一种时钟恢复算法及其 FPGA 实现

吴 铁 钟洪声

（成都电子科技大学电子工程学院 四川 绵阳 610054）

摘 要 时钟恢复(CDR)电路被广泛的应用于数据传输的各个方面。针对数字基带传输系统，对未知频率的码元信号，本文提出一种在宽频范围内自适应的时钟恢复算法，能够从频率为 2KHz 到 10MHz 的非归零码中恢复出时钟信号。本算法是基于数字逻辑的，特别适合在可编程逻辑器件上实现。最后用 Verilog[1] 编程并下载到 FPGA，给出了仿真波形和该系统的性能参数。

关键词 时钟恢复 数字锁相环 FPGA

1 引言

在数字通信系统中，位同步是首要解决的问题。其含义是：接收机和发射机的时钟必须保持同频、同相。常用的同步方法有插入导频法和直接法[2]。插入导频法的基本原理是：在基带信号频谱的零点 $f=1/T$ 处，插入所需的导频信号。直接法的原理是：直接从接收到的数字信号中提取出位同步信号。直接法较导频法具有更高的传输效率，因此在实际中常常采用直接法。

直接法中一般都要用到锁相环。一种思路是：将包含位同步信息的非归零码元信号经过非线形变换，经过滤波，再用一般的锁相环跟踪所期望的频率[3]。这种方法对对滤波器的要求比较苛刻，而且受噪声的影响比较大，产生同步信号的范围由滤波器决定。另一种思路是直接用锁相环从非归零码元中提取位同步信号。这种方式的难点在于鉴相器的设计，即消除在码流中缺乏时钟变化沿时，鉴相器产生的错误误差信号。有很多学者在这方面做了大量的工作[4,5]。

本文的思路仍是用锁相环直接提取位同步信号，但具体的方法不同：用最简单的异或门鉴相，用电路内部的高速数字逻辑，首先判断输入码元时钟信号频率，鉴别输入码元的时钟相位，再由数控振荡器产生与发送端时钟同频同相的时钟信号。该方法最大的优点是，输入信号的码元时钟频率范围很大（约为 2kHz 到 10MHz）。用随机变化的非归零信号作为输入，在 20 个码元时钟周期以内捕获的概率为 0.979122。由于这个方法是基于数字逻辑的，可用全数字电路实现，也非常适合用可编程逻辑器件实现。

2 时钟恢复算法及实现

2.1 信号沿检测模块

为了便于描述，先给出系统框图如图 1 所示。框图的左上部分是一个信号沿检测器。它的作用就是检测信号的变化，当信号变化时该模块产生一个脉冲，脉冲的宽度是固定的，且与本地振荡频率相关。该模块的输出与输入信号的时序关

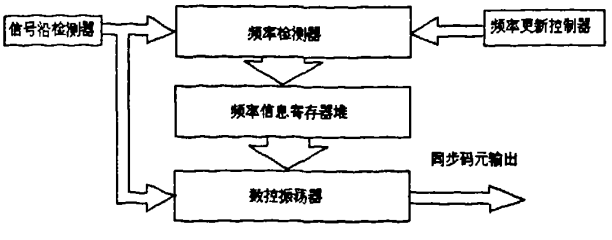

图 1 系统框图

系图如图 2 所示。

该模块可以由一个普通的异或门鉴相器实现。假设本地时钟周期为 T，则信号沿检测的误差 E_t 范围为：$-T \le E_t \le T$。

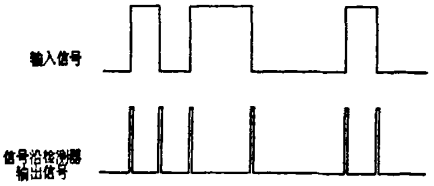

图 2 鉴相器输入输出时序图

2.2 频率检测模块

从非归零码中恢复出位同步信号的主要挑战在于：怎样处理好持续时钟周期码元无变化的情况。理论上，一个码元可能持续任意多个时钟周期。为了检测出码元的时钟频率，在这里假设：在检测的这段时间内，信号中存在两个相邻的信号沿，它们的间隔时间正好为一个时钟周期。如果把信号看作一串 0，1 序列，则信号中一定要包含“010”或“101”序列。

码元中出现“010”或“101”序列的概率与码元长度是有关系的。假设信号码元的高低电平是随机出现的，对于一个码元长度为 n+1 的信号段，可以推算出信号中出现“101”序列或“010”序列的几率。记一个码元长度为 n+1（n≥2）的信号中，包含“010”或“101”的排列个数为 S_n，可以证明，数列 S_n 的递推公式为：

$$S_n = S_{n-1} + S_{n-2} + 2^{(n-2)} \tag{1}$$

且：$S_2 = 2,\ S_3 = 6$

又长度为 n+1 的码元序列可检测的概率为：

$$P_{n+1} = \frac{S_n}{2^n} \tag{2}$$

经计算得：$P_{20} = 0.979122,\ P_{30} = 0.997492$

可见，随着码元长度的增加，可检测的概率是逐渐增加并接近 1。所以在实际应用中这种假设是可行的。

在前面的假设条件下，设信号沿检测器在一段时间内输出的相邻两个脉冲间的时间间隔为：$T_1, T_2 \ldots T_n$，则时钟周期 $T_{clk} = T_{min}$。因此，该算法中是通过在 $T_1, T_2 \ldots T_n$ 中寻找 T_{min} 来检测位同步信号的频率。在图 1 中，频率检测器和频率寄存器堆就是用来从 $T_1, T_2 \ldots T_n$ 寻找 T_{min} 的。它们的工作流程如图 3，图中的计数器是指频率检测器中用于计量时间的高速计数器。它在本地时钟的驱动下，在信号沿检测器输出控制下计数。寄存器中存储的数值就是被检测信号的时钟频率信息。

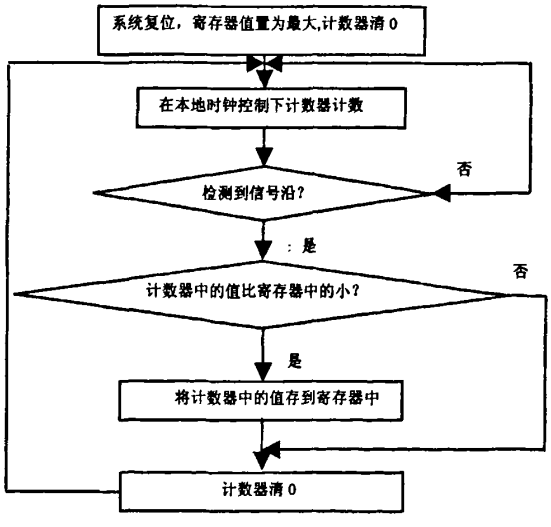

图 3 频率检测器流程图

2.3 数控分频器

前面的模块都是对输入信号的频率和相位进行检测。位同步信号的产生是由图 1 中的数控分频

器模块产生的。该模块读取寄存器堆中的频率信息，以及信号沿检测器输出的相位信息，调整对本地时钟信号的分频系数和输出位同步信号的相位，输出与输入信号相适应的位同步信号。该模块由一个多位的、模值可变的可逆计数器构成，并能在信号沿检测器输出脉冲的控制下置数和产生脉冲。

该模块还有一个重要的功能：它能根据前一个码元的持续时间动态的调整输出位同步信号的周期。设前一码元的持续时间为 L_n，前一个位同步时钟周期为 T_n，下一个位同步时钟周期为 T_{n+1}，假设上一个码元持续了 m 个时钟周期，数控振荡器的最小可调周期为 Δ,则：

$$T_{n+1}=\begin{cases}T_n-\Delta & (L_n<m\cdot T_n)\\ T_n+\Delta & (L_n>m\cdot T_n)\\ T_n & (L_n=m\cdot T_n)\end{cases} \tag{3}$$

也就是说，在外部频率变化不大时，该模块能自动的调整输出位同步信号的频率。而且，该功能也弥补了频率检测器所检测的频率不连续（由于是由计数器记录频率信息，而计数器本身的工作频率是有限大的）的缺点。所以在系统稳定工作时，输出的位同步信号的频率将在实际的码元时钟频率的小范围内抖动。该范围取决于计数器的工作频率以及输入码元的时钟频率。

2.4 频率更新控制器

前面已经提到，从码元中检测时钟信号的方法是寻找相隔最近的相邻信号沿。在输入信号的时钟频率变高时，该系统能迅速地检测到更新后的频率。但若输入信号的时钟频率变低时，这种寻找相隔最近的相邻信号沿的方法就不适用了。因此，需要外部电路控制频率检测器，让它“忘记”以前的频率信息，重新开始检测。当码元频率变高时，频率检测器能够捕获新的频率，因此不需要这个模块的参与；反之则需要这个模块产生信号触发频率检测器重新检测。

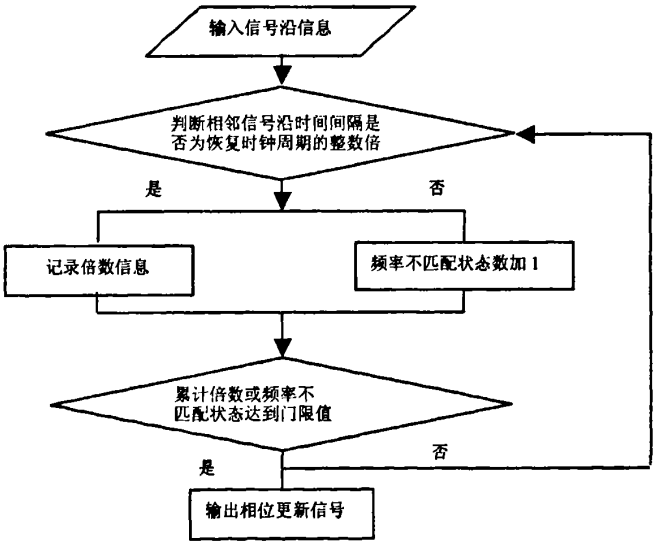

图 4 频率更新控制器流程图

频率更新控制器能够记忆前几个相邻信号沿之间的间隔，用逻辑的方法判断出频率寄存器堆中记录的频率信息是否与之相适应，它的功能可由流程图简要的表达。图中描述的是对一个信号沿间隔的处理过程。最后输出频率更新信号是根据记录的最近几个状态判断的。比如：当检测到最近记录的 8 个信号沿时间间隔，相对前一时刻所检测到的码元时钟周期的倍数的最大公约数大于 1，或者在最近检测的 8 个信号间隔时间，超过一半都与前一时刻所检测到的码元的时钟周期不匹配，这时频率更新信号就会有效。

2.5 系统性能

锁相环的精确程度主要由数控（或压控）振荡器决定。数控振荡器主要是由一个分频器实现。它的具体工作原理就是在外部信号控制下对本地高频振荡信号进行分频。在 FPGA 的实现过程中，

采用 altera 公司的 EP1C3T1446C 型 FPGA，在 Quartus II 软件环境下用 Verilog 语言编程，编译以及仿真。该码元同步电路的工作频率最大可达 130MHz。也就是说，输出位同步信号的时钟周期分辨率为 8 纳秒。

捕获频率范围：数控振荡器中的分频器是采用 16 位的计数器，所以能够输出的最低位同步信号频率为 2kHz。当输入码元频率逐渐逼近数控振荡器工作频率时，由于输入码元时钟的周期减小，而频率检测器可检测的最小时钟周期差别为 8 纳秒，所以会造成检测高频信号的精度降低。经实践验证，最大码率为 10MHz 时系统能稳定工作。故该系统能恢复的位同步信号频率范围为 2kHz 到 10MHz。

相位误差 θ_e 及同步保持时间:由于恢复的位同步信号周期的最大误差为 8 纳秒，所以每个周期的最大相位误差 θ_{max} 为：

$$\theta_{\max} = 2\pi \cdot \frac{8}{T} \tag{4}$$

$$|\theta_e| \le \theta_{\max} \tag{5}$$

其中 T 为输入码元的时钟周期（单位为纳秒），假设码元持续了 n 个时钟周期，则 n 个周期的最大相位误差为：

$$\theta_{\mathrm{nmax}} = n \cdot \theta_{\max} = 2\pi \cdot n \cdot \frac{8}{T} \tag{6}$$

可见当输入码元频率越高，码元保持 0 或 1 不变的时间越长，可能出现的相位误差就越大。同步保持时间 t_h 也由此可得：

$$t_h > \frac{2\pi}{\theta_{\max}} \cdot T \tag{7}$$

同步建立时间：当输入码元频率由低变高时，由于此算法是基于：检测最相邻的时钟沿。所以当输入码元频率由低变高时，能立即检测到位同步信号的变化。该系统立即开始检测新的输入码元时钟，当出现新的“010”或“101”序列后，该系统就处于捕获状态。当输入码元频率由高变低时，该系统需要 8 个输入信号沿来确认输入码元时钟的变化，确认后再开始检测新的输入码元时钟，检测到“010”或“101”序列后处于捕获状态。所以同步建立时间取决于输入码元频率的大小以及变化方向。

2.6 仿真结果

在 Quartus II 下经过编译，时序仿真结果如图 5 所示。其中第一排为 100MHz 的时钟输入信号；第二排是各种频率的码元信号的时钟信号；第四排是输出的位同步时钟信号。第五排是在对应码元时钟频率下产生的伪随机码信号，作为该系统的信号输入。可见，在输入码元频率变化后经过 8 个时钟沿后，系统捕获到新的码元频率。

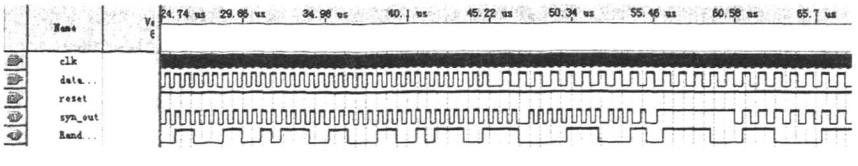

图 5 软件仿真波形图

2.7 结语

本文描述了一种时钟恢复算法以及其实现的逻辑。它能从非归零 PCM 波形中恢复出信号时钟，时钟频率的捕获范围为 2kHz 到 10MHz。通过用 Verilog HDL 对该算法进行描述，并在 Quartus II 平台上编译、仿真，给出了系统的捕获时间和误差等性能。而且，本设计已经在硬件平台上通过验证，完全符合设计要求。

参考文献

[1] Michael D.Ciletti.《Advanced Digital Design with the Verilog HDL》。电子工业出版社。2004。

[2] 段吉海，黄智伟。《基于 CPLD/FPGA 的数字通信系统建模与设计》。电子工业出版社。2004。

[3] 张厥盛，郑继禹，万心平。《锁相技术》。西安电子科技大学出版社，2002.

[4] C.R.Hogge, "A-self-correcting clock recovery circuit".J.LightWave Technol.,vol.LT-3,pp.1312-1314,1985.

[5] Gijun Idei and Hiroaki Kunieda. "A False-Lock-Free Clock/Data Recovery PLL for NRZ Data Using Adaptive Phase Frequency Detector." IEEE Transaction On Circuits And Systems—II: Analog And Digital Signal Processing, VOL. 50, NO. 11, NOVEMBER 2003

The algorithm of Clock Recovery and its realization based on FPGA

Wu tie，Zhong hongsheng

(School of electronic engineering, UESTC, Chengdu Sichuan 610054, China)

Abstract: Clock and Data Recovery (CDR) circuits have been widely used in data communication systems. In digital base band communication systems, an adaptive clock recovery algorithm is proposed, which is able to recover the bit-synchronous signal from the NRZ signal in the frequency ranging from 2MHz to 10MHz without awareness the input frequency. The algorithm is based on logic, which is especially suitable for realized on programmable devices. The algorithm is described in Verilog and downloaded to FPGA. The simulation result and the system performance are also proposed.

Keywords: clock recovery; Dpll, FPGA