CRC编码运算中移位寄存器的设计

许正荣[1,2],贾贤龙[2],李 阳[1],杨敦毅[1]
(1. 安徽农业大学 信息与计算机学院,安徽 合肥 230036;
2. 合肥工业大学,安徽 合肥 230009)

摘 要:循环冗余校验(CRC)是一种编码简单且有效的串行数据校验方法,在通信及计算机数据存储中得到了广泛应用。在串行CRC编码实现中,移位寄存器主要完成将并行输入数据转换成串行输出数据的功能,是整个设计的重要组成部分。以发送8位信息码为例,在Altera公司的开发工具Quartus II软件下,分别选用数字集成电路芯片74LS166和VHDL编程两种方法,成功地完成了移位寄存器的设计,可以满足不同的应用需求。仿真结果准确、可靠,符合设计需要,有一定的实用意义。

关键词:循环冗余校验;移位寄存器;74LS166;VHDL
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2008)11-0103-03

Design of Shift Register in CRC Code Operation

XU Zheng-rong[1,2], JIA Xian-long[2], LI Yang[1], YANG Dun-yi[1]
(1. School of Information and Computer Engineering, Anhui Agricultural University, Hefei 230036, China;
2. Hefei University of Technology, Hefei 230009, China)

Abstract: Cyclic redundancy check(CRC) is widely used in communication and data storage because of high dependability and easy of realization. The shift register's function is completion of parallel data input into serial data output. The design of shift register is an important part in the realization of CRC code. Takes the transmission of 8 bit data as an example, selects two methods of integrated circuit chip 74LS166 and VHDL, in Altera Corporation's Quartus II software. It has completed the design of shift register successfully and met the needs of different applications. The simulation result is accurate and reliable, conforms to the design need, has certain practical significance.

Key words: cyclic redundancy check; shift register; 74LS166; VHDL

0 引 言

数字信号在传输过程中会受到干扰的影响,通常由信道中乘性干扰引起的码间干扰,可以采用均衡的办法纠正;而加性干扰的影响则主要从合理地选择调制/解调方式、发送功率以及差错控制措施等方面考虑,从而尽可能提高通信的可靠性。循环冗余校验(CRC,cyclic redundancy check)是对一个传送数据块进行校验,是一种高效、检错能力很强的差错控制方法,而且实现编码和检码的电路简单,因此,在数字通信和计算机发展的很多领域有着广泛的应用[1]。

CRC运算的实现方法有多种[2,3],可通过软件方式计算,也可通过硬件方式计算,还可以通过查表方式获得[4]。运用硬件方法时,可由除法电路来实现[2]。除法电路的主体是一些移位寄存器和模2加法器,笔者分别选用集成电路芯片74LS166和VHDL语言两种方法,在Altera公司的Quartus II软件下,完成CRC编码运算中移位寄存器的设计,使并行输入的数据转换成串行输出数据。

1 CRC运算原理

循环冗余校验码是在严密的代数学理论基础上建立起来的,它采用多项式编码方法,将发送数据的串行位流看成系数为"1"和"0"的多项式($a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\cdots+a_1x+a_0$,系数$a_{n-1},a_{n-2},\cdots,a_1,a_0$为二进制数"1"或"0"),然后使用发送方和接收方"商定"的同一个生成多项式来做除法运算,并将发送数据和其校验码一起发送到接收端,再利用相同的生成多项式对所接收到的数据进行除法运算,如果能够整除,则传输无错误;否则,传输有错。二进制除法按照"模2"运算进行,加减时不进、借位,与逻辑异或运算

收稿日期:2008-02-21
基金项目:安徽省高校青年教师项目资助(2007jq1047)
作者简介:许正荣(1974-),女,硕士,讲师,研究方向为EDA技术、通信电路与系统。

一致。

循环冗余校验的编码步骤是:首先将信息元多项式 $m(x)$ 乘以 x^{n-k} 成为 $x^{n-k}m(x)$,然后将 $x^{n-k}m(x)$ 除以生成多项式 $g(x)$ 得到余式 $r(x)$,该余式就是校验元多项式,从而可得到码字多项式 $c(x) = x^{n-k}m(x) + r(x)$。

在用硬件实现上述运算步骤时,除法电路的主体由一些移位寄存器和模 2 加法器组成。不同的生成多项式需要移位寄存器和模 2 加法器有不同的连接方法。例如,当生成多项式为 $g(x) = x^4 + x^3 + x^2 + 1$ 时,具体的编码运算电路如图 1 所示。图中移位寄存器的个数为 $g(x)$ 最高项的次数,故有 4 级移位寄存器,分别用 a、b、c、d 表示,反馈线的连接与 $g(x)$ 的非 0 系数相对应,从右至左分别为 11101(由高位至低位)。另外有一双刀双掷开关 s,它的动作将控制信息位的直接输出、除法电路运算所得的余数存入寄存器以及监督码元(即余数)的输出。

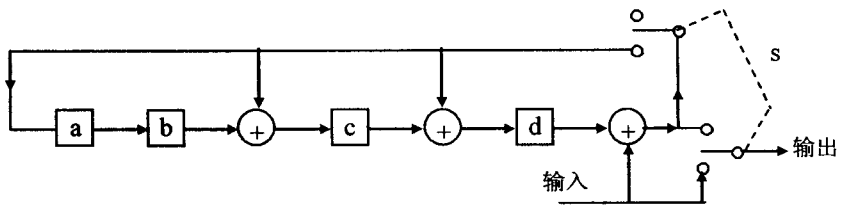

图 1 CRC 编码运算原理图

显然,在 CRC 编码运算中移位寄存器的设计非常重要,随着基于 PLD(可编程逻辑器件)的 EDA(电子设计自动化)技术的发展和应用领域的扩大与深入,目前已多采用一些先进器件和相应的开发软件来实现上述编码过程。

2 移位寄存器的设计

移位寄存器是一个具有移位功能的寄存器,是指寄存器中所存的代码能够在移位脉冲的作用下依次左移或右移。右移是指数据由左边最低位输入,依次由右边的最高位输出;左移时,右边的第一位为最低位,最左边的则为最高位,数据由低位的右边输入,由高位的左边输出。既能左移又能右移的称为双向移位寄存器,只需要改变左、右移的控制信号便可实现双向移位要求。根据移位寄存器存取信息的方式不同分为:串入串出、串入并出、并入串出、并入并出四种形式。

移位寄存器应用非常广泛,可构成移位寄存器型计数器:顺序脉冲发生器;串行累加器;可用于数据转换,即把串行数据转换为并行数据,或把并行数据转换为串行数据等。本实验研究移位寄存器用作数据的并、串行转换功能。

在研究 CRC 编码运算时,选用了 Altera 公司的 EDA 技术开发软件 Quartus Ⅱ 4.0。Altera 是世界上最大的可编程逻辑器件供应商之一,Quartus Ⅱ 是该公司提供的 FPGA/CPLD(现场可编程门阵列/复杂的可编程逻辑器件)开发集成环境,它提供了一种与结构无关的设计环境,使设计者能方便地进行设计输入、快速处理和器件编程。同时,Quartus Ⅱ 提供了完整的多平台设计环境,能够满足各种特定设计的需要,其设计工具完全支持 VHDL(VHDL'87 及 VHDL'97 标准)、Verilog 的设计流程;可以利用第三方的综合工具;具备仿真功能,也支持第三方的仿真工具。Quartus Ⅱ 软件的应用方法和设计流程对应其他流行的 EDA 工具的使用具有一定的典型性和一般性。

使用 Quartus Ⅱ 软件设计项目,逻辑设计的输入方法有图形(原理图)输入、文本输入、波形输入及第三方 EDA 工具生产的设计网表文件输入等。输入方法不同,生产的设计文件也不同。本次实验所采用的实例为发送端发送的并行数据为 8 位信息码(11011001),分别采用原理图输入方法(使用集成芯片 74LS166)和文本输入方法(使用 VHDL 编程)来设计移位寄存器[5],即并行输入的数码,经转换电路之后变换成串行输出。

2.1 基于 74LS166 的移位寄存器设计

74LS166 是一个带清零端的 8 位并行加载的移位寄存器,当控制端 SHIFT/LOAD = '1' 时是移位状态,在时钟脉冲上升沿的控制下右移,数据输入端 A～H 的信号就是装载到移位寄存器的 QA～QH;基于原理图输入的 74LS166 如图 2 所示,在图中,输入端 IN0～IN7 为八个并行的数据输入(由低位至高位),QH 端为

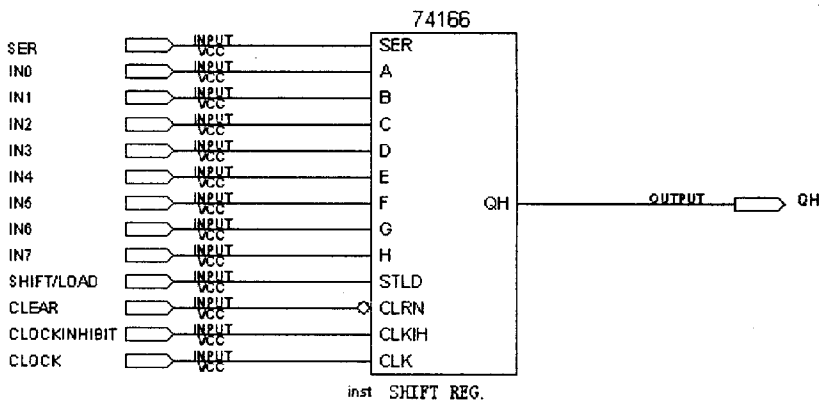

图 2 基于原理图输入的 74LS166

串行数据的输出端。

软件仿真波形如图 3 所示,其中 CLOCK 是发送端的时钟;CLOCKINHIBIT 是时钟禁止端,当它为'1'时将禁止时钟;当 SHIFT/LOAD = '1'时,在时钟脉冲上升沿的控制下已装载的并行数据右移(波形图中,从左至右 QH 为 8 位信息位的高位至低位输出)。

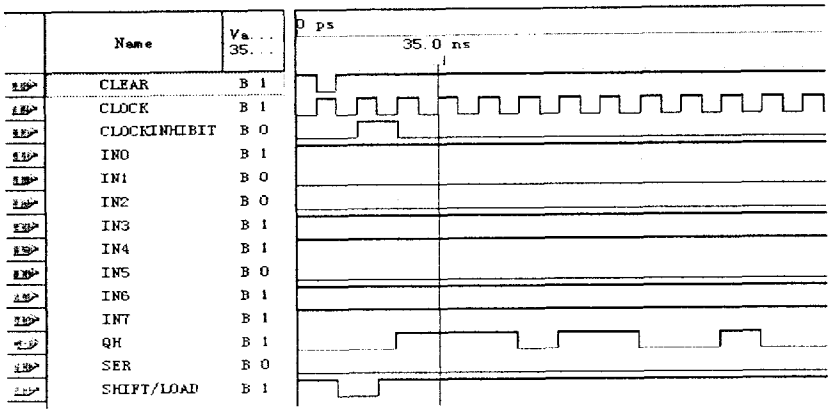

图 3　基于 74LS166 的移位寄存器仿真图

2.2　基于 VHDL 语言的移位寄存器设计

硬件描述语言(Hardware Description Languang, HDL)是一种能够以形式化方式描述电路的结构和行为并用于模拟和综合的高级描述方法,是 EDA 技术的重要组成部分。常见的 HDL 主要有 VHDL、Verilog HDL、System Verilog 和 System C 等几种,其中 VHDL、Verilog 在现在 EDA 设计中使用最多,VHDL 是硬件描述语言的业界标准之一,在电子设计领域得到了广泛应用。

用 VHDL 语言设计移位寄存器时,主要采用进程来描述它的工作原理。程序的主要代码如下:

```
PROCESS (CLK, LOAD)
VARIABLE REG8: STD_LOGIC_VECTOR (7 DOWNTO 0);
BEGIN
  IF CLK'EVENT AND CLK = '1' THEN
    IF LOAD = '1' THEN REG8: = DIN;
    ELSE REG8 (6 DOWNTO 0): = REG8 (7 DOWNTO 1);
    END IF;
  END IF;
  QB< = REG8 (0);
END PROCESS;
```

上述代码是一个带有同步并行预置功能的 8 位右移寄存器的核心部分。CLK 是移位寄存器时钟信号,DIN 是 8 位并行预置数据端口,LOAD 是并行数据预置使能信号,QB 是串行输出端口。当 CLK 的上升沿到来时进程被启动,如果这时预置使能 LOAD 为高电平,则将输入端口的 8 位二进制数并行置入移位寄存器中,作为串行右移输出的初始值;如果预置使能 LOAD 为低电平,则执行语句"REG8(6 DOWNTO 0): = REG8(7 DOWNTO 1)"。

仿真波形如图 4 所示,当第一个时钟到来时,LOAD 恰为高电平,此时 DIN 口上的 8 位数据为 D9(11011001),即"11011001"被锁入 REG8 中。第二个时钟,以及以后的时钟信号都是移位时钟(波形图中,从左至右 QB 为 8 位信息位的低位至高位输出)。值得注意的是,赋值语句 QB< = REG8(0)在 IF 语句结构外面,因此它的执行并非需要当前的时钟信号,属于异步方式。即在当前的时钟信号到来前,由于 IF 语句不满足时钟条件而跳出 IF 语句,于是便执行其后的语句 QB< = REG8(0)(在第一个执行并行数据加载的时钟后,QB 即输出了被加载的第一位右移数"1",而此时的 REG8 内仍然是"D9")。寄存器内数据变化情况见表 1。

表 1　基于 VHDL 的移位寄存器内数据变化情况

并行输入	寄存器值		串行输出(由低位至高位)
	二进制	十六进制	
11011001	11011001	D9	1
	11101100	EC	0
	11110110	F6	0
	11111011	FB	1
	11111101	FD	1
	11111110	FE	0
	11111111	FF	1
	11111111	FF	1

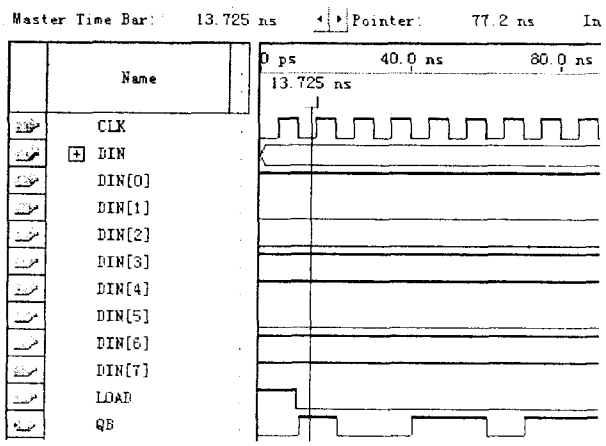

图 4　基于 VHDL 的移位寄存器仿真图

2.3　结　论

图 3、图 4 的仿真波形都是 CRC 编码运算中移位寄存器的实现,两种方法都完成了数据的并行输入向串行输出转换的功能。原理图输入方法显现设计的层

(下转第 108 页)

Rosenbrock 函数：$f_2 = \sum_{i=1}^{n-1}(100(x_{i-1} - x_i^2)^2 + (x_i - 1)^2)$，$x_i \in [-2.048, 2.048]$

Rastirgrin 函数：$f_3 = \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10)$，$x_i \in [-5.12, 5.12]$

Griewank 函数：$f_4 = \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$，$x_i \in [-600, 600]$

表 3 Rastirgrin 函数的实验结果

算法	维数	优化结果的均值
PSO	20	1.103487
	30	6.025851
MPSO	20	0.132085
	30	1.828539

表 4 Griewank 函数的实验结果

算法	维数	优化结果的均值
PSO	20	0.030128
	30	0.081736
MPSO	20	0.008361
	30	0.015027

从表 1～表 4 的对 4 个函数的实验结果可以看出，MPSO 比标准 PSO 有着更好的搜索性能。

4 结束语

标准 PSO 算法由于粒子采用统一的惯性权值，忽视了粒子的独特性特点，种群的多样性不好，易于早熟收敛、搜索精度不高，易于陷入局部最优解。惯性权值 w 直接影响 PSO 算法的局部和全局搜索能力，处于不同位置的粒子对全局和局部搜索的要求也不一样，因此改进的 MPSO 算法的惯性权值 w 做动态调整，并且在粒子群惯性权值 W 总体减少的同时，针对每个粒子所处的不同位置惯性权值 W 再做相应的微调；改进后的新算法跳出局部最优，同时加快了收敛速度，具有较强的全局搜索能力，显著提高了优化性能。

参考文献：

[1] Kennedy J, Eberhart R C. Particle swarm optimization[C]// Proceeding of 1995 IEEE International Conference on Neural Networks. New York, NY, USA: IEEE, 1995: 1942 - 1948.

[2] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]//Proceedings of the Sixth International Symposium on Micro Machine and Human Science. New York, NY, USA: IEEE, 1995: 39 - 43.

[3] Shi Y H, Eberhart R. Parameter selection in particle swarm optimization[C]//Proc of the 7th Annual Conf on Evolutionary Programming. Washington D C: [s. n.], 1998: 591 - 600.

[4] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]//Proceedings of 1998 IEEE International Conference on Evolutionary Computation. New York, NY, USA: IEEE, 1998: 69 - 73.

[5] Shi Y, Eberhart R C. Empirical study of particle swarm optimization[C]//Proceedings of the Congresson Evolutionary Computation. Piscataway, NJ: IEEE Service Center, 1999: 1945 - 1950.

[6] Bergh F, Engelbrecht A P. A Cooperative Approach to Particle Swarm Optimization[J]. IEEE Trans. on Evolutionary Computation, 2004, 8(3): 225 - 239.

(上接第 105 页)

次化，简单明了；文本输入方法具有类似于高级程序设计语言的抽象能力，便于多人配合工作。需要注意的是两种方法下输出数据的高位和低位次序正好相反，在实际使用时，根据需要可作适当的变换。

3 结束语

这种使用集成电路芯片 74LS166 和 VHDL 语言通过现代电子系统设计方法来实现串行 CRC 编码运算中移位寄存器的方案，既增强了对数字逻辑电路的理解，又在系统灵活性和系统可移植性等方面有了一定的提高。实际结果表明，两种方法都能够完全达到预定的设计要求，有一定的实用与借鉴价值。后续工作中，CRC 译码器设计及数据的串行/并行转换模块设计值得去做进一步探讨。

参考文献：

[1] 曹雪虹，张宗橙. 信息论与编码[M]. 北京：清华大学出版社，2004.

[2] Ramabadran T V, Gaitonde S S. A Tutorial on CRC Computations[J]. Micro IEEE, 1988, 8(4): 62 - 75.

[3] Derby J H. High - speed CRC computation using state - space transformations[C]// IEEE Global Telecommunications Conference GLOBECOM'01. San Antonio, TX, United States: [s. n.], 2001: 166 - 170.

[4] 蒋安平. 循环冗余校验码(CRC)的硬件并行实现[J]. 微电子学与计算机，2007, 24(2): 107 - 112.

[5] 潘 松，黄继业. EDA 技术实用教程[M]. 北京：科学出版社，2002.