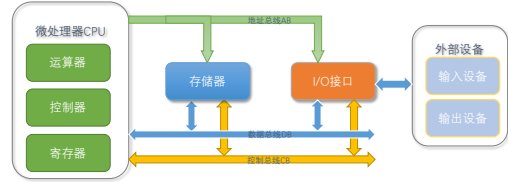


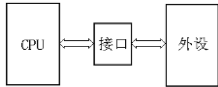
传送通道——接口

计算机系统



接口

- I/O接口是主机与外部设备（简称外设）之间所设置的逻辑控制部件，通过它实现主机与I/O设备之间的信息交换。



为什么需要接口?

- 一般情况下，存储器可以与总线直接相连，而外部设备却需要通过接口与CPU的总线相连，原因是由存储器和外设的特点决定：

存储器：

- 功能单一
- 传输方式单一（一次一个字或一个字节）
- 操作方式单一（读和写）
- 制造工艺与CPU相似，速度与CPU相匹配

为什么需要接口?

外设：

- 种类繁多（机械、机电、电子）
- 信号种类不一（A,D,开关量）
- 信号格式不同（串行、并行）
- 同一个时刻CPU通常只和一个外设交换信息
- 工作速度不同，且范围宽。如硬盘和打印机
- 工作时序不匹配，无法和CPU时序取得统一

接口

- 接口：解决以上差异，协调、匹配外设与主机正常工作的逻辑部件及相应控制软件。

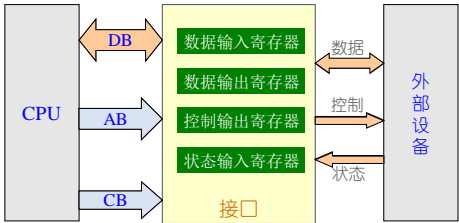
自顶向下-知识分解

自顶向下-知识分解

接口的任务

- 对数据提供缓冲（时间和电气性能上）：设置数据的寄存、缓冲逻辑；
- 信息格式相容性变换：如串并行的转换；电平转换、数 / 模或模 / 数转换等；
- 协调时序差异：提供“准备好”“空”“满”等状态信号；
- 提供地址译码或设备选择信号；
- 提供中断和DMA控制逻辑及管理。

外设通过接口与CPU的连接



访问接口的过程

- 打开相应端口：CPU先将地址信息发送到地址总线，将确定的控制信息发送到控制总线；
- 收发数据：CPU传输数据信息到数据总线上等待相应端口接收，或者CPU等待接口把指定端口的内容送到数据总线上。

几点说明

- 注意：地址是端口(寄存器)的地址，而不是接口部件的地址，一个接口部件包含多个端口，即多个地址。
- I/O端口即I/O接口的寄存器，接口中的每个寄存器都有一个端口地址，每个I/O接口都有一组寄存器。
- 数据输入和数据输出寄存器可以使用同一地址；控制输出和状态输入寄存器可以使用同一地址。

I/O端口的寻址方式

- 存储器映像寻址方式
 - 把每一个I/O端口都看作一个存储单元，并与存储单元一样统一编址，这样访问存储器的所有指令均可用来访问I/O端口，不用设置专门的I/O指令。
 - 优点：电路设计简单，由于I/O端口和存储器地址是相同的形式，因此不需要专门为I/O端口设计电路，可以与存储器地址访问硬件混合设计。
 - 缺点：增加程序设计难度，由于存储器地址和I/O端口在指令形式上没有区别，因此程序员在编写程序时需要更加小心，以避免对I/O端口的误操作。

I/O端口的寻址方式

- I/O单独编址方式
 - 对端口地址单独编址，使用专门的指令来访问这种具有独立地址空间的端口。
 - ◆ 优点：程序更清晰，可读性好；I/O指令长度短，执行速度快，不占用内存空间；I/O地址译码电路简单。
 - ◆ 缺点：指令系统中需专门的指令，且这些指令的功能没有访问存储器指令强；CPU需提供区分存储器读/写和I/O读写的控制信号。

数据传送方式之无条件传送

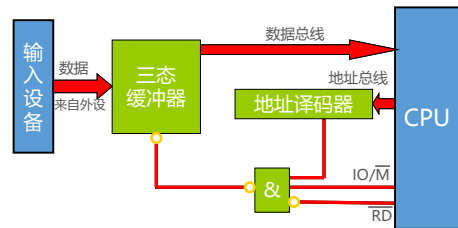
数据传送方式

- 程序方式
 - 无条件传送方式
 - 查询传送方式
- 中断传送方式
- 直接存储器存取(DMA) 控制方式

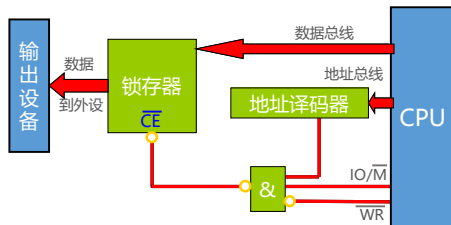
无条件传送方式

- 在数据传送过程中, 输入或输出数据一方不查询、判断对方的状态, 进行无条件的数据传送。CPU能够确信外设准备就绪, 就不用查询外设的状态而可以直接进行数据传输。
 - 驱动指示灯、继电器、启动电机等

无条件传送的输入方式



无条件传送的输出方式



数据传送方式之软件查询传送

查询传送方式

- CPU执行程序不断读取并测试外设的状态，如果外设处于准备好（输入）或空闲（输出）状态，则执行输入或输出指令，进行数据交换，否则等待。

完成一次数据传送的过程

- 1) CPU从状态端口读取外设的状态字;
- 2) CPU检测状态字对应位是否满足“就绪”条件;
- 3) 如不满足，则重复执行1) 2) 过程，直到条件满足;
- 4) 如果条件满足，表明外设就绪，则传送数据，同时I/O的状态复位。

查询传送方式

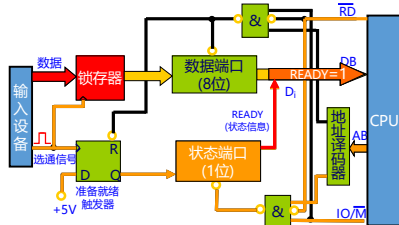
- 实现流程:
 - 在与外设进行传送数据前，CPU先查询外设状态，当外设准备好后，才执行I/O指令，实现数据传送
- 特点:
 - 1. CPU通过不断查询外设状态，实现与外设的速度匹配
 - 2. CPU的工作效率低



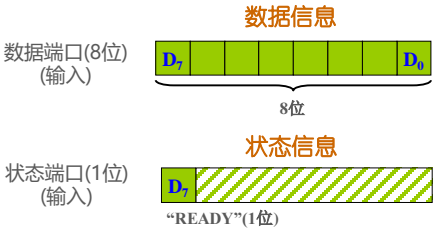
查询式输入方式

- 1. 输入设备准备好后，发送通信信号;
- 2. 数据进入数据端口，并使D触发器置1，从而使状态端口置位;
- 3. CPU从状态端口读入状态字;
- 4. CPU检测状态位，如果条件满足;
- 5. CPU从数据端口读入数据;
- 6. 清状态字。

查询式输入接口电路



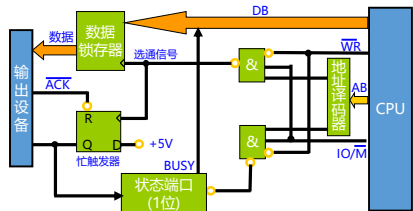
查询式输入时的数据和状态



查询式输出的过程

- 1. 将数据写入数据端口，同时将状态寄存器置位，防止CPU再次传送数据；
- 2. 外设读取数据；
- 3. 外设向接口发ACK信号，将状态位清零。

查询式输出接口电路

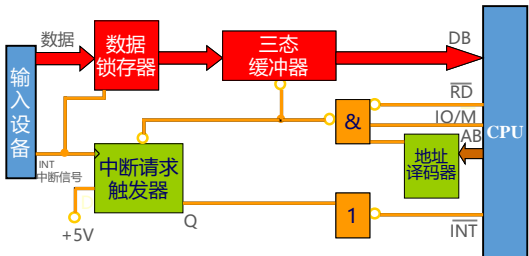


数据传送方式之中断方式传送

中断

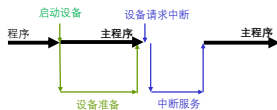
- 在CPU正常运行程序时，由于内部或外部某个非预料事件的发生，使CPU暂停正在运行的程序，而转去执行处理引起中断事件的程序，然后再返回被中断的程序，继续执行。这个过程就是中断。

中断传送方式输入接口电路



中断传送方式的特点

- CPU暂时中止现行程序的执行，转去执行为某个随机事件服务的中断处理子程序，处理完后自动恢复原程序的执行
- 实现主机和外设准备阶段的并行工作
 - 避免重复查询外设状态、提升工作效率



断点和中断现场

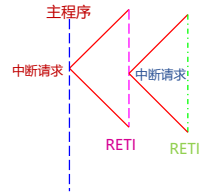
- **断点**:是指CPU执行的现行程序被中断时的下一条指令的地址,又称断点地址。
- **中断现场**:是指CPU转去执行中断服务程序前的运行状态,包括CPU内部各寄存器、断点地址等。

```
int H(int n, int a, int b) {
    if (n == 0)
        return b++;
    return ++a;
}

int main() {
    int rp, n=1, a=2, b=3;
    rp = H(n, a, b);
    printf("%d", rp);
    return 0;
}
```

中断系统的功能

- (1) 响应中断及返回
- (2) 能实现优先权排队
- (3) 能实现中断嵌套



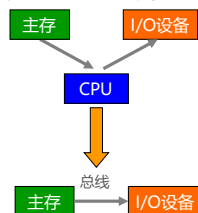
数据传送方式之DMA传送

直接存储器存取(DMA) 方式

- DMA(Direct Memory Access)直接存储器存取控制方式下, I/O设备是和存储器直接交换信息, 不需要CPU介入, 外设与存储器间的数据传输是在硬件的作用下完成的。
- 优点: 传输速度大幅提高。

直接存储器存取(DMA) 方式

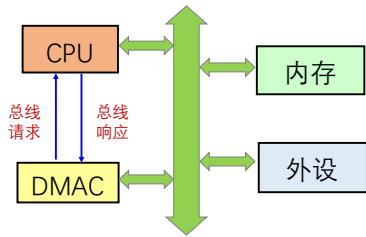
- DMA方式下，外设利用专门的接口电路直接和存储器进行高速数据传送，而不经CPU。数据的传输速度基本上决定于外设和存储器的速度。



直接存储器存取(DMA) 方式

- ### ■ 实现方法：
- 1.由专用接口芯片DMA控制器(称DMAC) 控制传送过程;
 - 2.当外设需传送数据时,通过DMAC向CPU发出总线请求;
 - 3.CPU发出总线响应信号,释放总线;
 - 4.DMAC接管总线,控制外设、内存之间直接数据传送.

DMA传送方式过程



直接存储器存取(DMA)方式

- DMA传送方式的特点
 - 1. 外设和内存之间, 直接进行数据传送, 不通过CPU, 传送效率高。适用于在内存与高速外设、或两个高速外设之间进行大批量数据传送。
 - 2. 电路结构复杂, 硬件开销较大。

常见的外部接口与分类

I/O接口分类

- 按照电路和设备的复杂程度, I/O 接口的硬件主要分为两大类:
 - I/O 接口芯片: 这些芯片大多都是集成电路, 通过CPU 输入不同的命令和参数, 并控制相关的I/O 电路和简单的外设作相应的操作, 常见的接口芯片如定时 / 计数器、中断控制器、DMA控制器、并行接口等。
 - I/O 接口控制卡: 有若干集成电路按一定的逻辑组成为一个部件, 或者直接与CPU 同在主板上, 或是一个插件插在系统总线插槽上, 通常称为外部接口。

I/O接口分类

- 按通信信号类型可分为数字接口和模拟接口
- 按通信方式可分为串行接口和并行接口

数字接口

- 数字接口指CPU与接口之间所传输的信息为数字信息, 用于视频信号传输的HDMI及DVI都是数字接口。



模拟接口

- 模拟接口指所传输的信息为模拟信息，当模拟信息要输入到计算机内进行处理或存储时，需要将模拟信息转换成数字信息，当计算机中的数字信息要通过模拟接口输出时，需要将数字信息转换成模拟信息。
- 数字信息和模拟信息之间的转换过程称为数模（D/A）转换和模数（A/D）转换。用于视频信号传输的模拟接口包括VGA接口、AV接口等。



VGA接口



AV接口

60

并行接口

- 并行接口是指数据的各位同时进行传输，数据传输通常以字节（8位）或字（16位）为单位。
- 其特点是传输速度快，适用于短距离传输。常见接口有LPT接口，用于连接打印机或扫描仪等设备。
- 随着技术的发展和USB等串行接口技术的普及，并行接口中的应用范围在逐渐缩小。



LPT接口

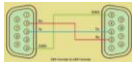
61

串行接口

- 串行通信是指数据按位顺序传送，其特点是通信线路简单，只要一对传输线就可以实现双向通信，并可以利用电话线，从而大大降低了成本，特别适用于远距离通信。
- 串行通信的基本单位为bps（bit per second），即每秒钟传送的二进制位数。
- 串行通信本身又分为异步通信与同步通信两种。串行接口RS-232，在工业控制、数据采集、仪器仪表等领域应用广泛。



RS-232



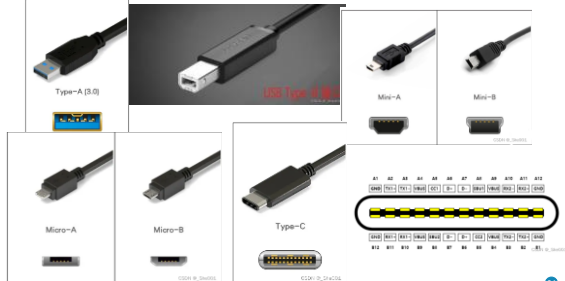
62

USB接口

- USB（通用串行总线，Universal Serial Bus）接口自1996年推出以来，已成功替代COM口和并口，成为21世纪大量计算机和智能设备的标准扩展接口和必备接口之一。USB接口的传输速率取决于具体的USB规范版本。例如，USB 1.0的传输速率只有1.5Mb/s，而USB 2.0的传输速率可以达到480Mbps。最新的USB 4.0版本的传输速率更是高达40Gbit/s。
- USB接口具有热插拔功能，支持即插即用，使用方便，传输速度快，连接灵活，独立供电等优点。
- USB接口有三种类型：USB Type-A（一般用于电脑设备）、USB Type-B（一般用于3.5in硬盘等设备）、USB Type-C（新型的USB类型接口，被广泛应用于各种电子设备，新版的USB4只采用了Type-C规格）。

63

USB接口



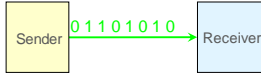
64

串行接口通信基础

65

串行接口通信基础

- 串行通信：通信双方按位进行，遵守时序的一种通信方式。
 - 多位数据通过同一条数据线按位依次传送
 - 因为数据线少，成本低，特别适用于远距离通信
- 由于要进行串并行转换串行接口电路相对复杂



串行接口通信基础

在串行通信时，收发双方必须考虑解决如下问题：

- 波特率——双方约定以何种速率进行数据的发送和接收
- 帧格式——双方约定采用何种数据格式
- 帧同步——接收方如何得知一批数据的开始和结束
- 位同步——接收方如何从位流中正确地采样到位数据
- 数据校验——接收方如何判断收到数据的正确性
- 差错处理——收发出错时如何处理

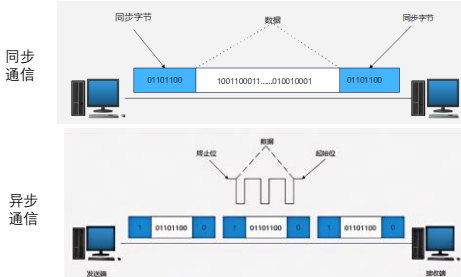
波特率 (Baud rate)

- 为了衡量串行通信的速度，描述数据传送速度的方式有：
 - 波特率定义为每秒传送信号的数量，单位为波特(Baud)。
 - 比特率定义为每秒传送二进制数的信号数(或每秒传送二进制码元的个数)，单位是 bps(bit per second)或b/s(位/秒)。
- 在串行通信中，传送的信号可能就是二进制、八进制或十进制等。只有在传送的信号是二进制信号时，波特率才与比特率在数值上相等。而在采用调制技术进行串行通信时，波特率就是描述载波信号每秒钟变化为信号的数量(又称为调制速率)。在这种情况下，波特率与比特率在数值上可能不相等。
 - 例如：电传打字机传输率为每秒10个字符/秒，每个字符包含11个二进制位，则数据传输率为：11位/字符×10个字符/秒=110位/秒=110bps=110波特(Baud)。

串行通信数据传送方式

- 同步：许多字符组成一个信息组，在每组信息的开始加上同步字符（不允许字符间有间隔），要求收发双方时钟（频率和相位）严格同步。
- 异步：按字符传输，字符可以不连续发送，每个字符的前后用一些数位来作为分隔位（两个字符之间的传输间隔任意），双方的收发时钟不要求严格同步。

同步通信和异步通信



同步通信

- 分为单同步、双同步，及外同步

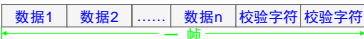
单同步



双同步

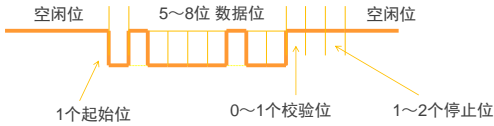


外同步



异步通信

- 异步传输以字符为单位，称为一个信息帧。
- 一帧包括起始位、数据位、校验位、停止位四部分。
- 异步传输信息冗余较大，例如：1+8+1+2中有效位数只有8位。

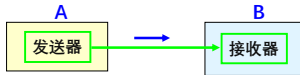


异步通信与同步通信的主要区别

	同步通信	异步通信
传输格式	面向比特的传输，每个信息帧中包含若干个字节	面向字符的传输，每个帧只包含一个字节
时钟	要求接受时钟和发送时钟同频同相，通过特定的时钟线路协调时序	不要求接受时钟和发送时钟完全同步，对时序要求较低
同步方式	从数据中抽取同步信息	通过字符起止的开始位和停止位同步
数据流	发送端发送连续的比特流	发送端发送完一个字节后，可经过任意长的时间间隔再发送下一个字节
校验字符	CRC 校验	奇偶校验
传输效率	控制字符开销较小，传输效率高	传输效率较低
通信结点	点对多点	点对点
应用	高速、实时设备，网络通信中，如：传感器接口、实时控制系统、数据库同步等	低速设备、网络通信中，如：http、网络聊天、分布式系统、工业控制领域等

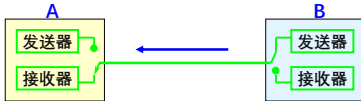
数据传送方向

- 按串行通信的数据传送方向，串行通信可分为单工、半双工、全双工三种。
- 单工传送方式：只能进行一个方向的数据传送。比如：电视系统和调频广播系统。



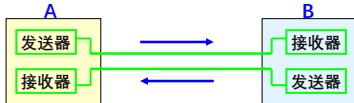
数据传送方向

- 半双工传送方式：可以进行两个方向的数据传送（双向），但不能同时进行双向传送；某一时刻只能进行一个方向的传送。比如：对讲机系统。



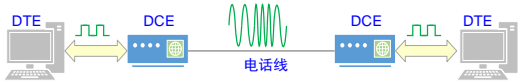
数据传送方向

- 全双工传送方式：收发双方通过两根单向信息线连接，分别负责数据的发送和接收，通信双方都能在同一时刻进行发送和接收操作，比如：电话。



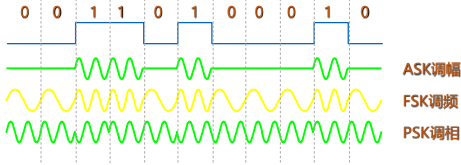
信号的调制与解调

- 调制：在利用模拟信号传输的通信系统中，发送方要利用调制器把数字信号转换成模拟信号，以两种不同频率的正弦波来表示“1”和“0”发送到通信线上，这一过程称为调制。
- 解调：接收方利用解调器把收到的模拟信号恢复成数字信号，这一过程称为解调。



信号的调制与解调

- 信号的调制有3种方法：调幅（AM）、调频（FM）和调相（PM）。



奇偶校验

- 串行数据在传输过程中，由于干扰可能引起信息的出错，例如，传输字符 'E'，其各位为：
0 1 0 0 1 0 1 = 45H
D₇ D₀
 - 由于干扰，可能使某位变为1，称为出现了“误码”
 - 如何发现传输中的错误，叫“校验”
 - 发现错误后，如何消除错误，叫“纠错”
- 最简单的校验方法是“奇偶校验”，即在传送字符的各位之外，再传送1位奇/偶校验位，分为奇校验或偶校验。

奇偶校验

- 奇校验：所有传送的数位（含字符的各数位和校验位）中，“1”的个数为奇数，如：
1 0 1 1 0 0 1 0 1
0 0 1 1 1 0 0 1 1
- 偶校验：所有传送的数位中，“1”的个数为偶数，如：
0 0 1 1 0 0 1 0 1
1 0 1 1 1 0 0 1 1

奇偶校验

- 说明：
 - 奇偶校验能够检测出信息传输过程中的部分误码，能检出奇数位误码，不能检出偶数位误码；
 - 奇偶校验不能纠错，在发现错误后，只能要求重发。
 - 但由于其实现简单，被广泛应用于各种数据传输和存储系统中，如串口通信、硬盘存储等。

CRC循环冗余校验

- CRC循环冗余校验（Cyclic Redundancy Check）是数据通信领域中最常用的一种差错校验码，其特征是信息字段和校验字段的长度可以任意选定。
- 循环冗余校验：在每个数据块（称之为帧）中加入一个FCS（Frame Check Sequence，帧检查序列）。FCS中包含了帧的详细信息，用于比较帧的正确性。如果数据有误，则再次发送。

CRC循环冗余校验

- 循环冗余校验码（CRC）的基本原理：在K位信息码后再拼接R位的校验码，整个编码长度为N位，因此，这种编码也叫（N，K）码。
- 对于一个给定的（N，K）码，可以证明存在一个最高次幂为N-K=R的多项式G(x)。根据G(x)可以生成K位信息的校验码，而G(x)叫做这个CRC码的生成多项式。
 - 任意一个由二进制位串组成的代码都可以和一个系数仅为‘0’和‘1’取值的多项式一一对应。
 - 例如：代码1010111对应的多项式为 $x^6+x^4+x^2+x+1$ ，而多项式为 $x^5+x^3+x^2+x+1$ 对对应的代码101111。

CRC循环冗余校验

■ 常用的CRC码的生成多项式有：

- $CRC8 = X^8 + X^5 + X^4 + 1$
- $CRC-CCITT = X^{16} + X^{12} + X^5 + 1$
- $CRC16 = X^{16} + X^{15} + X^5 + 1$
- $CRC12 = X^{12} + X^{11} + X^3 + X^2 + 1$
- $CRC32 = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X^1 + 1$

网络工程-网络层知识

CRC循环冗余校验

- 校验码的具体生成过程为：假设要发送的信息用多项式 $C(x)$ 表示，将 $C(x)$ 左移 R 位（可表示成 $C(x) \cdot x^R$ ），这样 $C(x)$ 的右边就会空出 R 位，这就是校验码的位置。用 $C(x) \cdot x^R$ 除以生成多项式 $G(x)$ 得到的余数就是校验码。

网络工程-网络层知识

CRC循环冗余校验

■ 举例：

需要发送的信息为 $M = 1010001101$ ，生成多项式

$G(x) = X^5 + X^4 + X^2 + 1$ ，所对应的生成式二进制代码为 $P = 110101$ ，

$R = 5$ （冗余码的位数）。

网络工程-网络层知识

CRC循环冗余校验

- 在 M 后加5个0，然后对 P 做模2除法运算，得余数 $r(x)$ 对应的代码：01110。故实际需要发送的数据是101000110101110。

```
1 1 0 1 0 1 | 1 0 1 0 0 0 1 1 0 1 0 0 0 0 0
              1 1 0 1 0 1
              1 1 1 0 1 1
              1 1 0 1 0 1
              1 1 1 0 1 0
              1 1 0 1 0 1
              1 1 1 1 1 0
              1 1 0 1 0 1
              1 0 1 1 0 0
              1 1 0 1 0 1
              1 1 0 0 1 0
              1 1 0 1 0 1
              0 1 1 1 0
```

网络工程-网络层知识

CRC循环冗余校验

■ 错误检测

当接收方收到数据后，用收到的数据对 P （事先约定的）进行模2除法，若余数为0，则认为数据传输无差错；若余数不为0，则认为数据传输出现了错误。

网络工程-网络层知识

CRC循环冗余校验

■ 说明：

- CRC可以检错，但不能用于自动纠错；
- 只要经过严格的挑选，并使用位数足够多的除数 P ，那么出现检测不到的差错的概率会很低；
- 用CRC循环冗余检验只能做到无差错接受（只是非常近似的认为是无差错的），并不能保证可靠传输。

网络工程-网络层知识