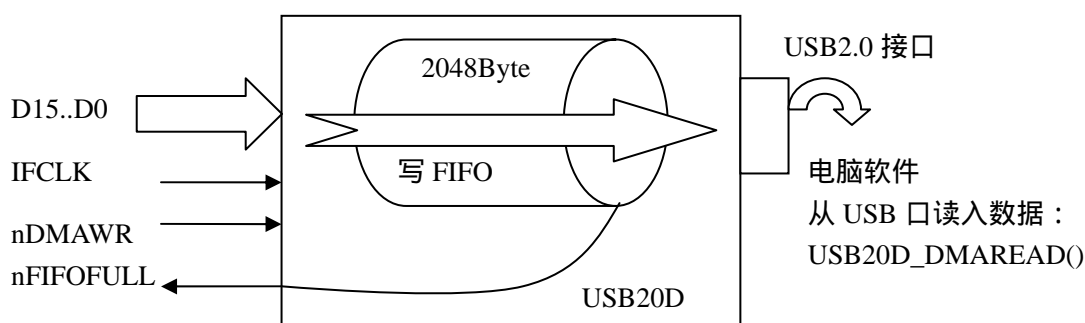
在 USB20D 模块内部有两个独立的 FIFO（先进先出）存储器，大小都为 2048 字节。从 USB20D 数据总线上输入或输出的数据实际上都经过了各自的 FIFO，就象一根水管一样，灌水是写数据，出水是读数据，在此水管的长度是 2K 字节。



## 1. 同步 DMA 写数据 FIFO 模型及时序图：

数据总线可以设置为 16 位，也可以设置为 8 位，设置由上位机软件调用 USB20D 的 API 函数 USB20D\_StartDMA；IFCLK 同步时钟可以通过调用函数 USB20D\_SetDMAClk 设置为由 USB20D 模块本身输出还是由外部输入；nDMAWR 信号与 IFCLK 信号要同步，具体关系参考以下的时序图。当 nFIFOFULL 信号（简化为 FF）变低时，表明 USB20D 模块的写缓冲区 FIFO 的 2K 字节已经满了，这时外部再也写不进数据了，强行写入的数据的话数据就丢失了，只有当上位机软件通过函数 USB20D\_DMAREAD() 读走数据后，nFIFOFULL 标志将重新变为高电平。所以用户的下位机程序在向 USB20D 模块写入数据之前一定要先判断 nFIFOFULL 的标志信号。

从下面的模型图中可以看出，数据流的方向是“写”入 USB20D 模块的 FIFO 后，又从 USB 接口“读”入到电脑内存中。有些用户对读写的概念不太清楚的话，可以仔细看一下这个图示说明。



同步DMA写时序图

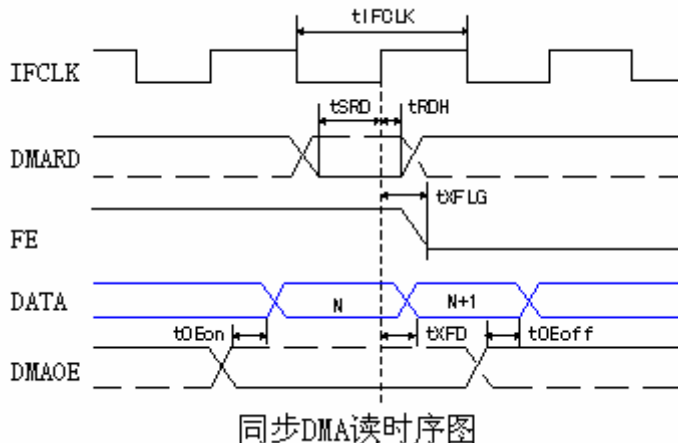
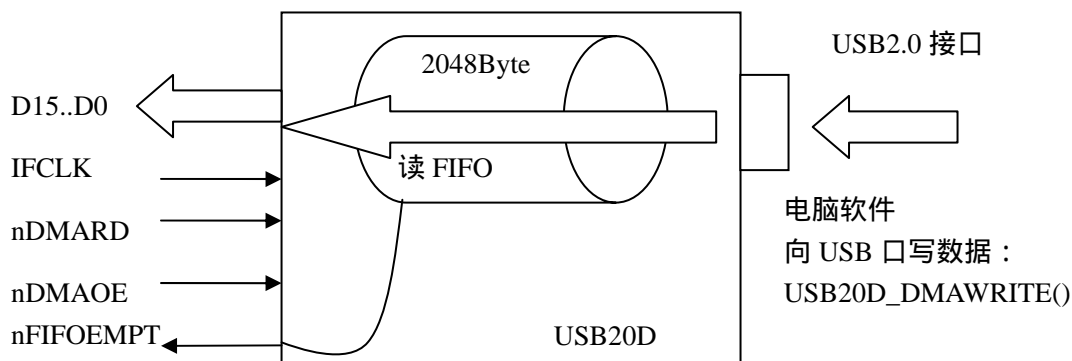
时序图中 FF 表示 FIFOFULL 信号线，用户产生的同步 DMAWR 信号的低电平应该包括一个 IFCLK 的低电平脉冲，同时在 IFCLK 的上升沿之前写入数据比较可靠。

## 2. 同步 DMA 读数据 FIFO 模型及时序图：

数据总线可以设置为 16 位，也可以设置为 8 位，设置由上位机软件调用 USB20D 的 API 函数 USB20D\_StartDMA；IFCLK 同步时钟可以通过调用函数 USB20D\_SetDMAClk 设置为由 USB20D 模块本身输出，还是由外部输入；nDMARD 信号与 IFCLK 信号要同步，具体关系参考以下的时序图。当 nFIFOEMPTY 信号（简化为 FE）变低时，表明 USB20D 模块的读缓冲区 FIFO 的 2K 字节已经空了，这时外部再也读不出数据了，只有当上位机软件通过函数 USB20D\_DMAWRITE() 写入数据后，nFIFOEMPTY 标志将重新变为高电平。所以用户的下位机程序在从 USB20D 模块读出数据之前一定要先判断 nFIFOEMPTY 的标志信号。



从下面的模型图中可以看出,数据流的方向是电脑软件从 USB 接口“写”入到 USB20D 模块的读 FIFO 后,又从 FIFO “读”出。有些用户对读写的概念不太清楚的话,可以仔细看一下这个图示说明。

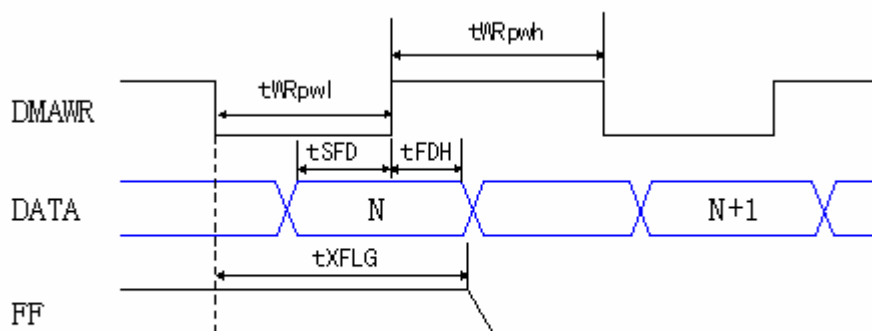
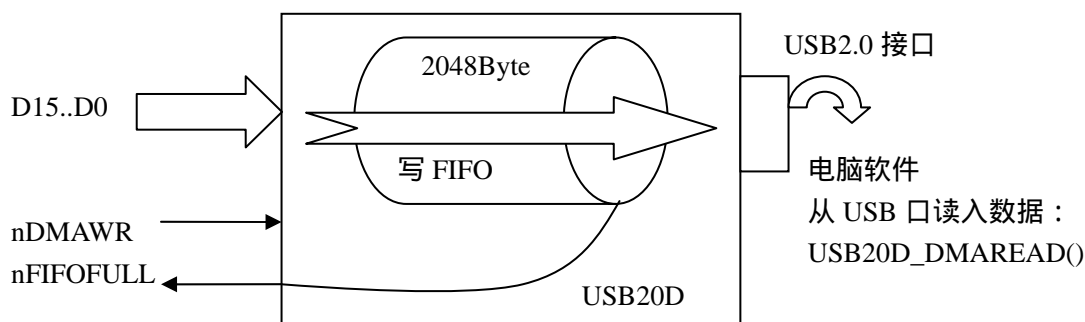


时序图中 FE 表示 FIFOEMPTY 信号线,用户产生的同步 DMARD 信号的低电平应该包括一个 IFCLK 的低电平脉冲,同时在 IFCLK 的上升沿读入数据比较可靠。在 DMA 读数据的过程中可以把 nDMAOE 一直置为低电平,不必在读过程中随 nDMARD 脉冲变化。

### 3. 异步 DMA 写数据 FIFO 模型及时序图：

数据总线可以设置为 16 位,也可以设置为 8 位,设置由上位机软件调用 USB20D 的 API 函数 USB20D\_StartDMA; IFCLK 同步时钟可以通过调用函数 USB20D\_SetDMAClk 设置为由 USB20D 模块本身输出,还是由外部输入;与同步方式不同的是 nDMAWR 信号与 IFCLK 信号无关,具体关系参考以下的时序图。当 nFIFOFULL 信号(简化为 FF)变低时,表明 USB20D 模块的写缓冲区 FIFO 的 2K 字节已经满了,这时外部再也写不进数据了,强行写入的数据的话就数据就丢失了,只有当上位机软件通过函数 USB20D\_DMAREAD()读走数据后,nFIFOFULL 标志将重新变为高电平。所以用户的下位机程序在向 USB20D 模块写入数据之前一定要先判断 nFIFOFULL 的标志信号。

从下面的模型图中可以看出,数据流的方向是“写”入 USB20D 模块的 FIFO 后,又从 USB 接口“读”入到电脑内存中。有些用户对读写的概念不太清楚的话,可以仔细看一下这个图示说明。



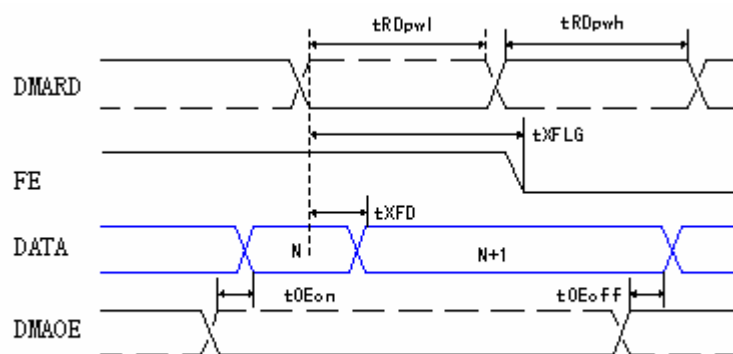
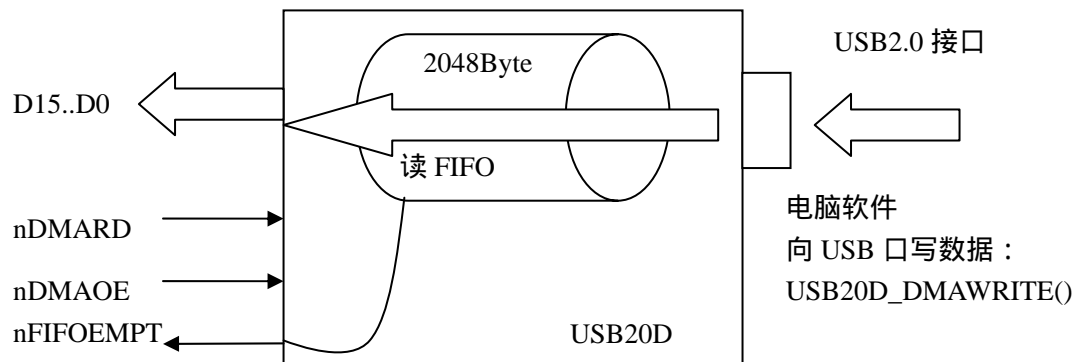
异步DMA写时序图

时序图中 FF 表示 FIFOFULL 信号线，用户产生的 MAWR 信号的频率决定了数据的写入速度，同时在 DMAWR 的上升沿之前写入数据比较可靠。

#### 4. 异步 DMA 读数据 FIFO 模型及时序图：

数据总线可以设置为 16 位，也可以设置为 8 位，设置由上位机软件调用 USB20D 的 API 函数 `USB20D_StartDMA`；IFCLK 同步时钟可以通过调用函数 `USB20D_SetDMAClk` 设置为由 USB20D 模块本身输出，还是由外部输入；与同步方式不同的是 nDMARD 信号与 IFCLK 信号无关，具体关系参考以下的时序图。当 nFIFOEMPTY 信号（简化为 FE）变低时，表明 USB20D 模块的读缓冲区 FIFO 的 2K 字节已经空了，这时外部再也读不出数据了，只有当上位机软件通过函数 `USB20D_DMAWRITE()` 写入数据后，nFIFOEMPTY 标志将重新变为高电平。所以用户的下位机程序在从 USB20D 模块读出数据之前一定要先判断 nFIFOEMPTY 的标志信号。

从下面的模型图中可以看出，数据流的方向是电脑软件从 USB 接口“写”入到 USB20D 模块的读 FIFO 后，又从 FIFO“读”出。有些用户对读写的概念不太清楚的话，可以仔细看一下这个图示说明。



异步DMA读时序图

时序图中 FE 表示 FIFOEMPTY 信号线 (FE), 用户产生的 DMARD 信号的频率决定了数据的读出速度, 同时在 DMARD 的上升沿读数据比较可靠。在 DMA 读数据的过程中可以把 nDMAOE 一直置为低电平, 不必在读过程中随 nDMARD 脉冲变化。