

《计算机组成原理》

（第九讲）

厦门大学信息学院软件工程系 曾文华

2022年5月19日

目录

- 第1章 计算机系统概论
- 第2章 数据信息的表示
- 第3章 运算方法与运算器
- 第4章 存储系统
- 第5章 指令系统
- 第6章 中央处理器
- 第7章 指令流水线
- 第8章 总线系统
- 第9章 输入输出系统



第9章 输入输出系统

- 9.1 输入输出设备与特性
- 9.2 I/O接口
- 9.3 数据传输控制方式
- 9.4 程序控制方式
- 9.5 程序中断控制方式
- 9.6 DMA方式
- 9.7 通道方式
- 9.8 常见I/O设备

9.1 输入输出设备与特性

- **输入输出系统**主要用于实现CPU与外部设备、外部设备与主存之间的信息交换。
- 输入输出系统是典型的**软硬件协同系统**，既包括I/O设备、I/O接口、总线、I/O管理部件等I/O硬件系统，也包括驱动程序、软件访问接口、用户程序等I/O软件系统。
- **输入输出设备**（外部设备）：
 - 输入设备：键盘、鼠标、扫描仪、摄像头等。
 - 输出设备：显示器、打印机等。
 - 输入输出设备：磁盘、网卡等。
- 输入输出设备的特性：
 - **异步性**：CPU与外部设备速度相差巨大，两者之间必须采用异步的方式进行数据交换。
 - **实时性**：不论是慢速的设备，还是高速的设备，设备准备好数据后，CPU都应及时处理；如键盘、鼠标按下后，CPU应及时响应；高速设备的数据如果得不到及时处理就会丢失；现场测试和实时控制设备的信息如果不能及时处理，可能导致严重的灾难。
 - **独立性**：外部设备应采用标准的总线接口与CPU进行连接，使输入输出与具体的设备类型无关，这便是输入输出的独立性。

9.2 I/O接口

9.2.1 I/O接口的功能
9.2.2 I/O接口的结构
9.2.3 I/O接口的编址
9.2.4 I/O接口的软件
9.2.5 I/O接口的分类

• 9.2.1 I/O接口的功能

- 计算机中的所有I/O设备均使用**I/O接口**（总线接口）与总线相连，CPU使用设备地址经总线与I/O接口通信来访问I/O设备。
- I/O接口应具有以下的功能：
 - ① **设备寻址**：接收来自总线的地址信息，经译码电路，选择对应外部设备中的寄存器或存储器。
 - ② **数据交互**：实现外部设备、主存与CPU之间的数据交换，这也是接口最基本的功能。
 - ③ **设备控制**：传送CPU命令。
 - ④ **状态检测**：反映外部设备的工作状态。
 - ⑤ **数据缓冲**：匹配CPU与外部设备的速度差距。
 - ⑥ **格式转换**：实现数据格式转换或逻辑电平信号转换。
- 此外，接口还应具有**中断**、**时序控制**和**数据检错**、**纠错**等功能。

• 9.2.2 I/O接口的结构

- I/O接口内部主要包括总线接口和内部接口两部分（图9.1）：
 - **总线接口**：连接总线的总线接口必须按总线标准进行设计，这部分逻辑为接口的标准部分。
 - **内部接口**：连接设备的内部接口逻辑因设备而异，是非标准的。
- I/O接口应包括以下基本的功能部件：
 - ① **数据缓冲寄存器（DBR）**
 - ② **设备状态寄存器（DSR）**
 - ③ **设备命令寄存器（DCR）**
 - ④ **设备存储器**：例如显卡中的显存。
 - ⑤ **地址译码器**
 - ⑥ **数据格式转换逻辑**：进行串并或并串传送的转换。

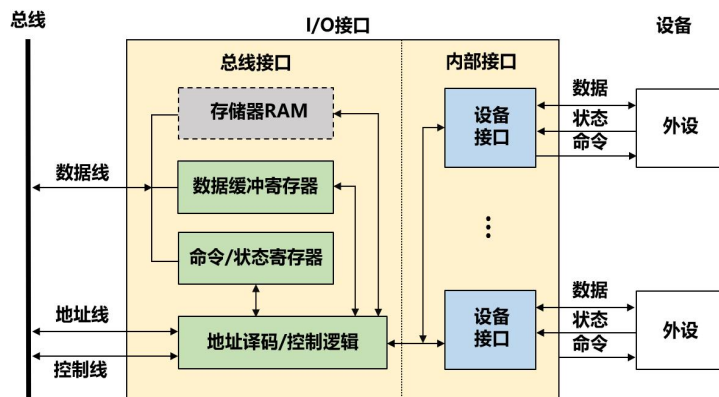


图9.1 I/O接口通用结构

• 9.2.3 I/O接口的编址

- I/O接口中的命令寄存器、状态寄存器、数据缓冲寄存器、设备存储器，都由CPU进行统一的设备地址分配，并通过对应的设备地址访问。

- 通常有两种编址方法：统一编址、独立编址。

– 1、统一编址

- 也称内存映射编址，Memory-mapped，外部设备与内存地址统一编址，两者在逻辑上处于同一个地址空间，通过不同的地址区域来区分是访问内存，还是访问外部设备；图9.2。
- 统一编址不需要设置专用的I/O指令，访存指令就可以访问外部设备。
- 由于I/O接口中的数据是动态变化的，因此统一编址中的I/O地址空间不能使用cache进行缓存，否则CPU无法了解设备状态的实时变化；在C语言中接口变量应该声明为volatile型，表明该变量是会经常自动变化的。

MIPS汇编代码

```
lw $t0,0x00000004    #从ROM读取一个字
sw $t0,0x00000004    #写ROM（会产生总线错误）
lbu $t0,0x00010001   #从内存读取一个字
sb $t0,0xff000002    #写一个字到内容
lbu $t0,0xffff0000   #从I/O设备读一个字
sb $t0,0xffff0004    #写一个字到I/O设备
```

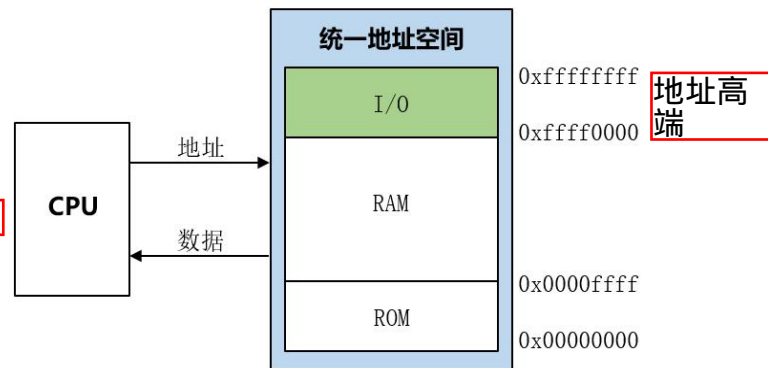


图9.2 I/O设备统一编址地址空间

– 2、独立编址

- 也称**端口映射编址**，Port-mapped，I/O地址空间与主存地址空间相互独立。
- **独立编址**需要使用特殊的**I/O指令**访问外部设备。
- 例如：80386处理器的主存空间为00000000H ~ FFFFFFFFH，4GB；而I/O地址空间是0000H ~ FFFFH，共64KB；用MOV指令访问内存，用**IN/OUT**指令访问外部设备。
- 表9.1为个人计算机中常见的I/O端口；图9.3为Windows系统（设备管理器）中的I/O编址。

表9.1 个人计算机中常见I/O端口

I/O地址范围（十六进制）	IRQ中断号	设备
0000 ~ 000F	00000004	DMA控制器
0020 ~ 0021	无	中断控制器
0040 ~ 0043	00000004	计时器
0060 ~ 0064		PS/2鼠标、键盘
0070 ~ 0071	00000008	CMOS实时时钟
02FB ~ 02FF	00000004	串口
F060 ~ F07F	FFFFFFFE	SATA控制器
03D0 ~ 03DF	FFFFFFFB	显卡控制器



图9.3 Windows系统中的I/O编址

• 9.2.4 I/O接口的软件

- 现代计算机中，用户并不能直接访问设备，必须通过操作系统**间接访问设备**。
- 操作系统中的I/O软件主要包括3个层次（图9.4）：
 - ① **与操作系统无关的I/O库**：如C语言中的标准I/O库stdio.h，包括printf()、scanf()、getchar()、putchar()、fopen()、fseek()、fread()、fwrite()、fclose()等函数；用户程序通过调用I/O库中的函数来访问设备，这些函数与操作系统无关，I/O库通常工作在用户态下。
 - ② **与设备无关的操作系统调用库**：如UNIX操作系统中的open()、read()、write()、seek()、ioctl()、close()等函数；这些函数与设备无关，屏蔽了设备的具体访问细节，向用户提供了统一的I/O调用接口。
 - ③ **独立的设备驱动程序**：设备驱动程序通过具体的I/O指令或访存指令，访问I/O接口中的数据缓冲寄存器DBR、命令寄存器DCR、状态寄存器DSR，与具体设备进行数据和命令交互。

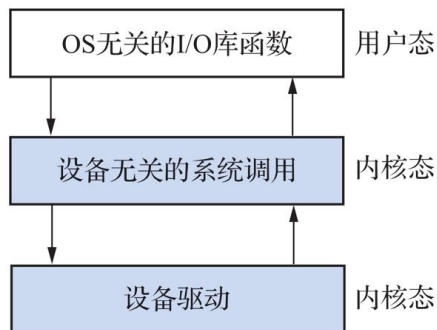


图9.4 操作系统中的I/O软件层次

• 9.2.5 I/O接口的分类

- ① 按数据传送方式分为：**并行接口、串行接口**；SCSI、IDE属于并行接口，SAS、SATA、USB属于串行接口。
- ② 按接口的灵活性分为：**可编程接口、不可编程接口**。
- ③ 按通用性分为：**通用接口、专用接口**；USB属于通用接口，SATA属于专用接口。
- ④ 按总线传输的通信方式分为：**同步接口、异步接口**。
- ⑤ 按访问外部设备的方式分为：**直接传送方式接口、程序控制方式接口、程序中断方式接口、DMA接口、通道处理机接口**。

9.3 数据传输控制方式

控制方式需要程序启动！

• 1、程序控制方式

- **程序控制方式**：首先，通过设置I/O接口的命令寄存器启动设备；设备准备的过程中，CPU通过读取I/O接口中的状态寄存器，查询设备是否已就绪，根据查询结果决定下一步操作究竟是进行数据传送，还是等待。
- 程序控制方式也称为**程序查询方式**。
- 程序控制方式的I/O接口设计简单，但是CPU与外部设备只能**串行工作**，CPU浪费大量时间进行查询和等待，系统效率较低。

• 2、程序中断控制方式

外设准备数据时，CPU在处理别的事务

- **程序中断控制方式**：CPU启动外部设备后，不再查询外部设备的状态，当外部设备准备好后，主动向CPU发出中断请求；CPU响应中断请求，暂停正在执行的程序，并调用相应的中断服务程序，完成CPU与外部设备的一次信息传输。

返回断点（第n+1）

- 程序中断控制方式中，CPU与外部设备是**并行工作**，CPU利用率高。

• 3、直接存储器访问方式

- **直接存储器访问方式**：DMA方式，Direct Memory Access。
- DMA方式中，由**DMA控制器（DMAC）**临时代替CPU控制总线，控制外部设备与内存之间进行直接的数据交换，信息传送不再经过CPU寄存器中转；DMA方式主要用于存储器与外部设备（如磁盘）之间的大量数据传送。

• 4、通道方式

- 为进一步减少CPU被I/O操作中中断的次数，提高CPU效率，出现了**通道技术（通道方法）**，由通道分担CPU的I/O管理，能有效提高系统效率。
- 通道拥有独立的**通道指令系统**，可以通过执行通道程序来完成CPU指定的I/O任务。

• 5、外围处理器方式

- **外围处理机方式**（PPU，Peripheral Processor Unit）是通道方式的进一步发展，通常用于大中型计算机系统中。

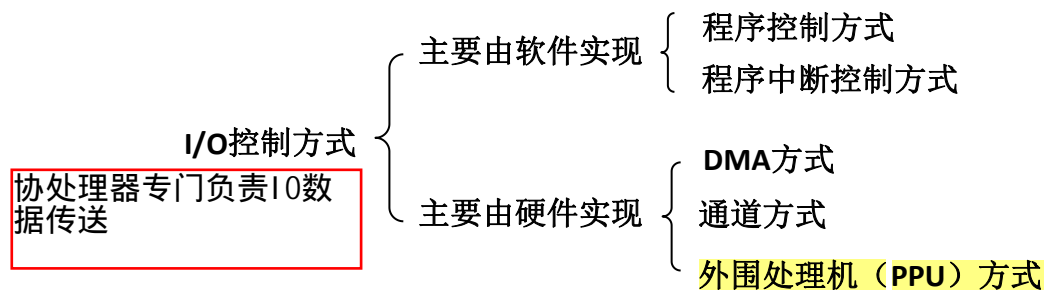


图9.5 I/O控制方式

9.4 程序控制方式

9.4.1	简单设备程序查询流程
9.4.2	复杂设备程序查询流程
9.4.3	程序查询特点

• 9.4.1 简单设备查询流程

- 程序控制方式（Programed I/O）是最原始的、最简单的方式。
- 程序控制方式又分为程序查询方式（也称为轮询方式，Polling）和直接传送方式两种；程序查询方式属于有条件传送方式，直接传送方式属于无条件传送方式。
- 表9.2为键盘和字符显示等简单设备的I/O地址。

先判断状态，再确定传输

表9.2 设备I/O地址

设备	寄存器（8位）	内存映射地址	端口映射地址	备注
键盘设备	数据缓冲寄存器DBR	0xFFFF0004	0x0004	
键盘设备	设备状态寄存器DSR	0xFFFF0000	0x0000	最低位为Ready位
字符显示设备	数据缓冲寄存器DBR	0xFFFF000C	0x000C	
字符显示设备	设备状态寄存器DSR	0xFFFF0008	0x0008	最低位为Ready位

— **键盘**设备的程序查询流程如图9.6a所示，其MIPS和x86查询程序代码如下：

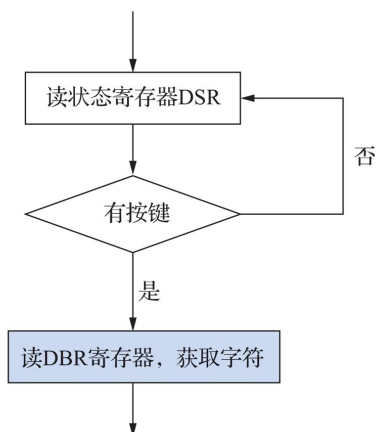
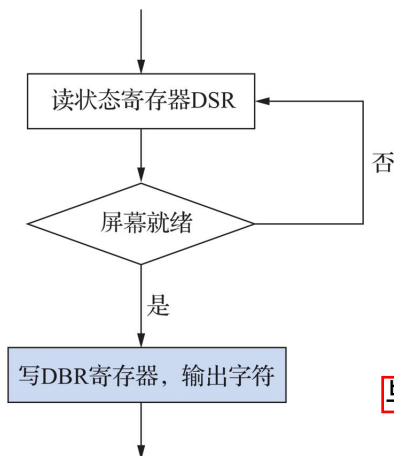


图9.6a 键盘程序查询流程

keyPoll:	<pre> lui \$t0,0xffff0000 lbu \$t2,(\$t0) andi \$t2,\$t2,1 beq \$t2,\$0,keyPoll lui \$t0, 0xffff0004 lbu \$s0,(\$t0) </pre>	<pre> #载入键盘DSR内存映射地址至\$t0 #载入键盘DSR的值至\$t2 #获取最低位Ready位 #如果Ready==0，则继续查询 #载入键盘DBR内存映射地址至\$t0 #载入键盘DBR中的字符数据至\$s0 </pre>
-----------------	---	---

keyPoll:	<pre> MOV DX,0x0000 IN AL,DX TEST AL,1 JZ keyPoll MOV DX,0x0004 IN AL,X </pre>	<pre> #载入键盘DSR端口地址至DX #载入键盘DSR的值至AL #测试就绪位Ready #如果Ready==0，则继续查询 #载入键盘DBR端口地址至DX #载入键盘DBR中的字符数据至AL </pre>
-----------------	--	--

— **字符终端**设备的程序查询流程如图9.6b所示，其MIPS和x86查询程序代码如下：



keyPoll:	<pre> lui \$t0,0xffff0000 lbu \$t2,(\$t0) andi \$t2,\$t2,1 beq \$t2,\$0,keyPoll lui \$t0, 0xffff0004 sb \$s0,(\$t0) </pre>	<pre> #载入字符终端DSR内存映射地址至\$t0 #载入字符终端DSR的值至\$t2 #获取最低位Ready位 #如果Ready==0，则继续查询 #载入字符终端DBR内存映射地址至\$t0 #将\$s0中的字符数据输出至DBR </pre>
-----------------	--	--

keyPoll:	<pre> MOV DX,0x0000 IN AL,DX TEST AL,1 JZ keyPoll MOV DX,0x0004 OUT DX,AL </pre>	<pre> #载入字符终端DSR端口地址至DX #载入字符终端DSR的值至AL #测试就绪位Ready #如果Ready==0，则继续查询 #载入字符终端DBR端口地址至DX #将AL中的字符数据输出至DBR </pre>
-----------------	--	---

与1相与，相当于屏蔽前7位

状态信息送寄存器

数据传送

图9.6b 字符终端程序查询流程

• 9.4.2 复杂设备程序查询流程

- 图9.7为复杂设备的程序查询流程。
- CPU首先查询设备的状态 (①)，如果设备就绪 (②)，就通过总线向I/O接口发送命令与参数，启动设备 (③)。
- 设备收到命令后，即刻去准备或处理；设备准备好后，会将状态寄存器中相关位置位，表示命令执行完毕或准备就绪。
- 而CPU在启动设备后，就开始不断查询设备状态 (④)，当设备就绪 (⑤)，即可进行实际的数据传输 (⑥)。

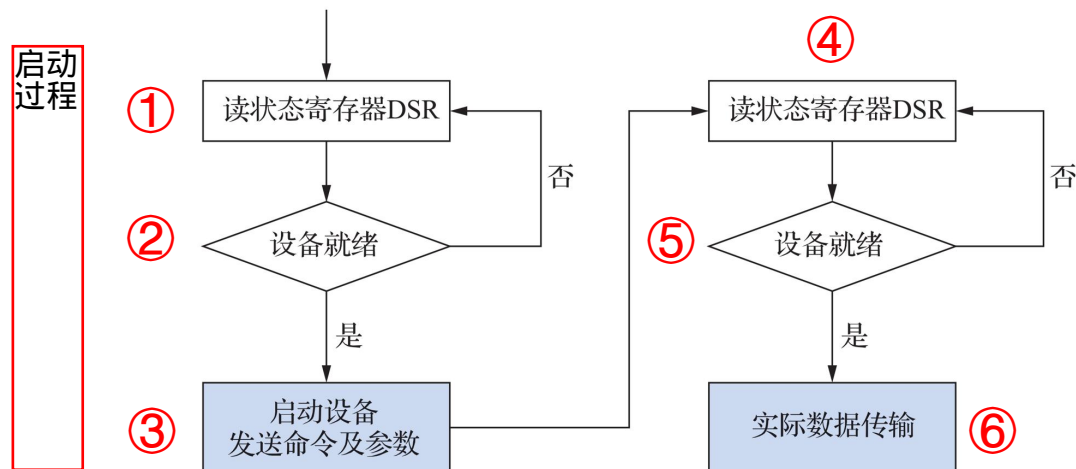


图9.7 复杂设备的程序查询流程

• 9.4.3 程序查询特点

- 程序查询方式中，CPU要不断查询I/O接口中的状态寄存器DSR，当设备就绪时才能进入下一操作，否则继续查询。
- 程序查询（也称为轮询）有两种策略：忙等待和定时轮询。
- 1、忙等待（Busy-waiting）
 - 忙等待方式也称为独占式查询，程序运行轨迹如图9.8a所示；在设备准备数据阶段，CPU不能执行其他任务，称为忙等待状态（轮询等待，busy-waiting）。
 - 忙等待方式的缺点：CPU浪费了大量的时间进行轮询操作。

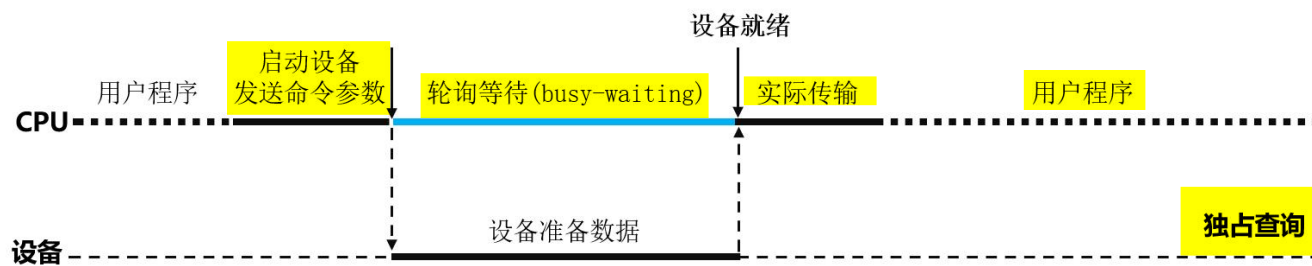


图9.8a 独占式查询方式的程序运行轨迹

– 2、定时轮询（Polling）

- **定时轮询**不需要反复查询，**程序运行轨迹**如图9.8b。
- CPU启动设备后，会启动一个定时中断（①）；然后挂起当前用户进程P1，并放入I/O等待队列，调度用户进程P2运行（②）；定时时间到后（③），CPU会执行定时中断服务程序，查询设备状态，如果设备准备好，则唤醒等待进程P1（④），否则将继续定时查询。
- 定时轮询方式的**优点**：CPU可以执行其他任务，避免轮询等待CPU时间的浪费，有效节约了CPU时间。
- 定时轮询中的定时时间间隔比较关键；**时间间隔过短**，则用于定时查询的中断服务开销浪费较多；**时间间隔过长**，外部设备数据可能得不到及时处理。

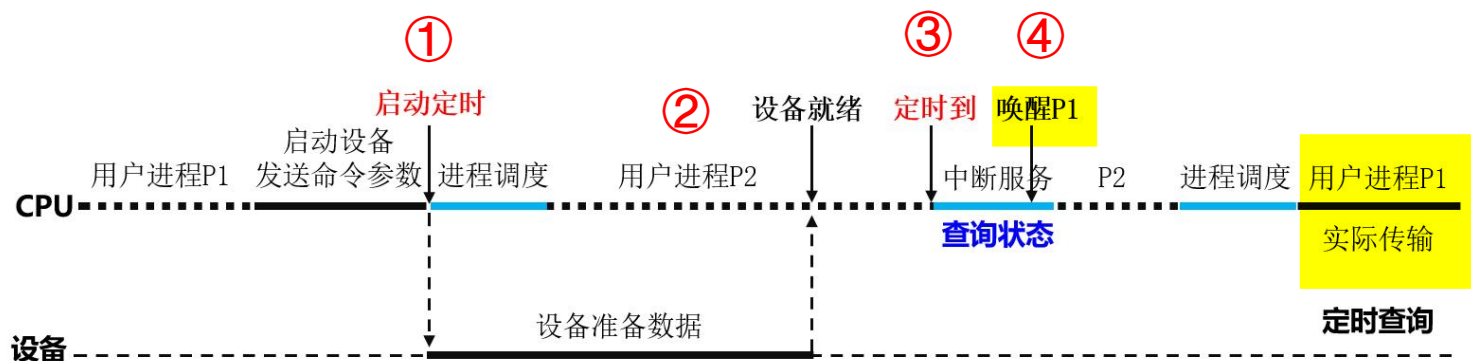


图9.8b 定时轮询方式的程序运行轨迹

- 例9.1 假设某程序查询方式的输入输出系统采用**定时轮询方式**，每次定时中断服务开销需要**400个**时钟周期，CPU的时钟频率为**200MHz**，包括**鼠标**和**硬盘**两个外部设备，求两种不同外部设备进行I/O操作时的**CPU时间占有率**。
- （1）鼠标以字节为单位进行数据传输，假设每秒必须进行**50次**轮询才能保证不会错过如何鼠标操作，每次轮询成功后的实际数据传输需要**13个**时钟周期。
- （2）硬盘以**512字节**的扇区为单位传输数据，启动阶段发送命令和参数需要**90个**时钟周期，实际传输阶段需要**1555个**时钟周期，CPU访问硬盘的速率为**20MB/s**。
- （3）如果硬盘以字节为单位进行数据传输，其他参数不变，会发生什么情况？
- 解：
- （1）
 - 时钟周期 $T = 1/200\text{MHz}$
 - CPU每秒用于鼠标I/O操作的时间 = $(400+13) \times 50 \times T = 20650T = 0.00010325\text{s}$
 - CPU时间占有率 = $0.00010325\text{s} / 1\text{s} = 0.010325\%$

- 例9.1 假设某程序查询方式的输入输出系统采用**定时轮询方式**，每次定时中断服务开销需要**400**个时钟周期，CPU的时钟频率为**200MHz**，包括**鼠标**和**硬盘**两个外部设备，求两种不同外部设备进行I/O操作时的**CPU时间占有率**。
- （1）鼠标以字节为单位进行数据传输，假设每秒必须进行**50**次轮询才能保证不会错过任何鼠标操作，每次轮询成功后的实际数据传输需要**13**个时钟周期。
- （2）硬盘以**512**字节的扇区为单位传输数据，启动阶段发送命令和参数需要**90**个时钟周期，实际传输阶段需要**1555**个时钟周期，CPU访问硬盘的速率为**20MB/s**。
- （3）如果硬盘以字节为单位进行数据传输，其他参数不变，会发生什么情况？

• 解：

• （2）

- 硬盘每次传输**512**个字节，要达到CPU访问硬盘的速率**20MB/s**，则每秒传输的次数 = $20\text{MB}/512\text{B} = 20 \times 10^6 / 512 = 39062.5 \text{次/s}$
- 每次传输的开销（每次传输**512**个字节）= $(90+400+1555) \times T = 2045 \times (1/200\text{MHz}) = 0.000010225\text{s}$
- CPU时间占有率= $0.000010225\text{s} \times 39062.5 / 1\text{s} = 39.94\%$

- 例9.1 假设某程序查询方式的输入输出系统采用**定时轮询方式**，每次定时中断服务开销需要**400个**时钟周期，CPU的时钟频率为**200MHz**，包括**鼠标**和**硬盘**两个外部设备，求两种不同外部设备进行I/O操作时的**CPU时间占有率**。
- （1）鼠标以字节为单位进行数据传输，假设每秒必须进行**50次**轮询才能保证不会错过如何鼠标操作，每次轮询成功后的实际数据传输需要**13个**时钟周期。
- （2）硬盘以**512字节**的扇区为单位传输数据，启动阶段发送命令和参数需要**90个**时钟周期，实际传输阶段需要**1555个**时钟周期，CPU访问硬盘的速率为**20MB/s**。
- （3）如果硬盘以字节为单位进行数据传输，其他参数不变，会发生什么情况？
- 解：
- （3）
 - 如果硬盘以字节为单位进行数据传输，假设从接口读出一个字节转存至内存的开销是**15个时钟周期**，则512个字节的传输开销 = $(90+400+15) \times 512 \times T = 258560T = 0.0012928s$
 - 硬盘每次传输**512个**字节，要达到CPU访问硬盘的速率**20MB/s**，则每秒传输的次数 = $20MB/512B = 20 \times 10^6 / 512 = 39062.5 \text{次/s}$
 - CPU时间占有率 = $0.0012928s \times 39062.5 / 1s = 5005\%$
 - 即如果硬盘以字节为单位进行数据传输，即使CPU所有的时间都花在硬盘上，也达不到**20MB/s**的速率
 - 最高速率 = $512B/0.0012928s = 396039.6B/s = 0.396MB/s$
 - 因此，硬盘不适合轮询方式！

本章小结

习题 (P368-371)

Thanks